A ROBUST METHODOLOGY FOR STRATEGICALLY DESIGNING ENVIRONMENTS SUBJECT TO UNPREDICTABLE AND EVOLVING CONDITIONS

A Thesis
Presented to
The Academic Faculty

by

Ethan T. Minier

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
May 2019

A ROBUST METHODOLOGY FOR STRATEGICALLY DESIGNING ENVIRONMENTS SUBJECT TO UNPREDICTABLE AND EVOLVING CONDITIONS

Approved by:

Professor Dimitri N. Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Professor Daniel P. Schrage
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Olivia J. Pinon-Fischer
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Simon I. Briceno
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Morvarid Rahmani
Scheller College of Business
Georgia Institute of Technology

Date Approved: January 11, 2019

ACKNOWLEDGEMENTS

First and foremost, I am beyond grateful for the opportunity afforded to me by Professor Dimitri Mavris to pursue a Doctor of Philosophy at Georgia Tech. He took a chance on me six years ago when others did not, and for that, I will forever be grateful. He has continually encouraged me, and all those around him, to strive for excellence both academically and professionally, yet at the same time to remain humble in this pursuit. His compassion, guidance, and care know no bounds. He has truly been instrumental in further forming me into the person I am today.

I would further like to thank each of my committee members who have contributed greatly to this work. Dr. Olivia Pinon-Fischer has invested more time than I could have ever hoped for in helping shape this work into what it is today. Throughout the process she has not only guided me, but been a great mentor, teacher, and friend. Thank you also for encouraging me in my forward-looking pursuits outside of the classroom. Dr. Morvarid Rahmani has also been instrumental in the success of this work. Since the first day I met her as an MBA student in her class, she has continually inspired my pursuits, several of which became important components of this dissertation. Professor Daniel Schrage, from the beginning, has provided valuable insights and constructive criticism, both of which have been important in further advancing this work. Finally, thank you Dr. Simon Briceno for joining my committee and providing yet another source of perspective and advice.

On a personal note, I am beyond grateful for everyone who has encouraged and supported me throughout not just this dissertation process, but also my complete journey academically. Specifically, I would like to acknowledge my parents, Bill and Denice Minier, who have always been supportive of my pursuits. They instilled in me a work ethic, perseverance, dedication, and creativity; each of which became instrumental during this dissertation process. Without their support, love, and guidance in life, I truly believe I would not be where I am today. Also, thank you for continually asking the question: when are you going to get a job? I would also like to thank my brother, Justin Minier, for his support and constant reminder that I will always be his little brother and that he would never call me Dr. Minier. Thank you for making me an uncle to the most amazing twins, Chase and Peyton Minier. Perhaps one day, they will be engineers like their uncle! Finally, thank you Teagan Kimpton for your continually support, excitement, and empathy over the last year and half. You handled with grace my crazy workload as I completed my MBA in just a years' time all while continuing to make progress on this dissertation. Thank you for keeping my spirits high even during the most stressful times during this process. You truly are my guiding light and for that I am beyond grateful!

There are many other loved ones and individuals who have supported me, guided me, and shaped me into the person I am today. Individuals like Mr. Tanea, Mr. Harrington, and Coach Ceravolo all from back in my primary school days, along with Professor Dannenhoffer from my years at Syracuse University and where this journey all began. Thank you once more to those of you I personally acknowledged along with everyone else unnamed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

5S	Sort, Set in order, Shine, Standardize, Sustain
ACO	Ant Colony Optimization
b	Cost of facility placement
B&B	Branch-and-Bound
BSH	Boundary Search Heuristic
COP	Combinatorial Optimization Problem
CSA	Colonal Selection Algorithm
d	Distance between facilities
DLP	Dynamic Layout Problem
DP	Dynamic Programming
f	Presence of flow between facilities; objective function
FPA	Flower Pollination Algorithm
FSA	Fast-Simulated Annealing (Fast-SA)
g	Vector of inequality constraints
GA	Genetic Algorithm
h	Vector of equality constraints
I/O	Input / Output
k, c	User specified FSA parameters
LCS	Longest Common Subsequence
LP	Layout Problem

MHC Material Handling Cost

MILP Mixed Integer Linear Programming

MINLP Mixed Integer Non-Linear Programming

MIP Mixed Integer Programming

n Number of facilities and locations; iteration number index

N Number of departments

NP Non-deterministic polynomial-time

P Probability of acceptance

PSO Particle Swarm Optimization

QAP Quadratic Assignment Problem

QAP/S Quadratic Assignment Problem / Structured

QAP/U Quadratic Assignment Problem / Unstructured

QAP/U-SP Quadratic Assignment Problem / Unstructured – Sequence-Pair

RC Rearrangement Cost

RDLP Robust Dynamic Layout Problem

RLP Robust Layout Problem

RSLP Robust Static Layout Problem

S Current configuration

S' New configuration

SA Simulated Annealing

SLP Static Layout Problem

SP	Sequence-pair
T	Number of periods; annealing temperature
T	Indexed period
T_0	Initial annealing temperature
T_1	Initial annealing temperature
tmax	Number of periods
TPC	Total Penalty Cost
TS	Tabu Search
VLSI	Very-Large-Scale-Integration
x	Vector of continuous variables
X	Vector of continuous variable domains
y	Vector of discrete variables
Y	Vector of discrete variable domains
Δavg	Normalize average quality change for the current annealing temperature
ΔQ	Objective function (quality) change
Δ_{UHavg}	Average uphill quality change for uphill moves
λ	Annealing cooling rate
JGO	Jumping Gene Operator
FSPPM	Feasible Sequence-Pair Promoting Method

SUMMARY

To manage the growing challenge of remaining competitive in today's saturated market, businesses in the manufacturing industry often turn to lean manufacturing practices. The layout design process, a lean technique, has the potential to provide a manufacturer with significant reductions in operating and capital costs. The major challenge for layout designers is then ensuring these benefits can not only be maximized, but also realized when implemented in practice. Guaranteeing this realization requires both the real-life behavior and characteristics of the environment as well as the market and business model conditions to adequately be captured.

Unfortunately though, after an extensive literature review, it is identified that current methods fail to accurately capture real-life considerations such as flow path feasibility while evaluating a layout design's performance, consider continuous representations of evolving layout designs subject to financial restrictions and uncertainty, and moreover provide sufficient insight into the design of an environment subject to evolving and uncertain conditions. It therefore became the objective of this research to establish an improved methodology for exploring the design space of a detailed evolving environment, enabling more informed and collaborative design decisions to be made in the presence of evolving and uncertain market and business model conditions.

In the process of achieving this goal, critical gaps in the literature are identified and systematic approaches subsequently formed to provide closure to said gaps. A methodology, titled LIVE, is formed during this process. Along with its formation an

extensive array of novel methods, revolutionary optimization techniques, and new applications of existing genetic operators are developed in addition to a detailed performance model; all to facilitate the effective solution to the uniquely complex and arduous formulation of the layout problem requiring solution in this dissertation. The composition of these methods, approaches, and models form, what is collectively referred to as, the bi-model multi-stage solution approach. It is then believed, if the problem of designing an environment subject to evolving and uncertain market and business model conditions was to be solved with this LIVE methodology then, designers would be capable of making more informed and collaborative decisions on its design.

Substantiation of this is then pursued following the formation of this methodology and further the development of the bi-model multi-stage solution approach deployed by it. The methodology, and the approaches it deploys, are subsequently systematically tested according to an experimental approach. In the first stage of the solution approach, whereby a quadratic assignment problem, unstructured, sequence-pair model is leveraged to represent the layout, the novel feasible sequence-pair promoting method developed to handle the unique characteristics of the problem relating to constrained objects in the space is tested. It is shown to be effective at promoting the more frequent discovery of feasible designs in comparison to the standard random assignment method of the literature. It is further shown that it enables problems otherwise unsolvable to then become solvable. Following this testing, the importance of considering flow path feasibility while designing a layout is proven to be crucial. Failure to do so confirms that sub-optimal designs would be produced by the layout design process. It is further demonstrated that the novel advanced flow distance method developed to provide such consideration was effective in doing so, leading to designs far better representative of reality and thus more optimal.

Next, the two stages of the developed bi-model multi-stage solution approach are examined for their effectiveness in providing solution. Optimization parameter studies are performed to identify the settings of the parameters that best facilitate effective solution. It is demonstrated that different setting should be deployed depending on the problem type being solved; dynamic vs. static. Ultimately, the best settings to deploy are identified leveraging main effects plots, an analysis-of-variance, and a weighted average approach to balance key metrics where applicable.

Finally, while applying the LIVE methodology to a real-world layout design problem, it is shown that the methodology effectively facilitates improved insight and potential collaboration into the layout design process. The implemented performance model proves significant in enabling new insights to be drawn and further for a richer understanding of the operations and layout design to be gained. Overall, the methodology demonstrates its ability to provide an improved layout design process that can effectively handle design problems subject to uncertain and evolving conditions; enabling strategic business decisions to be considered in parallel to the design of the layout.

CHAPTER 1

INTRODUCTION

A major challenge faced by businesses globally is remaining competitive in today's saturated market. To manage this challenge, some seek entrance to new or emerging markets while others implement business strategies to improve their operational effectiveness. By reducing operating costs, businesses can choose either to absorb the larger profits that result or reduce costs for their customers. The latter of these scenarios enables the business to appear more affordable and therefore more appealing to consumers. As one can predict, this appeal is vital to remaining competitive in a saturated market. In addition to these two measures, some may also alter their business model, going through periods of restructuring or production redistributions. Dawar and Frost observed the necessity for this in stating that when "globalization pressures are strong, managers can't just build on their company's local assets; they will have to rethink their business models" [53]. More often than not, businesses will be required to perform one or multiple of these measures in order to remain competitive. In the manufacturing industry, businesses often turn to lean manufacturing practices to manage this growing challenge.

1.1 A Paradigm Shift in the Manufacturing Industry

In the early 1950's the Toyota Motor Company, inspired by the simple waste elimination concepts developed by Henry Ford in the early 1900's, implemented (what are considered to be by most) the first advanced forms of lean manufacturing [1,2]. The success Toyota had incorporating these techniques caught the attention of the industry, inspiring a paradigm shift in how manufacturer's approached operations. This paradigm shift from the previous approach of batch production to that of lean production was further aided in 1990 by Roos, Jones, and Womack's publication of the book "The Machine That Changed the World" [172]. Their book highlighted MIT's five-year, five-million-dollar study on the future of the automobile industry and further made lean production a worldwide known term. The lean production approach focuses on the systematic elimination of all waste (i.e., inefficiencies) related to an organization's operations. Since its initial acceptance, and as estimated by Tompkins, White, Bozer, and Tanchoco in their textbook Facilities Floorplanning, on an annual basis since 1955 about 8% of the gross national product (GNP) in the United States has been delegated towards the development of new, more efficient facilities [163].

1.1.1 The Value of Lean Manufacturing

These early lean manufacturing strategies implemented by the Toyota Motor Company have since been advanced and expanded, forming what are now commonly recognized as the eight core techniques of lean manufacturing. They include the kaizen rapid improvement process, 5S (sort, set in order, shine, standardize, sustain), total productive maintenance, cellular manufacturing, just-in-time production, six sigma, pre-production planning, and lean enterprise supplier networks [1].

Implementation of any one or several of these techniques concurrently can provide several benefits. It is widely observed that performance improvements in the range of 30 to 70 percent can be achieved through practicing lean manufacturing [1]. In general, lean techniques help reduce operational costs through improved performance, as just observed, and capital manufacturing costs by enabling profitable consolidation options and minimal initial facility investments to be identified. The latter cost has never been better exemplified than by Goodrich Aerostructures' use of lean methods to facilitate the consolidation of their operations while concurrently increasing production output. Furthermore, consolidation of their operations enabled them to recover initial capital investment costs through the sale of previously required properties [43].

Reductions in these costs have the major advantage of resulting in reduced overhead and increased profit margins. They also have the added benefit of improving the environmental performance of the manufacturer's production flow [1]. As such; these techniques can make a manufacturer more competitive and increase their likelihood of entering new or emerging markets with success. These substantial benefits have led to such lean techniques becoming adopted by manufacturers at an accelerating rate [1]. As a result, the following assertion can be made:

> **Assertion 1:** As more companies begin to adopt and benefit from these techniques, transitioning to lean production has become less about improving one's competitiveness and more about a necessary action to remain competitive in the national and global markets.

1.1.2 Benefits of Layout Design

Of the eight core lean techniques, cellular manufacturing, a subset of the broader concept of layout design, produces some of the most significant performance improvements and capital savings for the general manufacturer. This is a result of it being the first of the methods that produces a major adjustment in operations making it a key enabler for increased production velocity and flexibility [2]. This is achieved by adjusting the design of the environment with the goal of improving the overall operational flow of material as is notionally demonstrated in Figure 1. This improvement in the operational flow and thus performance can quantifiably be observed as reductions in operating costs. With that said, to what degree or frequency the environment can realistically be adjusted is limited by several factors including financial restrictions.

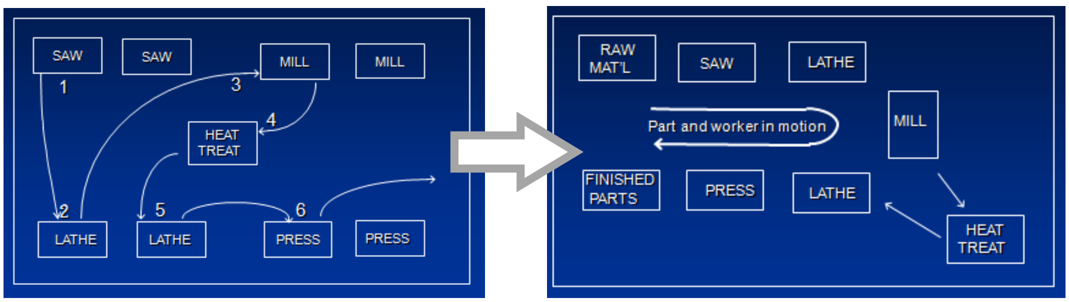


Figure 1 – Example of implementing cellular manufacturing to improve flow [2]

Although a disruptive technique, as production must be halted to adjust the layout of the environment, the reward is high. Material handling costs represents anywhere from 20-50% of a manufacturer's operating costs and 15-70% of the total cost of manufacturing a product [163]. Therefore, even marginal improvements to the design of the layout can yield large savings over time. In fact, it has been estimated that anywhere between 10-30% annually can be saved through the reduction of operating costs with effective layout design [71]. Layout design also has the benefit of providing potential capital cost reductions as it enables consolidation options and minimal facility size requirements to be identified. As observed before, Goodrich Aerostructures is an example of how implementing cellular manufacturing enables manufacturers to consolidate and reduce capital costs. Its implementation enabled them to consolidate their operations into two facilities from five while simultaneously doubling output. They were in turn able to sell the remaining three facilities thereby reducing their initial capital costs [43].

1.1.3 Importance of Addressing Layout Design in Manufacturing

Given the paradigm shift, the observed importance of layout design, and the earlier observation regarding the percentage of GNP in the United States being delegated towards new facilities (8%), it is reasonable to then assume that more than 250 billion is spent annually on layout or relayout processes [100]. With such a large sum of capital being invested in layout design each year it is hard to ignore the problems relevance. The importance of addressing layout design as it pertains to the manufacturing industry in particular can be understood by acknowledging the role that it plays in the economy.

According to the Bureau of Economic Analysis, manufacturers contributed 2.17 trillion dollars to the U.S. economy in 2015, a statistic that since 2009 has been continually rising. Furthermore, the manufacturing industry accounted for over 12 percent of the gross domestic product in the U.S. economy in 2015 [40]. The manufacturing industry also retains one of the higher economic multiplier's in that for every $1 spent in manufacturing, another $1.81 is added to the economy as a whole [40]. These statistics demonstrate how important the manufacturing industry is to the health of the U.S. economy. With layout design acting as the primary enabler to more effective operations and therefore performance of the U.S. manufacturing industry as a whole, layout design in manufacturing becomes a topic of great importance.

In addition to these noteworthy statistics it should also be acknowledged that the majority of manufacturing businesses in the U.S. are relatively small in size. As of 2013, of the 251,857 manufacturing businesses, all but 1.46% where considered to be small businesses (i.e. having less than 500 employees). Furthermore, just a quarter of the other 248,155 firms had over 20 employees [166], meaning that over 70% of the manufacturing businesses in the U.S are very small businesses. These businesses must rely on lean manufacturing strategies to remain afloat and competitive against not only the giants in their market, but also the substantial number of other small business present. For this large percentage of businesses, layout design becomes that much more important to their success in the market.

1.2 The Layout Design Problem

The extent to which these manufacturers can potentially benefit from layout design has since been established as being quite significant; however, the most effective layout must first be identified before any of these benefits can be realized. Identifying this layout is achieved by solving, what literature classifies as, the layout problem (LP). The quality of this problem's solution becomes imperative to ensuring the maximum benefit is achieved from performing this layout design process.

Besides the solution method's ability to establish this desired solution, two other factors greatly contribute to the degree of benefit that can be realized from the layout design process. One is how well the problem formulation captures the necessary detail of the environment to accurately establish a solution of realistic viability: more accurately the model captures the real-life behavior of the environment, the more viable the design will be when actually implemented in practice. The second is how well the solution process accounts for the market and business model conditions the environment will experience during operation: the better these conditions are captured, the better suited the solution established by the design process will be at ensuring the maximum benefit is achieved.

1.2.1 Major Layout Design Challenges

These two factors also coincidentally correlate to the two major challenges faced by manufacturing layout designers. The first of these challenges relates to how accurately the real-life behavior of the environment is captured by the problem formulation. In an attempt to ensure the realistic viability of the layout can be adequately assessed, it has required that the problem formulation become significantly more complex. The solution difficulty is proportional to this complexity however. Therefore, the more realistic the problem formulation is, the more complex the problem becomes and the more difficult it then is to solve. As such, balancing how accurately the problem formulation captures the real-life behavior and detail of the environment with the problem's solvability is a challenge often faced by designers. As observed before however, the benefit that can be realized from layout design is directly influenced by how well the environment is modeled. If one seeks to achieve the maximum benefit from the layout design process, then sacrificing the model accuracy becomes unwise. Therefore, the major challenge for designers then becomes how to preserve the problem's computational tractability as the layout detail and design capability requirements increase.

The second of these challenges relates to ensuring that the conditions the environment will ultimately face are adequately captured. Operating conditions, or more fundamentally market and business model conditions, are often highly unpredictable and prone to fluctuations with time. Therefore, a major challenge faced by designers is how to accurately define the forecasts of these conditions such that the designed layout remains effective when implemented.

It should be understood that manufacturing designers are not the only designers facing these challenges, designers across several industries are also having to manage these challenges. Designers of VLSI (Very-Large-Scale-Integration) circuits, ships, houses, airports, hospitals, and many others are also experiencing these challenges, more so with respect to the second of the two challenges. For this reason, solution to the layout design problem and the development of more effective methods is relevant and useful across many industry applications, not just that of the manufacturing industry and the design of manufacturing environments.

1.2.2 Managing These Challenges

1.2.2.1 Evolution of the Problem Formulation

Since its original formulation, the layout problem has continually evolved, becoming more complex and difficult to solve. Demands for improved design capabilities and greater layout detail have in large part driven this evolution. These demands have, more often than not, been derived from the desire to maximize the benefit that can be realized from performing the layout design process. Early efforts focused on very basic models of the environment that encapsulate a limited degree of detail. These models represent layouts discretely where the departments of the environment are either stacked upon each other or more often, placed in an apriori prescribed set of possible locations. Furthermore, they neglect details such as interior obstructions (pillars, walls, safety zones, etc…), inputs and output points of the departments, and flexible departments (i.e. departments whose size can change) to name a few. Even to this day, a majority of the research focuses on these basic models of the environment due to their solution being considerably more manageable compared to formulations that provide more layout detail.

In recent years however, research has begun to shift towards formulations that better represent the environment's actual characteristics and behavior as designers seek improved design viability. These formulations employ continuous representations of the layout, which provide designers with improved design capabilities. The inclusion of the aforementioned layout details has also accompanied this, further providing designers with a more realistic depiction of the environment. Though such formulations improve a designer's ability to establish realistic designs, the added complexity and difficulty of their solution has prevented such problems from being extensively studied in the literature. Researchers that have entertained such a problem formulation have had to manage the challenge of keeping such a problem computationally tractable. This has often required effective solution methods to be developed. The limited research present in the literature regarding the effective solution of such a detailed problem formulation is the first of the noteworthy gaps present. As will be observed, this gap only widens as other characteristics of the problem are coupled with such a problem formulation.

1.2.2.2 Presence of a Major Gap in the Assessment of a Layout Design

The performance metrics that establish a given design's effectiveness also contribute to the capability of establishing realistically viable layout designs. Performance metrics that inadequately account for the real-life behavior of the environment can lead to inferior layout designs. Across the literature the standard measure of a manufacturing layout's performance is its material handling cost. This is comprehensible given the earlier observation of how largely material handling contributes to the operating and total manufacturing costs.

The majority of research implements rudimentary methods of establishing the material flow distances for the segments of a process performed in the environment, which when coupled with unit process segment flow costs, enables the material handling cost to then be defined. These methods typically neglect flow path feasibility when determining these distances present in the environment. An extremely limited amount of research has considered the importance of addressing this flow path feasibility, which, from the author's previous observations, can greatly impact the design deemed most effective. In some cases, this yields a design that is over thirty-five percent less effective in practice than one generated while considering such flow feasibility. As such, this presents a noteworthy gap in the literature that required addressment in this dissertation.

1.2.2.3 An Unpredictable and Evolving Landscape

As noted before and also observed by Kulturel-Konak in his comprehensive review of approaches to uncertainty in the LP, capturing the dynamics of the global economy is another major challenge faced by layout designers [100]. The dynamics of the market are often volatile, unpredictable, and consistently evolving. This makes it not only difficult, but also highly unrealistic, to capture the precise market conditions that the environment will experience over the span of its planned lifetime. Furthermore, the models that businesses employ, often to account for these changes in the market, are also likely to evolve. Dawar and Frost, recognized this parallel evolution when stating that "not only will managers find their strategies likely to evolve over time, but the nature of their industry may change as well" [53]. With so much ambiguity present, it becomes a principle concern of designers, seeking a layout design with long-term viability, to establish accurate methods of accounting for said uncertainty and evolution in the market conditions and business model.

1.2.2.3.1 Manufacturing Uncertainties

As it applies to the manufacturing industry, there are two categories of uncertainties. The first of these are uncertainties associated with internal disturbances such as equipment breakdowns, queuing delays, and rework. The second of these are uncertainties derived from external forces such as product demand levels, product mixes, product market values, manufacturing costs, layout restructures, and equipment changes [100,152]. Addressing uncertainties pertaining to internal disturbances is often the main objective of what is referred to in the literature as the scheduling problem. On the other hand, addressing uncertainties associated with external forces is often the primary concern of the layout problem. These external uncertainties collectively address variations in the aforementioned market and business model conditions and for this reason the following assertion can be made:

> **Assertion 2:** Accounting for these uncertainties in the layout design process is essential to accurately designing a layout that will remain effective over the course of its planned lifetime and as market conditions and business practices evolve.

1.2.2.3.2 Addressing Uncertainty and Evolution of the Conditions

Too often, the difficulty of establishing methods that adequately account for the uncertainty and evolution in these conditions and further the solution of such problems has discouraged researchers in the field of manufacturing layout design. As such, far less research has been performed in this area. In recent years however, researchers have begun to acknowledge the value and more often the necessity of such an endeavor.

1.2.2.3.3 Accounting for the Evolutionary Nature of the Conditions

To account for the likely evolution of market conditions and business strategies, researchers have entertained allowing the layout design itself to evolve concurrently. Nicoli and Hollier established the importance of such an approach to the problem when identifying that at least a third of a manufacture's key operations are replaced in just over three years, with nearly half of the companies surveyed having replacement occur in two years or less [130]. With operations changing so frequently, the layout's performance will understandably be affected, necessitating the need for the environment to evolve accordingly before the end of its planned lifetime.

Often such evolving approaches partition the planned lifetime into periods of shorter length where the layout is allowed to evolve (i.e. be rearranged) from one period to the next to provide a more effective layout design for the ensuing period of expected conditions. Too frequently however, these partitions are defined in a uniform and structured manner with evolution occurring at the onset of each period. This can be limiting to the understanding of the evolving layout design problem and further are poorly representative of actual planning schedules. This presents yet another, albeit minor, gap in the literature that must be addressed.

1.2.2.3.4 Growth of the Overarching Problem Gap

This evolving approach to the problem, as one can postulate, further increases not only the solution space, but also the difficulty of the problem. A good amount of research has addressed this approach, but only a marginal percentage has considered such an approach coupled with a detailed problem formulation. Observation of this acknowledges the growth of the overarching problem gap mentioned earlier.

For the layout to be capable of rearranging under this approach, sufficient financial resources must also be available at the time of this evolution. This places an additional constraint on the problem, one that is often neglected by researchers in the literature. To the author's knowledge, only a few [17,48] have accounted for this capital resource restriction on the evolution of the layout, despite its importance in establishing evolving layout designs that are viable. Furthermore, none of this research overlaps with a detailed problem formulation and dynamic approach to the problem. This subsequently establishes the aforementioned gap as a major one in the literature that had to be addressed in this dissertation.

1.2.2.3.5 Accounting for the Uncertain Nature of the Conditions

Evolution of the conditions is also accompanied by uncertainty, where this uncertainty is expected to grow the further downstream the layout is planned for. As noted before, establishing the precise forecasts of these conditions is unrealistic. To overcome this challenge, researchers, in recent years, have begun to adopt stochastic approaches to the layout design problem that focus on providing designs that are robust to the unpredictable nature of these conditions. The value of such a robust approach has caused it to become the focal point of research in this century. As demonstrated in Figure 2, researchers have more often adopted this concept of robustness while completely abandoning the outdated one of dominance (i.e. optimality) in the 21st century. Solution to the problem that incorporates a detailed problem formulation, dynamic approach, as well as robustness has scarcely been performed in the literature. The difficulty of the problem's solution has largely contributed to this gap in the literature despite such a problem providing the designer with extensive design flexibility and accuracy.

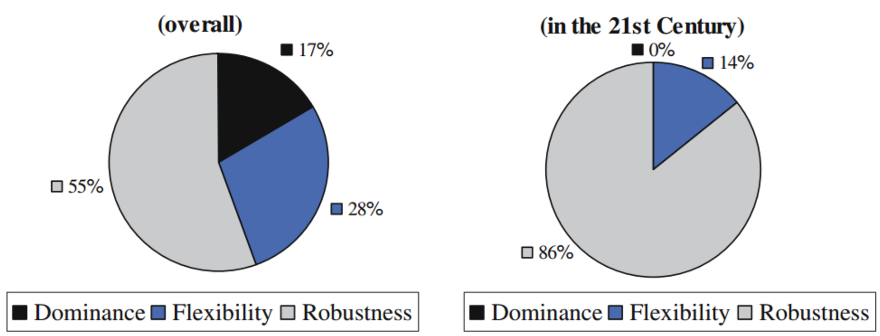


Figure 2 – Adoption of designing a layout for robustness this century [100]

Instead of defining the forecasts deterministically, as has often been done in the past, researchers have begun to define them stochastically in order to establish layout designs that will remain effective over a range of different potential conditions. In other words, layout designs which are robust. Conventionally, either a scenario based or statistical modeling-based method is employed to capture this uncertainty in the problem. Scenario based methods rely on a predefined set of discrete scenario representations of the conditions. A given design is then evaluated according to each of these collectively to assess its overall effectiveness. Statistical modeling based methods on the other hand represent the uncertainty continuously by modeling the conditions as random variables of known distribution parameters.

In each of these methods, the uncertainties in these conditions are infused directly into the solution procedure. As a byproduct of this, each of these techniques suffers from a lack of problem transparency. The impact that these conditions have on the performance of the design cannot be directly observed by a designer, which greatly impedes the designer's ability to understand the influence that these conditions have on the design of the environment. This in turn inhibits the designer's ability to make more informed decisions regarding its design. From this and the observation of the aforementioned gaps in the literature, the following assertion can be made:

> **Assertion 3:** A systematic and efficient methodology is needed to enable the exploration of a large combinatorial design space, support quantitative trade-off analyses, and improve problem insight to facilitate a more informed and collaborative selection of a realistically viable layout design in the presence of highly uncertain and evolving conditions.

1.3 Improving the Layout Design Process

As observed before, there remains much room for improvement in how environments subject to unpredictable and evolving market conditions and business models are designed. To ensure the maximum benefit is achieved by performing the layout design process, the process of designing the environment must adequately account for the evolution and uncertainty associated with the operating conditions of the environment. Furthermore, the process must adequately model the environment's real-life behavior and characteristics. Failure to do so on either account will diminish the benefit provided by performing the layout design process and therefore reduce the company's competiveness in the market. The focus of this dissertation is therefore to improve the layout design process with the goal of enabling more realistically viable designs to be identified and more informed decisions regarding its design to be made, all with the intention of maximizing the benefit that can be attained by performing this process.

To facilitate the establishment of more realistically viable designs, this dissertation addresses the detailed problem formulation. Details such as layout boundaries, interior obstructions (pillars, walls, safety zones, etc…), inputs and output points of the assets, and layout continuity are collectively addressed to ensure the environment's real-life characteristics are adequately accounted for. Inclusion of layout boundaries further enables existing building redesigns to be performed while including input and output points in the formulation helps provide more accurate material handling cost assessments. Additionally, the aforementioned major gap regarding the assessment of this material handling cost is addressed and an advanced flow distance method that ensures flow feasibility pursued. The intent is for this advanced flow distance method to supplement a detailed cash-based performance model, which will provide access to additional performance data beyond that of what is conventionally provided by methods implemented in the literature to solve the layout problem. It is believed that access to this additional data will enable more informed and strategic business designs to be made regarding the design of the layout and moreover the system as a whole.

Another focus of this dissertation is in identifying a viable solution method that will be effective in solving the problem formulated in this manner and moreover will provide effective solution in the presence of evolving and uncertain operating conditions. A major emphasis is on providing improved transparency to the designer regarding the impact that the evolution and uncertainty associated with these conditions has on the design of the environment. To facilitate this, a flexible approach to considering the evolution of the design and its robustness to uncertain operating conditions is pursued. Pursuit of such an approach in conjunction with the aforementioned detailed problem formulation attempts to fill the aforementioned overarching gap in the literature, i.e. provide effective solution to the robust, evolving, and detailed layout problem. Though it is unlikely that a rapid solution of such an intractable problem will be attainable, the layout planning process is rarely a time sensitive action. Typically, layout planning is a process that occurs over several months, thus solving such an involved characterization of the problem at the expense of considerably larger computational solution times is not all that concerning. A synthesis of these goals and observations establishes the following objective of the research:

> **Research Objective:** To establish an improved and robust methodology for exploring the design space of a detailed evolving manufacturing layout, enabling more informed and collaborative design decisions to be made under evolving and uncertain market and business model conditions.

Though the application and thus focus of this research is on designing manufacturing layouts and analyzing system level business decisions relating to this application, it is intended that the robustness of this methodology will enable it to be applied to other layout design applications and accompanying layout performance related business decisions. This declared research objective then leads to the first major research question:

> **Research Question 1:** How does one effectively design a manufacturing environment subject to evolving and uncertain market and business model conditions?
>
> > **Sub Question 1:** How are manufacturing environments modeled?
> >
> > **Sub Question 2:** How is the evolution and uncertainty associated with these conditions accounted for?
> >
> > **Sub Question 3:** What defines a well performing layout design?
> >
> > **Sub Question 4:** How are such layout designs established (i.e. solved for)?

In the section that follows, the literature is explored in an attempt to more thoroughly acknowledge the previously observed gaps. Furthermore, it seeks to provide further motivation as to why a robust, evolving, detailed layout problem must be solved to ensure realistically viable layout designs are established in the presence of unpredictable and evolving conditions. This motivation will be established as each of the sub-questions of **Research Question 1** are explored and subsequently answered.

CHAPTER 2

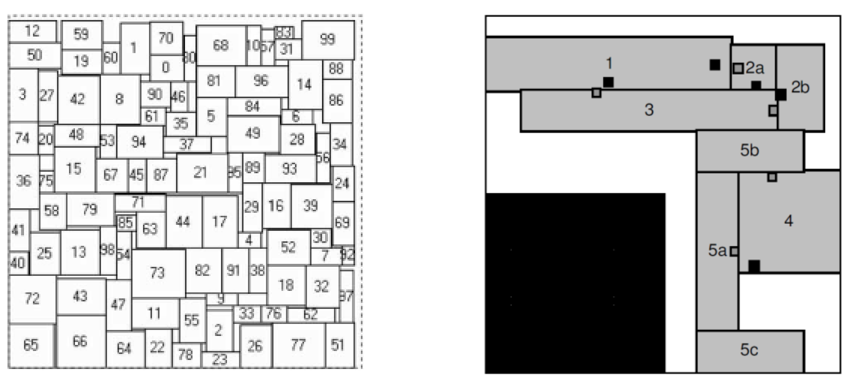
MOTIVATION AND BACKGROUND

As it pertains to the problem of this dissertation, this chapter provides an expansive survey of the literature on the layout problem. As the chapter progresses and the problem is more thoroughly understood, the scope of the research discussed will narrow accordingly such that only research of direct relevance and general applicability to the problem is observed. The densely researched nature of the layout problem makes this narrowing essential to prevent this document from becoming excessively long and nothing more than a survey of the problem, which is not the intent of this dissertation. Furthermore, as the chapter progresses the aforementioned **Research Question 1** of how to effectively design a manufacturing environment subject to evolving and uncertain conditions will be answered by addressing each of its sub-questions stated before. Before diving into addressing these sub-questions an understanding of the layout problem itself, general relevance, and characteristics is required.

2.1 The Layout Problem

The layout problem (LP) is a combinatorial optimization problem (COP) that attempts to identify the position of objects relative to one another within a layout for which a predefined measure of layout performance is maximized. Said problem has been extensively researched with a wealth of it having studied the use of different formulation and solution methods. These studies span across several fields and applications which include, but are not limited to, ship, building room, VLSI circuit board, and manufacturing facility design with the latter two being the most commonly studied [108,30,47,87,21].

Research on the LP can often be categorized according to two fundamental problem characteristics; brown versus green and rough versus detailed layouts where a complete formulation encompasses at least one characteristic of each of the divisions (e.g. a rough green LP). The brown versus green characteristic establishes whether the problem is constrained or unconstrained with the latter being representative of a green LP. A green LP, such as the one demonstrated in Figure 3a, does not have external boundaries that constrain its dimensions and therefore the outer dimensions of each solution to the problem are likely to be different [154]. Brown LPs, like the constrained example in Figure 3b, differ from green LPs in that they are confined to a fixed area.



(a) **(b)**

Figure 3 – A (a) rough green layout problem [47] (b) detailed brown layout problem [21]

The second division; rough versus detailed, is representative of the level of detail being considered. The rough LP is only concerned with achieving a relative block layout, as demonstrated in Figure 3a where the blocks are simply packed together. The detailed

LP, on the other hand, simultaneously accounts for characteristics such as I/O points, aisles, and infeasible regions when solving the problem [21]. Research on the detailed LP is far less extensive as it is often more computationally expensive and difficult to solve. Constructing the LP requires formulating the problem and identifying a suitable solution method. Both of the divisions identified above greatly contribute to the formulation required and solution method that will be most effective in solving said problem.

As was established in the preceding section, researchers have sought capturing more layout detail in their formulations as the demand for improved design capabilities has grown considerably since the LP's initial inception. Varying levels of detail have been captured by researchers. Types of layout detail studied in the literature include, but are not limited to, fixed boundary constraints or its variant, fixed aspect ratio boundary constraints (brown vs green distinction), flexible non-rectangular and same size objects, input and output points (I/O points) of objects, multi-floor layouts, aisle and routing paths, fixed objects (pillars, interior walls, stations, etc…), and safety buffers (additional spacing about objects) [21,30,47,87,108]. In general, the ordering of this list also corresponds to the relative frequency of implementation, where boundary constraints are the most often addressed and details such as aisle and routing paths, fixed objects, and safety buffers are those less often studied. How these details are captured in the problem formulation is directly influenced by the model implemented to represent the layout. This observation acknowledges the first of the sub-question, **Research Question 1.1**, regarding how layouts are modeled?

2.2 Approaches to Modelling the Layout

There are three main model formulations of the LP. These include formulating it as a quadratic assignment (QAP), a mixed integer programming (MIP), or a graph theory problem where the latter is the less commonly implemented and as such will not be discussed further. For information on graph theory and studies performed using it in relation to the LP readers may refer to Foulds and Hassan and Hogg [70,86]. Of the two more common models implemented, the QAP formulation of the problem has dominated the research focus. Recently though this focus has shifted more towards the MIP formulation as the desire for improved layout representation and analysis capabilities has grown. The major differentiation between these two models is in their continuity property. As will be observed, the QAP is capable of only discrete layout representation whereas the MIP can represent continuous layouts, or in other words layouts in which the coordinate positions of the objects can take any value between a range of boundary values and are defined independently of one another [127,100]. This distinction becomes important when addressing the problem of this dissertation.

2.2.1 The Quadratic Assignment Problem Formulation

The most general formulation of the LP is as a QAP. The original mathematical model for the QAP was introduced by Koopmans and Beckmann in 1957 to solve the problem of allocating indivisible economic resources [97]. In other words, they sought to allocate a set of facilities to a set of locations, thereby establishing the relative position of each to one another. This is nothing more than the LP in its simplest form. Despite being the most basic form of the LP, it remains a difficult problem to solve. The problem maintains an NP-hard solution complexity and as such, no exact algorithm can solve problems that are larger than twenty objects in size to optimality [154,149,41,103]. The mathematical model proposed by Koopmans and Beckmann and applied generally to the LP is as follows [97,41]:

$$\min_{\varphi \in S_n} \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} d_{\varphi(i)\varphi(j)} + \sum_{i=1}^{n} b_{i\varphi(i)}$$

where *f* represents the presence of flow between facilities (discrete: 0 or 1), *d* the distance between the facilities, *b* the cost of facility placement, and *n* the number of facilities and locations. As can be observed, the model structure is discrete in nature. It provides a purely binary or integer representation of the variables and for this reason only discrete layouts can be evaluated [154]. This inherent discreteness leads to the following important observation:

> **Observation 1:** QAP models are fundamentally incapable of representing continuous layouts.

2.2.1.1 Topological Representations of the Layout

To represent a layout in a binary or integer string structure that can then be efficiently manipulated, a topological mapping between this representation and the physical layout is required [47]. Topological representations facilitate this mapping or characterization of the physical layout. Some of these representations are more effective than others and some more flexible. Topological representations can be divided into two groups, structured and unstructured representations.

Structured topological representations will from here forth be referred to as QAP/S formulations of the problem with the *S* denoting the structured nature of the formulation. This type of representation correlates to the binary string format where the location in which objects can be allocated are predefined and known. Under this condition the problem embodies precisely that of Koopmans and Beckmann's original formulation where the object is or either is not placed in a given location. This is limiting as one must a priori know the potential locations of the objects in the environment. As one can then visualize, such a representation forms a gridded skeleton or structured layout of the objects, which is how it gets its structured distinction. Such a formulation is highly constraining to the problem making it less than ideal for application to the problem of this dissertation. With that said, the majority of research on the LP has been on such a model. This is a result of these restrictions producing more tractable problems that researchers can more easily manage.

Unstructured topological representations, which will be referred to as QAP/U formulations of the problem going forward, remove these restrictions on the physical layout and do not require a priori definition of the object location in the environment. Instead these representations maintain a discrete data structure that models the geographical relationships among entities (e.g. left, right, below, above relations) through the implementation of stacking based rules [170]. Several variants of these unstructured representations have been proposed in the literature. Normalized polish expressions, B*Tree, and sequence-pair representation are among the most popular. Others include Corner Block List (CBL), Twin Binary Sequences (TBS), Transitive Closure Graph (TCG), TCG-S, O-Tree, Corner Sequence (CS), BSG, and Adjacent Constraint Graph (ACG) [170].

A normalized polish expressions representation is a skewed binary tree model that implements pruning measures to prevent redundant mappings from existing [170]. This representation characterizes the layout by slicing it horizontally or vertically. The B*-Tree representation is also a binary tree model, but its tree is ordered. The benefit of this structure is that it guarantees a unique B*-Tree to physical layout mapping and the presence of an area-optimal solution [47,170]. The latter relates to its inherent packing nature making it ideal for solving green layout designs where a minimal area design is often desired. As shown in Table 1, the contour data structure of the B*-Tree allows for the evaluation process to be performed in amortized linear-time [170].

The last representation, the sequence-pair, is the most flexible, or put another way the least rigid, of the three. Its meta-grid data structure enables it to retain more information than the prior representations are capable of retaining. As such it can represent general layout designs rather than just sliced or compact ones. This overall layout flexibility is a result of the ordered sequence-pair (SP) model this representation employs to characterize the layout [170,128,129,160].

Table 1 – Comparison of popular topological representations [170]

Representation	**Solution Space**	**Packing Time**	**Flexibility**
Normalized Polish Expression	$O(n!2^{3n}/n^{1.5})$	O(n)	Slicing
B*-Tree	$O(n!2^{3n}/n^{1.5})$	O(n)	Compacted
Sequence-Pair	$(n!)^2$	O(n log log n)	General

As is often the case, the added flexibility that the SP representation provides comes at a cost. Even with the longest common subsequence (LCS) technique formulated by Tang, Tian, and Wong implemented (the most efficient approach in the literature) to evaluate the SP in $O(n \log \log n)$ time, this evaluation time that is required to pack/unpack the SP remains greater than that of the other two popular methods [160]. Despite the added computational time associated with the SP representation, from here forth referred to as the QAP/U-SP formulation of the problem, and its larger solution space, as shown in Table 1, its ability to characterize a general layout design makes it the best topological representation option of the three for solving the problem of this dissertation. As such the following assertion can be made:

> **Assertion 4:** Of the QAP layout models and as it pertains to the problem of this dissertation, the sequence-pair representation (QAP/U-SP) is the most suitable.

2.2.1.2 The Sequence-Pair Representation (QAP/U-SP)

Murata, Fujiyoshi, Nakatake, and Kajitani (1995) proposed the first formulation of the P-admissible guaranteeing sequence-pair representation and an algorithm evaluating said representation for the VLSI layout design problem [128,129]. The P-admissible characteristic of the representation requires that the space be finite, every solution be feasible, evaluation time be polynomial, and a representation that corresponds to the optimal solution exists. The requirement for every solution being feasible in the space is imperative to ensuring proper convergence by heuristics as it maintains continuity amongst the solutions.

Murata et al.'s (1995) original formulation relied on a longest path algorithm for vertex weighted directed acyclic graphs to evaluate a SP representation in $O(n^2)$ time and obtain the physical layout [128]. Takahashi built upon this original formulation by proposing an algorithm capable of evaluating a sequence-pair in *O(n log n)* time [156]. The major shortcoming of his algorithm was that it could provide the height and width of the layout, but not the actual individual block positions.

Tang et al. later sought to provide an algorithm that could not only provide the block positions, but furthermore improve the algorithms evaluation time of the SP. They improved the algorithm for evaluating a sequence-pair by proposing what they titled the Fast-SP [160]. Their algorithm, which relies on a longest common subsequence (LCS) technique to facilitate the unpacking of the SP, is capable of *O(n log log n)* evaluation times. This evaluation time includes both establishing the layout dimensions, as Takahashi was able to achieve earlier, as well as the block positions in the physical layout. In a continuation of this work, Tang, as part of his dissertation, and Tang and Wong extended the formulation to encompass the handling of placement constraints while retaining the same evaluation time as noted before. These constraints included that of fixed block placements, block range placements, and layout boundaries [161,159].

Table 2 – Capability of Tang and Wong's SP formulation in modeling key characteristics of the layout

Characteristic	Capability
Layout Boundaries	*Yes*
Overlap Avoidance	*Yes*
Fixed Objects	*Yes*
Flow Feasibility	*Quasi*
Layout Continuity	*No*

QAP/U model formulations of the problem inherently guarantee that the blocks do not overlap in the space. This is the result of the relative relationships (left, right, above, below) these representations employ to characterize the layout. Coupling this with Tang and Wong's formulation of the sequence-pair representation, which further accounts for layout boundaries and fixed objects in the space, such a formulation of the problem enables layout solutions that are realistically viable to be established. Establishing layout designs that are more realistically viable is one of the main objectives of the dissertation, which makes this formulation of the problem promising. Flow path feasibility throughout the layout is yet another characteristic of a layout that distinguishes its realistic viability. If the appropriate spacing for material to flow about the blocks is accounted for by expanding each blocks spatial footprint, then said formulation of the problem can in a quasi-way account for flow feasibility. By expanding each blocks spatial footprint, it indirectly accounts for flow feasibility by ensuring that material can flow between each and every block. In reality however, it may be more beneficial to account for this spacing between only certain blocks and that is where this formulation becomes limited.

Although a QAP/U-SP model is the most capable of the QAP formulations in accounting for the majority of the essential layout characteristics of the problem, **Observation 1** acknowledges that such a formulation remains insufficient in characterizing a continuous layout due to its fundamental mathematical construct. This is reflected in Table 2 which summarizes the characteristics that such a formulation can sufficiently capture. Understanding this limitation, the following conclusion can be made:

Conclusion: Although a SP representation (QAP/U-SP) is the most suitable, its inability to represent continuous layouts establishes the need for a more flexible modeling approach.

Despite the QAP/U-SP model's inadequacy in characterizing the layout, its discrete data structure does make it more conducive to a solution. As such, the following question is posed:

Question 1.1.1: Could a QAP/U-SP model be leveraged during the solution procedure to reduce computational times where the layout continuity property is not as imperative?

As concluded above, a more flexible modeling approach is required to sufficiently characterize the layout of this dissertation. For improved model flexibility and the ability to capture layout continuity, which the QAP formulations are incapable of achieving, researchers have turned to MIP model formulations of the problem.

2.2.2 The Mixed Integer Programming Formulation

It is commonly acknowledged that George Dantiz was the first to introduce the linear programming model in 1947 along with his simplex algorithm for solving said problem effectively [31,169]. Years later, Martin Beale and R.E. Small initiated the development of the first mixed-integer form of the LP/90/94 code, later completed by Max Shaw in the late 1960's [31]. The introduction of the MIP model has been instrumental in enabling researchers over the years to overcome the limitations stated before regarding the QAP model of the LP. The MIP formulation can overcome these limitations as it permits a mixed variable structure consisting of both continuous and discrete (binary or integer) variables. The model's ability to handle continuous variables in addition to discrete variables makes it an extremely versatile model and more importantly one that is capable of representing continuous layout designs. The generic mathematical formulation of the MIP model is as follows [15]:

$$\min_{x,y} f(x,y)$$

$$\begin{aligned} s.t. \quad & g(x,y) \leq 0 \\ & h(x,y) = 0 \\ & x \in \boldsymbol{X} \\ & y \in \boldsymbol{Y} \text{ integer} \end{aligned}$$

where *f* is the objective function of the problem (e.g. material handling cost), *g* are inequality constraints (e.g. budget constraints where the readjustment of an existing layout has an upper limit), *h* are equality constraints, *x* are continuous variables belonging to the domains defined in the vector ***X***, and *y* are discrete variables (integer or binary) with ranges defined in the vector ***Y***.

Depending on the linearity of the *f*, *g*, and *h* functions the MIP model can be further classified as either a mixed integer linear programming (MILP) or mixed integer nonlinear programming (MINLP) problem. The distinguishing characteristics between the two is that MINLP has nonlinearities in the objective function and/or constraints, whereas the MILP form of the problem does not [42,131]. As will be observed later while discussing solution methods, often the LP is either formulated directly as a MILP problem or the MINLP problem is linearized prior to solution. The MILP formulation of the problem can be effectively solved by linear programming as originally done by

Dantiz. This is the main motivation behind why researchers have handled the linear form of the problem more often. Regardless of the specific form solved, the MIP model is of NP-hard solution complexity just like that of the QAP model [7]. Generally speaking though, solving the MIP problem is more difficult. This is a byproduct of its mixed variable structure and constraints which often conflict with one another.

The MIP formulation of the LP was first introduced by Montreuil in 1990 [125]. Since then the MIP formulation of the LP has evolved steadily. This evolution has been driven primarily by the need for enhanced design capabilities and improved layout detail. Montreuil's formulation modeled a continuous rectangular layout composed of fixed area flexible departments. Montreuil, in his formulation, did not address the location of departmental I/O points and this is where Kim and Kim sought to extend the formulation. Kim and Kim proposed a MIP formulation for determining the best placement of the department I/O points for a provided department layout [91]. Although their formulation established the placement of the department I/O points it did not establish the department positions themselves in the layout. Barbosa-Póvoa, Mateus, and Novais (2000) proposed a more generic MIP formulation for the detailed LP that simultaneously considered the placement of the departments in the continuous layout and the placement of the I/O points within the rectangular departments [20]. In continuation of this work, Barbosa-Póvoa et al. (2001) extended their original MIP formulation to handle the presence of irregular shaped rectangular departments [21].

Barbosa-Póvoa et al.'s (2001) MIP formulation for the detailed LP, considers each of the layout characteristics essential to designing layouts of realistic viability. It accounts for boundary constraints, maintains overlap avoidance by imposing the appropriate constraints, can handle objects fixed in the layout space, and most importantly captures layout continuity. Capturing layout continuity by representing department placements continuously in the layout also enables said formation to directly account for flow feasibility throughout it. Considering that the formulation proposed by Barbosa-Póvoa et al. (2001) captures many of the most essential layout characteristics to ensuring designs of realistic viability are established, the following observation is made:

> **Observation 2:** A MIP formulation such as that employed by Barbosa-Póvoa et al. (2001) is a viable modeling approach for this problem, where viability is defined as the flexibility required to characterize the layout adequately.

With the first of the sub-questions regarding how to mathematically model a layout such that characteristics essential to establishing layout designs of realistic viability are captured, discussions can now turn towards addressing the second sub-question, **Research Question 1.2** about how evolution and uncertainty pertinent to the LP has been captured by researchers in the literature.

2.3 Capturing Evolution and Uncertainty in Conditions

As was well established before with **Assertion 2**, accounting for uncertainties in market conditions and business practices is imperative to designing layouts that will continue to perform well in the future. As aforementioned, there are two categories of manufacturing uncertainties, those associated with internal disturbances and those accompanying external forces, with the latter off these encompassing uncertainties in demand levels, product mixes, product profit margins, layout restructures, and equipment changes [100,152]. These uncertainties, which are interdependent, collectively address variations in both the market and business model conditions. As such, both must be accounted for in parallel to design a layout capable of remaining viable long term.

Demand levels and product profit margins (combination of manufacturing costs and selling value) can be grouped under the market conditions envelope while restructures, equipment changes, and product mixes can be placed under the business model envelope. Addressing the uncertain and evolving nature of each of these groups leads to two further characterizations of the layout problem. Often to capture the evolution of these conditions, researchers introduce the idea of a planning horizon. Depending on this horizon's structure, the layout problem can then become either a static or a dynamic problem. Furthermore, the problem can be characterized as one that is deterministic or stochastic. This characterization is often the result of researchers attempting to capture uncertainties in these conditions in order to design a robust layout. In the sub-sections to follow, a further exploration of these two problem characterizations and their roles in facilitating the infusion of evolution and uncertainty into the layout problem will be addressed.

2.3.1 The Role of the Planning Horizon

A planning horizon encompasses the duration of time considered while solving the layout problem. Depending on the length of this horizon, market conditions and business models can dramatically evolve rendering an otherwise well performing layout obsolete. To more effectively account for this evolution in conditions, a parallel evolution of the layout may be considered in an attempt to maintain operational performance. In the layout problem, researchers partition the planning horizon into periods in order to encourage this layout evolution. This partitioning of the planning horizon constitutes the differentiation between a static and a dynamic LP [127,100].

2.3.1.1 The Static Layout Problem

As observed by Benjaafar, Heragu, and Irani, relayout "can be expensive and disruptive, especially when factories must shut down" and production stopped. [26]. When such an evolution of the layout would become too costly, adopting a static layout approach may prove more advantageous. For the static layout problem (SLP), the planning horizon consists of just a single forecast period that spans the entire duration of the horizon. As such, the layout remains fixed over this entire horizon and the objective of the problem is to establish the best *layout* under these conditions. Given the layout does not evolve over the horizon, such a problem is often solely concerned with minimizing the total material handling cost (MHC). It does not need to be concerned with costs associated with layout rearrangement or losses in income due to production stoppage. A problem of this form is most appropriate for scenarios where market conditions and business models do not change dramatically over the horizon. When any one of these scenarios does not hold true, which is often the case, a different approach to the problem is required. This is where formulating the problem as a dynamic layout problem (DLP) becomes imperative.

2.3.1.2 The Dynamic Layout Problem

Rosenblatt was the first to introduce the DLP, which entertains that it may be desirable for the layout to evolve over time [146]. Unlike that of the SLP, who's planning horizon consist of just a single forecast period, the horizon of a DLP is partitioned into multiple periods in order to encourage the layout to evolve throughout. With the presence of now multiple periods, the DLP must consider a series of sequential forecast periods with the goal of establishing the best *layout plan* under these collective conditions of the horizon. When considering such a compilation of multiple layouts deployed over the planning horizon, the term *layout design* will be used to more broadly describe this collection, or plan, going forth in this dissertation. The DLP in many ways can be thought of as the simultaneous solution of a SLP for each period in the horizon [106]. The cost of layout evolution, or in other words rearrangement costs (RCs) from one period to the next, is what couples these SLPs and requires them to be solved simultaneously as opposed to independently. As one can imagine, this further complicates the solution of the problem.

As observed before and restated here, Lacksonen and Enscore establish that the DLP is "required when one must balance the tradeoff between increased flow of inefficient layouts and added rearrangement costs" [106]. While layout performance for a period is only a function of its characteristics, rearrangement costs are dependent upon both the preceding and succeeding period layouts. This is because any alteration to the layout in the current period would then result in a change in how much the layout need be rearranged to achieve the current layout from that of the previous one and likewise to achieve the succeeding period layout from that of the current one. Under the DLP formulation, if the decrease in the layout performance (i.e. increase in MHC) does not outweigh the cost of this rearrangement, the layout will remain unaltered. On the other hand, if conditions change enough to result in an increase in MHCs that surpasses the cost of rearrangement to maintain a better performing layout, the layout may undergo this evolution. The magnitude of this evolution again will depend on the tradeoff between improved performance and the costs to achieve this improvement. In the upper limit where a consistent evolution is beneficial, each period of the multi-period planning horizon could potentially yield a unique layout.

Rearrangement of the layout requires an upfront capital investment however. The observation of this requirement can further prevent the layout from evolving from one period to the next despite a rearrangement being favorable. If at the time, insufficient funds are available to perform said rearrangement, the layout will again remain unaltered. The decrease in the layout performance will then have to be accepted and the diminished profit margin absorbed until the next period rolls around where sufficient funds for rearrangement are available. More likely, a partial evolution of the layout, which maximizes the improvement in the layout performance for the funds available at the time, would occur. This capital investment restriction and the affect it has on the evolution of the layout in the DLP has been captured by only a few researchers, including that of Balakrishnan et al. (1992) and Conway and Venkataraman, through the implementation of budget constraints [17,48]. These constraints represent the funds available (e.g. from profits reinvested in the company or capital raised from stock offerings) to restructure the layout in order to maintain operational performance and therefore sustain competitive profit margins. Conway and Venkataramanan's formulation extends Balakrishnan et al.'s (1992) original budget constrained DLP formulation by accounting for scenarios where left over budget resources from preceding periods may be used to further facilitate the evolution of the layout [48].

2.3.1.3 The Benefit of Formulating the Problem as a Dynamic Layout Problem

One of the major advantages of the DLP is that it can handle scenarios involving the integration of a new asset into the environment, whereas a SLP cannot. The integration of a new asset inherently requires the environment to be restructured, something the SLP cannot consider as the layout remains fixed throughout the planning horizon. Furthermore, in the absence of changes in market conditions and the business model, the DLP reduces to just that of the SLP where the layout remains unchanged throughout the horizon. In other words, the DLP formulation has the capability of representing both a static and a dynamic problem simultaneously making it a more versatile approach to the LP. Note that the terms *environment* and *layout* are not used interchangeably. While layout refers to the physical configuration of the objects in the space, environment more broadly encapsulates this as well as other characteristics of the referred period (e.g. process present, market conditions, etc.). This subtle distinction is important to understand from here forth in this dissertation. Now in consideration of the preceding discussions, the following assertion can be made.

> **Assertion 5:** To design a layout that will continue to perform well in the presence of evolving conditions, the LP must be structured as a DLP.

2.3.2 Defining a Planning Horizon for the DLP

So far it has been established that in order to design a layout that will continue to perform well in the presence of evolving market conditions and business models, a layout design that evolves according to a planning horizon consisting of several periods is imperative. What has yet to be established is the structure of the periods composing the planning horizon and what factors can influence this structure.

2.3.2.1 What Drives the Structuring of the Periods?

In general, market conditions directly drive the evolution of the layout design and therefore the planning period structure. The only exception to this is where a new asset is integrated into the environment. In order to accommodate said asset the layout inherently needs to be rearranged. With that said, market conditions still indirectly drive this business decision. Observations of market trends may indicate the integration of the asset would provide a fiscal benefit to the company. For example, said asset may enable the company to enter a profitable sector of the market (i.e. one that is either growing or marginally competitive). To fully understand how the conditions of the market drive this evolution, their influence on layout performance must first be observed.

2.3.2.2 The Impact of Market Conditions on Layout Performance

The performance of a layout changes under three scenarios. The first relates to the earlier scenario involving the integration of a new asset into the environment. In addition to the requirement that the layout undergo a rearrangement to accommodate said asset, its integration also requires that the business alter its product mixes. With the new asset, additional products can be manufactured and furthermore new process flows become relevant, all of which will affect the layout's performance.

The second is a byproduct of demand levels for products changing. Once more, as demand levels change, product mix ratios change and thus the performance of the layout will likely diminish as a result. In response to this, a business may consider redistributing their production loads (product mixes) to mitigate this impact. Regardless of this, the performance of the layout remains suboptimal without restructuring.

The third scenario relates to changes in costs to manufacture products (MHC for various processes, labor costs, etc...) and market values of products. Just like the previous two scenarios this one may encourage a business decision to be implemented that involves altering the product mix in order to adjust to these changes. For example, if a product is no longer yielding a reasonable profit (decrease in market value or increase in cost to manufacturer) while another product's profit margin has grown, it may become advantageous to alter the production distribution accordingly. This redistribution of the production would in turn place a larger emphasis on a product line that otherwise didn't have as much emphasis before. As a result, MHCs will increase, leading to a poorer performing layout. If the business does not respond to this change, the performance of the layout still changes. For example, if a product with a longer flow path experiences a large increase in the cost to manufacture (i.e. increase in MHC) while also experiencing a decrease in its market value, the performance of the layout will degrade.

According to the preceding discussion of potential scenarios affecting layout performance, the evolution of the layout can be beneficial under scenarios where business decisions related to the integration of a new asset and/or alteration to production distributions (i.e. product mixes) are implemented. Furthermore, changes in market conditions (i.e. demand levels, costs of manufacturing, and product market values) in the absence of a decision to alter the business model accordingly can too warrant a favorable evolution of the layout. Therefore, planning periods in theory should be structured such that partitions between periods align with such occurrences. Establishing where these period partitions should fall is an arduous task however, especially in the presence of uncertainty. Often to avoid solving this sub-problem of the DLP, researchers have relied on broad assumptions to make establishing the planning period structure more manageable.

2.3.2.3 Conventional Approach to Planning Period Structuring in the Literature

One assumption often implemented by researchers is to define the planning periods according to uniform durations of time. Although there is no restriction on the lengths of these periods, a year in length is often chosen. The justification of this assumption is likely that it aligns with the common business practice of performing a business plan review annually. With these reviews reconsidering current business strategies based on past performance and addressing future strategies according to market forecasts, it is reasonable to assume that at each of these reviews, business decisions regarding product mixes and asset integrations would be made. As observed before, alterations to product mixes to account for changes in market conditions and asset integrations drive the evolution of the layout. Therefore, aligning the period partitions according to these annual reviews is a reasonable assumption.

Unfortunately, this subsequently assumes that all such business decisions are implemented at the onset of this period, including layout rearrangements. None of these assumptions alone are all that unreasonable, except when coupled together. Assuming that layout rearrangement is performed at the beginning of the period and that these periods are uniformly spaced to be a year in length is extremely limiting. Additionally, it is very likely that the best solution to the problem will not be achieved. For example, it may be more advantageous to restructure mid-year during a stint of low production demand. Several situations can make this so including, but not limited to, production demands being too high at the beginning of the year to stop production, presence of a less favorable budget available then, or a marginal change in conditions that do not warrant such a restructuring.

These assumptions coupled together result in uniform and rigid restructuring schedules which are not well representative of actual ones. To avoid this limitation, some researchers such as Yang and Peters have considering first solving the period structuring problem before proceeding to then solve the DLP associated with this established planning horizon [173]. Yang and Peters implemented a heuristic procedure for establishing the best period lengths for a given planning horizon. To solve the sub-problem of identifying the best set of planning periods to implement, Yang and Peters employed a procedure motivated by the Silver-Meal lot-sizing heuristic developed by Silver and Peterson [173,153]. Solving this problem first not only enables the best restructuring schedule to be identified, but also has the advantage of reducing the dimensionality of the DLP that then needs to be subsequently solved. For example, market conditions may dictate that only two restructures would be beneficial during a five-year planning horizon. In this case, a DLP of just three periods would need to be solved as opposed to one of five periods if the earlier annual period structuring was adopted. As will be observed later, this has a significant impact on the solution time.

Although this approach avoids the drawbacks accompanying uniform and rigid planning period structures and is capable of more effectively identifying the best evolution of the layout, it lacks the necessary transparency required to make informed decisions regarding such evolutions. This is because with the approach identifying the best planning period structure internal to the solution of the DLP itself, the designer cannot observe potential alternative schedules nor the schedule's sensitivity to further fluctuations in market conditions. This is a crucial gap that must be closed. As such, the following assertion can be established:

> **Assertion 6:** An approach to establishing the planning period structure of the horizon that provides the necessary transparency to make more informed decisions regarding the design of evolving layout designs is required.

2.3.3 Designing for Robustness

The preceding discussion observed how the evolution of these aforementioned conditions can be captured, but not the uncertainty associated with them. These conditions and their evolution can be highly uncertain and the further into the future they are forecasted, the less certain they will become. To account for this uncertainty, researchers often rely on a stochastic approach to the problem. Under this characterization, the behavior of the product mix, product demands, manufacturing costs, and product market values are assumed to be stochastic in nature [100,127]. In the absence, or more often neglection, of uncertainty these conditions are instead assumed to be deterministic and known with certainty across the span of the planning horizon. In other words, a single evolutionary path of the conditions dictates the design of the environment. This is a rather risky approach to designing an environment for the future. If the conditions deviate from this path, the performance of the designed environment could degrade substantially.

To mitigate this risk, the idea of designing an environment for robustness is proposed. Generically, a robust environment or layout is one that behaves well over a variety of scenarios, or in other word a series of evolutionary paths of the conditions.

Others such as Rosenblatt and Lee and Kouvelis, Kurawarwala, and Gutierrez defined robustness more thoroughly as the frequency the layout falls within a prescribed percentage of the optimal solution for different sets of product demands [147,99]. Generalizing this to all such conditions more adequately defines a robust manufacturing layout design.

Solving the problem according to this approach characterizes it as a robust layout problem (RLP). It should be understood that the RLP differs from another stochastic variant, referred to as the flexible layout problem. This problem attempts to establish a layout that can most readily adapt to changes without significant loss in performance. This is achieved by considering multiple routings of process flows [100]. As observed before in the introduction and in Figure 2, research this century has shifted towards solving the RLP and for good reason. Conditions are difficult to predict and highly uncertain, therefore solving the RLP has the key advantage of establishing a layout that will continue to perform adequately regardless of the conditions it is subjected to. Such an approach reduces the risk associated with designing a layout for the future making it an attractive option for companies.

2.3.3.1 Establishing Robustness

Defining the set of conditions, or condition scenarios that will establish the robustness of a layout is an important task. All potential scenarios that could be encountered by the layout need to be spanned by this set to ensure the layout will remain robust under all conditions of potential relevance. Such a scenario-based approach to the problem assesses the robustness of a layout according to this predefined set of discrete scenario representations of the conditions. In other words, these scenarios collectively represent the uncertainty associated with the conditions. The layout that collectively performs best across these scenarios is then deemed as the most robust layout. Researchers like that of Kouvelis et al. implemented such a scenario based approach to solve the QAP formulated DLP under product mix and demand uncertainty [99].

Others like that of Pillai, Hunagund, and Krishnan have established robust layouts considering just an average scenario for this set [139]. Pillai et al. assumed that the average product demands, mix, and other conditions were known throughout the horizon for this average scenario. To compare their adaptive DLP approach to defining a robust layout (i.e. one that remains fixed across all periods of the horizon) to traditional DLP solutions and test the suitability of the layout's robustness, Braglia, Simone, and Zavanella's Total Penalty Cost (TPC) function was applied posterior. This function establishes the minimum re-layout cost acceptable to support an agile strategy, where a layout with a TPC under fifteen percent is deemed acceptable [36]. Results of their approach proved effective in establishing a layout with acceptable robustness when compared to the DLP solutions to the same scenario set, which was chosen to be Balakrishnan and Cheng's (2000) 48 DLP test problems.

An alternative to the scenario-based approach is the statistical modeling approach. The statistical modeling approach establishes a robust layout in a more generalized manner. Instead of relying on a discrete set of scenarios to represent the uncertainty, this model assumes that the conditions are random variables with known distribution parameters (expected value and variance) [100,127]. Forghani, Mohammadi, and Ghezavati have proposed such an approach where it is assumed that only uncertainty in the demand is present [69]. Using an approach developed by Sim, they solved the discrete optimization SLP subject to uncertain demands. In Sim's formulation, a decision variable, affecting the range of scenarios or demand deviations from the nominal value considered, is implemented to provide the designer with the added ability to adjust the degree of conservatism of the robust solution [69].

Norman and Smith, to establish a robust block layout subject to production uncertainty, considered more directly product forecast uncertainty using normal distributions with expected values and standard deviations [132]. This representation of the uncertainty has the major advantage of continuously assessing a range of production scenarios as opposed to a subjectively defined set of them as is done in the scenario based approach. Furthermore, their approach accounts for individual product variability and thus indirectly the product mix. Individual representation of the product variabilities enables well-established products with low forecast uncertainties to be differentiated from those of new products that may have higher uncertainty. Using a statistical percentile approach to the definition of the MHC objective function enables Norman and Smith to effectively identify robust layout designs and furthermore demonstrate the layout design can change dramatically depending on the level of uncertainty considered by the designer.

Norman and Smith's observation of how significantly the layout can change given the presence of varying degrees of uncertainty in the conditions coupled with the almost certain likelihood of this uncertainty being prevalent in real-life applications, the following statement can be made:

Assertion 7: To design a layout that will continue to perform well in the presence of uncertain conditions, the LP must be solved stochastically for robustness.

2.4 Establishing the Performance of a Layout Design

With the characterization of the layout problem having since been well established in the preceding sections, the focus of this section is on answering the question of what defines a quality or well-performing layout, in other words **Research Question 1.3**. As observed before, changes in layout performance encourage its evolution. Therefore, establishing what constitutes a layout as being well performing is important. Just as importantly, are the methods implemented to model the qualities that define this layout performance. Until now, a generic perspective on the LP has been entertained. Going forward though, a focus on what establishes a manufacturing layout in particular as being well performing will be discussed.

Most often in the literature, researchers have defined a quality manufacturing layout design as one that does well in managing the MHCs and/or rearrangement costs (RCs). For SLPs and RSLPs where the layout remains fixed throughout the planning horizon, the primary objective is to minimize the MHCs. The MHCs is of such importance as it represents 20-50% of operating cost and 15-70% of the total cost of manufacturing a product [72]. When the layout design is allowed to evolve throughout the planning horizon, as is the case for DLPs and RDLPs, RCs must too be considered in addition to the MHCs. For such problems the primary objective is then to minimize the sum of the MHCs and RCs across the entire horizon. The best solution then becomes the one that best balances these two manufacturing costs.

2.4.1 Rearrangement Cost

RCs account for all costs associated with the rearrangement of assets on the floor from one period to the next. This can include the physical movement of the asset, rerouting of necessary conduit, and profit loss due to production stoppage of affected processes. A variable horizon cost approach is the standard across the literature and establishes that the RCs can change throughout the planning horizon or in other words from one period to the next, which would often be the case in real-life. How these costs change within a planning period can either be discrete or continuous. The former is the preferred method as it aligns well with the assumption noted before of restructuring the layout at the onset of a period. Methods of defining these variable horizon RCs in the literature are further classified as either constant or distance based methods [12]. This distinguishes RC methods that rely on a discrete modeling of the cost functions from those that use a continuous model.

A discrete or constant method assumes, independent of how far an asset is relocated from its previous location, that the costs of rearrangement will remain the same. This method has the advantage of being relatively simple to implement and as such is the method most often employed by researchers in the literature while solving the QAP/S formulated DLP [23,121,148]. The discrete nature of this problem formulation makes this a reasonable method, which is why it is so often implemented. The method's inability to accurately establish differences between switching neighboring assets with that of assets at opposite ends of the layout space is its major drawback however. This drawback results in a poor representation of the RCs while handling a MIP formulation of the DLP, where assets are continuously located throughout the space. To overcome this drawback and provide an improved representation of the actual RCs, a distance based continuous model is generally implemented.

A continuous or distance-based method no longer assumes that the RCs are independent of how far an asset is moved. Instead, the RCs become a function of this distance moved in the continuous space from one period to the next. This representation provides a much better depiction of real-life RCs with only a marginal increase in additional computational overhead. This overhead is the result of having to compute the move distance, which is often determined according to a Euclidean or rectilinear approach in the literature. While handling the continuous layout MIP formulation of the DLP, this is the more accurate approach to defining the RCs. Given that the MIP formulation of the DLP has been less entertained in the literature, so too has the distance-based method of defining the RCs.

As observed, the aforementioned distance-based method represents costs directly associated with the movement of the asset. This includes both the physical movement of the asset and any necessary rerouting of conduit to properly support the asset (power, exhaust, etc…). It does not however encapsulate the loss of production cost due to the interruption of processes associated with moved assets. This cost is independent of distance and as such is often handled separately and according to a fixed cost method [118]. The value for this cost can be established by accounting for production volumes (via the product mix at the time of rearrangement) and the profit margins of the products (via cost to manufacture and market value of products) yielded by said interrupted processes. Given the preceding discussion on methods of defining RCs, the following assertion can be made:

Assertion 8: To most realistically define the rearrangement costs, a distance-based variable horizon cost approach will be required. The cost of asset movement and support conduit rerouting must be a function of the distance each asset is moved in the space. Furthermore, a constant cost method for establishing the loss of production cost will be required to completely define the costs associated with rearrangement.

2.4.2 Material Handling Cost

In addition to RCs, costs associated with the flow of material throughout the environment must also be accounted for. This cost is often referred to as the material handling cost (MHC). As observed earlier, MHCs play a major role in establishing the performance of a layout design. Therefore, the method implemented to represent said costs can have a substantial impact on establishing which layout design is considered best. To ensure the best realistic design is deemed superior, the method to represent the MHCs should be chosen wisely. MHC methods are composed of two components, the unit cost of handling the product per unit length (flow cost) and the length of distance the product is handled (flow distance).

2.4.2.1 Defining Flow Costs

Flow costs are product-dependent and furthermore stage dependent, where the latter is in reference to the stage in the manufacturing process the product is in (i.e. between which two assets the product is being handled). With this structure, a flow cost matrix can be formed with each row representing a specific product and each element in this row representing the product's flow cost for a specific flow segment in its manufacturing process, or in other words a segment between two sequential assets in its process.

Furthermore, these process segment flow costs can change with time. As such, a series of flow cost matrices, one for each period in the horizon, can be constructed. In most applications in the literature the flow costs are assumed to be discrete across the entire period, despite this being less than representative of reality. In this case the process segment flow costs in each period would remain fixed. Few have considered the more accurate representation, by implementing a continuous representation of the process segment flow costs within each period [132]. Such an approach enables cyclic yearly fluctuations among others that may occur within a period to be more accurately captured. In turn, this approach can enable more realistic MHCs to be defined.

2.4.2.2 Determining Flow Distances

To completely define MHCs, flow costs must be coupled with flow distances. Just like that of the flow costs, flow distance matrices can be formed for each period. Since flow distances are just a function of the physical layout and a layout remains fixed within a period, the flow distance matrices will only ever be discrete across the entire period. To establish the process segment flow distances in these matrices, rectilinear or Euclidian distance methods are often employed in the literature. A rectilinear method defines the flow distance as the summation of the absolute differences in the geometrical coordinates between the start and end point of each segment. The Euclidean method on the other hand defines the flow distance as the linear distance between the start and end point of each segment.

Both methods have the major advantage of maintaining a continuous and linear cost function. With such a cost function, the DLP can be made linear and thus partially solvable by linear programming. As will be observed later, this is essential to the efficient solution of MIP formulated problems. In addition to maintaining a linear and continuous function, rectilinear and Euclidean methods can populate the flow distance matrices rapidly. The combination of these two advantages, coupled with batch production environments typically being of upmost concern by researchers, has resulted in their frequent implementation in the literature to define the flow distances. The major disadvantage of such methods however, is their inability to ensure flow feasibility. Such methods ignore assets boundaries often providing paths that bisect assets or other internal layout boundaries. Not only is this not well representative of reality, but it also can lead to suboptimal solutions for variable production environments where several interrelated processes are occurring concurrently.

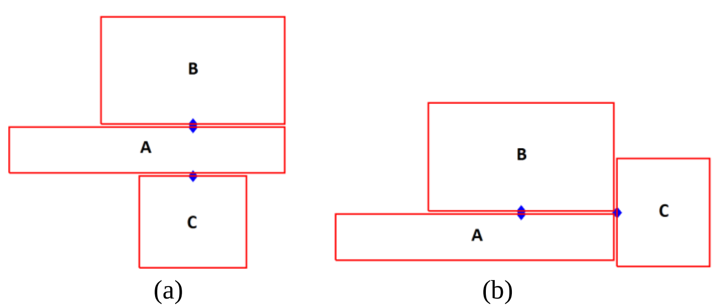


Figure 4 – Process flows from B-A and C-A (a) optimal solution using a rectilinear approach (b) better solution when path feasibility is considered

To demonstrate this, consider the basic example as shown in Figure 4 of such an environment where two processes, B to A and C to A, have a common station (station A). Figure 4a provides what would be identified as the optimal solution for a rectilinear or

Euclidean method whereas (b) considers path feasibility. If path feasibility was enforced for the 4a configuration the path would have to go around the right end of station A in order to connect the two I/O points, represented by the blue diamonds. This results in an overall flow distance that is now longer than that of the (b) layout making it a suboptimal solution. As such, the following observation is verified:

> **Observation 3:** Failure to account for flow feasibility can result in the identification of suboptimal layout designs.

Additionally, this example enables the following assertion to be made:

> **Assertion 9:** To accurately evaluate and subsequently design variable production environments with several concurrent interrelated processes, a distance-based method that considers flow path feasibility is imperative.

The few researchers in the literature, who have considered such flow path feasibility while determining the flow distances, have implemented Dijkstra's algorithm. One such example is Lee, Roh, and Jeong's use of it while solving the QAP formulated multi-floor SLP and applied specifically to the multi-deck ship design problem [108]. Although such a flow distance method ensures flow path feasibility thereby avoiding the limitation noted before with regards to rectilinear and Euclidean methods, it presents its own disadvantages. Firstly, it requires the a priori definition of the graph nodes. In the above noted example, structured aisles/passages about the ship were already well defined making it simple to assign such nodes in the space. For unstructured layouts, where aisles/passages are to be a derived layout characteristic, the a priori definition of the graph nodes is an impossible task. In addition to this, population of the flow distance matrices is significantly more time consuming. Determining each flow distance requires an exhaustive solution procedure to guarantee the best feasible flow solution is identified. Furthermore, such a method no longer maintains a linear and continuous cost function; rather the function is more likely to be non-smooth and discontinuous. As one can postulate, this only further contributes to the difficulty of solving the overarching LP.

2.4.2.3 Importance of Considering Path Feasibility in Layout Design

Despite the less than favorable consequences that accompany implementing a flow path feasibility guaranteeing method such as Dijkstra's algorithm, neglecting to account for such feasibility, as is done by rectilinear and Euclidean methods, is potentially disastrous. As was observed before, failure to account for flow feasibility can lead to suboptimal layout designs. An inferior performing layout design is highly detrimental to a manufacturer as it can substantially reduce their potential profit margin as observed before. Based on observations this could be up to 0.18M for a company with current operating costs of 1.2M dollars. This number is derived from the earlier observations that MHCs constitute up to fifty percent of the operating cost and additionally with effective layout planning, MHCs can be reduced anywhere from ten to thirty percent [72]. With less capital derived from profits available to reinvest in the company, this can subsequently diminish the business' growth rate and adaptability to unforeseen future events (e.g. the need to purchase a new asset and/or restructure the layout). Moreover, this can contribute to the business becoming less competitive in the market, which, as observed before, is imperative to success.

2.4.2.4 Effectively Considering Path Feasibility

Of the shortcomings that Dijkstra's algorithm has in determining the flow distances, its requirement to a priori define the graph nodes is especially troublesome when considering the problem that must be solved in this dissertation. With the expectation that an unstructured approach to the layout design will inherently yield the desirable flow aisles/passages, defining the graph nodes a priori becomes impossible. Furthermore, and more generally, the computational overhead associated with guaranteeing flow feasibility is unavoidable, but manageable to some extent. An efficient method of determining the feasible flow distances can minimize this overhead. Unfortunately, the failure to maintain both continuity and linearity in the cost function is an unavoidable outcome of ensuring flow feasibility. Therefore, regardless of the method implemented to do so, this property will have to be accepted as an unavoidable outcome of the more realistic flow distances obtained.

The requirement to account for flow feasibility coupled with the incompatibility of Dijkstra's algorithm to the problem of this dissertation establishes a noteworthy gap in the literature. The following assertion can then be derived:

> **Assertion 10:** A more robust alternative to the Dijkstra's algorithm that ensures flow feasibility, yet is not as limited by its shortcomings, is required to effectively provide realistic flow distances for defining the MHCs.

Subsequently, the following question also arises:

> **Question 1.3.1:** How can unstructured layouts that accurately consider flow path feasibility be evaluated in a reasonable duration of time?

2.4.2.5 A Novel Flow Distance Method

Fortunately, preliminary research performed by the author has addressed this very gap in the literature [123]. In this research a more robust flow feasibility guaranteeing alternative to Dijkstra's algorithm has been proposed. Furthermore, it directly addresses the aforementioned question of how unstructured layout flow distances can be obtained effectively.

In this research, an advanced flow distance method leveraging a branch and bound optimization technique tailored to the problem, which effectively identifies the shortest feasible process segment flow distances, was proposed. Infeasible region bisections (assets, internal building structures, safety zones, etc…) by the parent segment path were used to identify potential branches while the maximum and minimum of these violations were used as the splitting procedure. The algorithm's pruning step was implemented by comparing the best discovered feasible segment path to each of the potential paths. Once each of the process segment flow distances were determined, the flow distance matrices became fully populated and the MHCs were then defined.

This advanced flow distance method, which guarantees flow feasibility when defining the flow distances, demonstrated that it could effectively handle identifying such flow distances for unstructured layout designs while mitigating the computational overhead of ensuring flow path feasibility. As demonstrated in Figure 5, the layouts generated using both a rectilinear and the advanced method to define the flow distances are starkly different. Even for a very basic environment such as this, which is composed of a single fixed station, five movable stations, and just two interrelated process flows equally weighted in terms of importance (i.e. a 50/50 product mix), the differences in the layouts are noticeable.

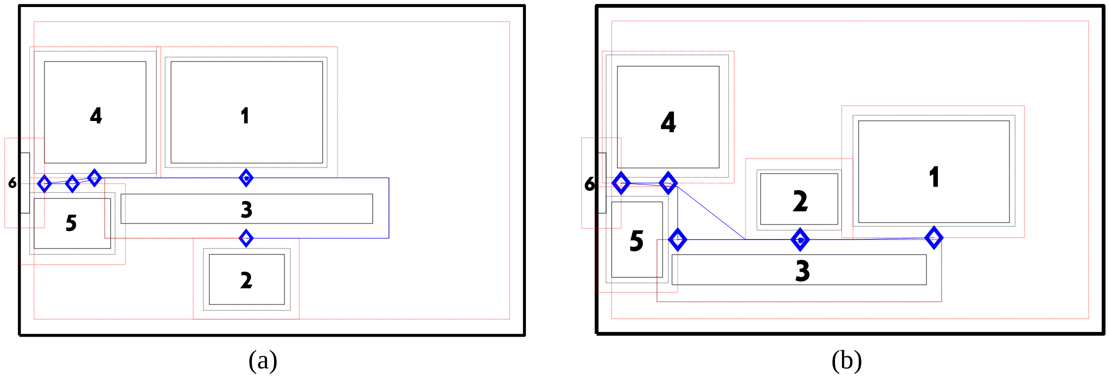


Figure 5 – Optimized layout using (a) a rectilinear approach and (b) the proposed advanced approach to evaluate the handling cost

In addition to the clear visual difference, a difference in the total flow distance is also present. As shown in Table 3, when the rectilinear method is implemented to generate the solution, the result is a suboptimal layout when considering flow feasibility. In fact, a nearly 35 percent less optimal layout is achieved, which is quite significant. Additional studies identified scenarios where layout designs up to 230% less optimal resulted from the use of a non-flow feasibility guaranteeing method to generate the flow distances.

Table 3 – Material handling cost results

Approach	$f_{optimizer}$	$f_{rectilinear}$	$f_{feasible}$
Rectlinear	47.02	---	74.09
Advanced	55.42	59.86	55.42
		Difference	34%

These results demonstrate the sheer importance of using a flow feasibility guaranteeing method such as the proposed advanced method implemented in this research. The advanced method further demonstrated its capability of guaranteeing this flow feasibility in a reasonable duration of time and for unstructured layouts. As observed here the following hypothesis can then be stated:

> **Hypothesis 1:** **<u>If</u>** an advanced flow distance method that ensures flow feasibility is implemented to define the MHCs, **<u>then</u>** improved layout designs that are better representative of reality can be established for variable production environments where several interrelated processes are occurring concurrently.

2.4.3 Other Measures of Performance

Although MHCs and RCs have been the primary focus of researchers solving the DLP in the literature, and justifiably so given their considerable contribution to layout performance, it is useful to acknowledge that other measures of layout performance have too been considered in the literature. Other, both quantitative as well as qualitative, measures of performance have been addressed.

Lin and Sharp provide a brief overview of both types in the plant layout problem [112]. Qualitatively, Lin and Sharp observe several criteria, which they categorize as either surrounding or environment quality related concerns of the designer. Under the environment quality group, these concerns range from HVAC quality and ergonomics to the complexity and compatibility of material handling equipment. Quantitatively, they discuss measures ranging from clearness, which addresses how clear a layout is of fixed or permanent building obstructions to aisle arrangement, which addresses how effectively the aisle placements promote the flow of material and personnel throughout the space. Not surprisingly, Lin and Sharp also observe the MHC measure as one of the quantitative measures. Other frequently encountered quantitative measures of layout performance include flexibility, spatial utilization, and work-in-progress. These three measures are reviewed in more detail in Appendix A.

2.4.3.1 Concluding Remarks on Other Measures of Layout Performance

Although quantitative measures, such as flexibility, spatial utilization, and work-in-progress (WIP), along with qualitative measures, such as those observed by Lin and Sharp and others in the literature, are important and should, without question, be considered in addition to MHCs and RCs, these measures are often better evaluated manually by the designer. Additionally, apart from WIP, these measures are extremely difficult to quantify in a cost function format like that of the MHCs and RCs. One of the primary goals of this dissertation is to quantitatively evaluate the layout design on a monetary basis such that business decisions can be analyzed in a management friendly way. Thus, the focus going forth remains on that of assessing layout performance according to the conventional method of considering the combination of MHCs and RCs. With that said, this is not to say that a designer could not a-posteriori apply such measures to further evaluate a potential layout design. In fact, this could easily become an extension of the work performed in this dissertation if desired. As for WIP, the eventual formulation of the problem in this dissertation will render measuring WIP mute as a products throughput will be characterized by the most constraining time value (travel or process time) in its flow path. Thus, inclusion of WIP will not be considered as part of the cost function at this time, though it could be in the future if desired.

2.4.4 Methods of Computing Production Costs

The measures of MHCs and RCs deal with quantifying the indirect costs associated with production. They do not however account for direct costs of production or in other words, those costs that are incurred to alter the physical form of the product. Up until now, such costs have, for the most part, been overlooked. The goal of this dissertation is to enable more informed decisions to be made regarding the design of a layout and moreover to observe how the design impacts decisions pertaining to product introduction and mix. To accomplish this goal, completely defining the cost function becomes imperative. Therefore, the discussion that follows, attempts to elaborate on how direct production costs and manufacturing costs in general are modelled in the literature. The goal here is to gain a high-level understanding of how such costs could be estimated when eventually establishing the complete cost function that will be employed in this dissertation to evaluate a given design's performance.

A variety of cost estimation methods have been implemented in the literature [34, 83,96,73,145,111,164,133,93] and commercially [35,51,3,25,4]. As observed by Layer et. al. in their review of trends in cost estimation, these methods can be grouped into three distinct quantitative cost estimation categories. These include those that employ statistical models, generative-analytical models, or alternatively, analogous models to estimate costs. While statistical models rely on historical data and data analytics to establish costs, generative-analytical models leverage a more analytical, physics-based approach. Contrary to these, analogous models leverage functional and physical commonalities to establish costs from similar known cost structures (i.e. similarly known part and/or feature costs) [107].

Each of these approaches has its own advantages and similarly drawbacks. For example, while generative-analytical models can provide extensive cost granularity, they require an immense degree of detailed data and information to achieve this [107]. Ashby et. al. depicted such a generative-analytical cost model in their textbook. To generate cost estimations the model required knowledge of material costs, basic overheads, capital write-off times, load factors, dedicated tool costs, capital costs, batch rates, and furthermore tool lives for each process present in the environment [10]. As one can imagine, collecting and defining all this data would be an arduous task, especially given how detailed and extensive a list it is. Furthermore, establishing values for data such as "dedicated tool costs" is likely to be an estimation itself. This then raises the question of accuracy and subjectivity of the estimation. An over-optimistic estimation of its cost could then lead to inaccurate conclusions downstream by the designer and/or management. Despite these cautionary observations, many in the literature have still implemented such generative-analytical models to estimate costs with success [102,68,92,93,133,134].

Statistical models are a viable alternative to generative-analytical models as they can maintain the necessary level of cost granularity while avoiding the requirement for such detailed input data. As such, and as observed by Layer et. al. [107], several statistical model-based approaches and computer-aided engineering (CAE) techniques have been implemented in the literature to estimate costs [138,82,34,24]. Of the statistical cost estimation models identified by Layer et. al. [107] and discussed by Boehm et. al. [35], SEER from Galorath Inc. [51] could be an attractive option for estimating the manufacturing costs in this research. SEER is a widely trusted, commercially available tool, backed by over two decades of research and experience, for estimating manufacturing costs [107, 51]. Galorath Inc. has an array of products which offer a range of cost estimation capabilities to its customers. SEER products rely on parametric cost functions, which are continually updated and based on historical data from various enterprises and empirical examinations, to estimate manufacturing costs.

One of these products is SEER for Hardware, Electronics, and Systems (SEER-H) [50]. SEER-H is a comprehensive weight-based cost estimation tool for mechanical, electrical, electronic, structural, and hydraulic hardware project applications [76]. Cost estimations of development and production (indirect and direct) on the system, subsystem, and system of systems level are provided along with operation, support, and life-cycle costs. Like Galorath's other cost estimation products, SEER-H leverages "sector-specific mathematical models derived from extensive project histories, behavioural models, and metrics" as well as knowledge bases to provide these aforementioned costs from a user provided product weight [51]. While an attractive cost estimation option, SEER-H's generalized weight-based models are better suited for rough estimations of the product costs as opposed to more refined estimations.

A process-based cost estimation alternative to SEER-H's weight-based estimation is another Galorath product titled SEER for Manufacturing (SEER-MFG), formerly SEER-DFM [50,77]. Unlike that of SEER-H, SEER-MFG provides just direct production costs. Being process-based, SEER-MFG can provide far greater cost granularity and accuracy compared to SEER-H, which makes it not only an attractive, but also very viable direct production cost estimation option going forth in this research. Like that of the generative-analytical models discussed earlier, SEER-MFG does however require a far greater number of inputs, compared to SEER-H (weight and size predominately), to generate cost estimations. SEER-MFG requires detailed information regarding the product (beyond that of just weight and size) and the processes involved in manufacturing the product. The benefit gained from using SEER-MFG, over say SEER-H, is directly proportional to how detailed and accurate the designer is in defining the process models. The more detailed the model is, the more accurate the cost estimations will be, assuming of course these details remain accurate. In total, SEER-MFG has over seventy-five manufacturing process modelled. An obvious limitation to statistical models, like that of SEER-MFG, are their inability to evaluate costs for manufacturing processes that are unknown and for which data is unavailable. For novel manufacturing processes or in the case of SEER-MFG, for processes outside of the seventy-five where this is the case, one would need to resort to an analytical approach to categorize the costs for these process [33].

This concludes the brief review of cost estimation methods. For a more thorough review of cost estimation methods implemented in the literature and developed commercially, one may refer to Layer et. al. [107] and Boehm et. al. [35], where the latter focuses on providing a comprehensive review of software cost estimations models and techniques. At this point, a discussion of the dominant indirect costs (MHCs and RCs), other measures of performance (e.g. flexibility), and direct production cost methods have been reviewed. All bases for evaluating a layout design have been covered and as such the conversation now turns toward discussing the difficulty of solving the required problem formulation of this dissertation.

2.5 Compounding Difficulty of the Problem

Preceding discussions have established the need to address a MIP formulation of the stochastic robust dynamic layout problem (RDLP). Such a problem formulation is the most versatile of the LP formulations. The added capabilities and accuracy it provides comes at the cost of the problem becomingly considerably more intractable to solve. Figure 6 provides a characterization of the LP as discussed in length before and notionally demonstrates the relative difficulties between different definitions of the problem at each level of its characterization.

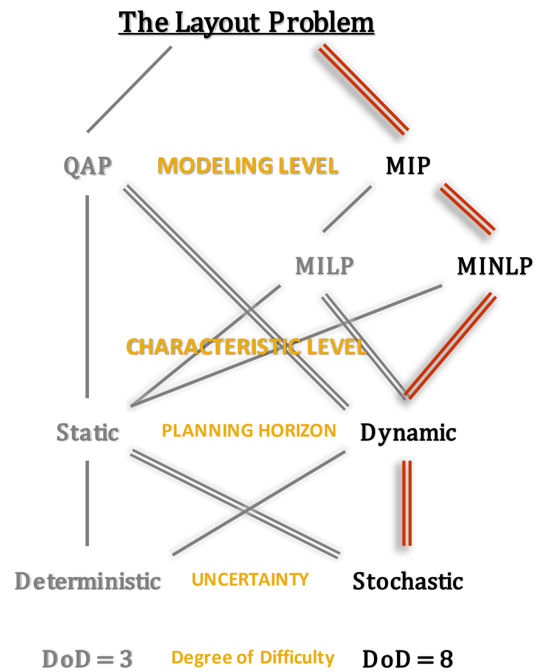


Figure 6 – Characterizations of the layout problem

Regardless of the formulation used, the LP is NP-hard, which identifies that as the size of the problem increases, the time it takes to solve the problem to optimality increases exponentially [154,103]. As observed before though, solving a MIP formulation of the LP is significantly more difficult than that of a QAP formulated one. This is especially true when said MIP formulation cannot be linearized and solved by linear programming techniques. The inability to linearize the problem was also established earlier when it was identified that a flow distance method that ensures flow feasibility was not only required to provide realistic layout designs, but would also result in a MHC function that was likely to be non-smooth and discontinuous.

Furthermore, it was observed that such a flow feasibility method would be considerably more computationally extensive than other more rudimentary methods. In fact, this guarantee requires a form of the traveling salesman problem to be solved for each process flow segment. The traveling salesman problem generally has a NP-hard complexity itself [85]. This implies that for each NP-hard LP that must be solved, a large sum of NP-hard sub-problems defining the MHCs will need to additionally be solved. The magnitude of this can be understood as follows: for every layout assessed during the LPs solution, *n* number of NP-hard flow distance sub-problems will need to be solved where *n* defines the number of unique process flow segments present. This, as one can imagine, presents an exponential increase in the solution difficulty.

It was also established prior, that to effectively account for evolving market conditions and business models, an evolving layout design would be required. This only further contributes to the difficulty of the problem as solving a DLP involves the simultaneous solution of multiple SLPs. This subsequently implies that for the complete assessment of a single layout design, as just described, it would now require $t \times n$ NP-hard flow distance sub-problems to be solved where *t* defines the number of periods implemented to adequately define the planning horizon for the evolving layout, which could theoretically be different for each of these periods. Furthermore, having to balance two competing cost functions and *t* times as many variables and constraints, could prove more difficult.

Lastly, it was recognized that to design a layout that would remain effective in the presence of uncertain conditions a robust stochastic approach would be necessary. Statistical model-based approaches to establishing a robust design have the advantage of more effectively capturing the range of uncertainties present as well as being more computationally manageable (remains a single NP-hard LP). The drawback to this approach is that the infusion of the uncertainty into the solution procedure results in a loss of problem behavior transparency. This is where scenario-based approaches are better suited. Their drawback however, is that by maintaining this behavior transparency, it requires a series of NP-hard LPs to be evaluated, one for each scenario in the scenario set defined a priori by the designer. In the context of the preceding discussion, this then implies that a series of these DLPs would need be solved, further compounding the difficulty and time intensive nature of the overall problem.

The majority of the time, a solution to the RLP has focused on the identification of the most robust static layout where the layout remains fixed throughout the planning horizon. From here forth this form of the problem will be referred to as the robust static layout problem or RSLP. On the other hand, few have solved the more generic robust dynamic layout problem (RDLP). As has been observed by the preceding discussion, this is due to how overwhelmingly difficult such a characterization of the problem is. Although more difficult to handle, this problem is the most general form of the problem and as such provides a designer with the greatest degree of flexibility and understanding of how to design the layout such that it will remain effective in the presence of evolving and uncertain conditions. As such, the following summarizing assertions can be made:

> **Assertion 11:** To effectively design a layout subject to evolving and uncertain market conditions and business models, the problem must be structured as a robust dynamic layout problem (RDLP).

> **Assertion 12:** To handle the difficult and time intensive nature of the RDLP, identifying an efficient solution method is imperative.

To gain improved insight on how the layout problem can been solved, the section that follows explores the literature in an attempt to identify viable methods for effectively solving the prior established form of the problem. The LP has been densely studied and to provide a complete survey of the literature would be an insurmountable task, but more importantly an unconstructive one as it pertains to understanding the problem of this dissertation. As such, the literature presented is not an exhaustive survey. Rather, it is a thorough review of the literature most relevant to the problem and for which solution novelty or exemplary results were achieved. This section attempts to answer the final sub-question of **Research Question 1**, on how the layout problem of this dissertation can be solved effectively.

2.6 Solving the Layout Problem

Over the last few decades, researchers have solved the LP in a number of different ways. These range from exact to meta-heuristic and more recently hybrid methods. This diversity in how researchers have been solving the problem is directly related to the variety of ways the LP has been formulated and to the identification of more effective solution methods. Here effectiveness can be understood as being how robust (consistent) and efficient (quick) the method is at achieving the desired solution results.

To follow in this section, the solution approaches implemented in the literature are categorized into four core groups: exact methods, heuristics, metaheuristics, and hybrid and evolutionary approaches. Each of these categories will be discussed briefly while a more comprehensive review of each of these solution approach categories applied in the literature can be found in Appendix A. To provide motivation in identifying a suitable solution approach to the layout problem being studied in this research, the association between solution method and problem formulation and its effectiveness will be observed. Although it has since been established that a DLP formulation will be required, the discussions will encapsulate both relevant static and dynamic formulations of the problem. Understanding SLP solution methods in addition to those applied to the DLP is important to the fundamental understanding of how the LP can be solved. Furthermore, the DLP is, for the most part, an extension of the SLP, hence it is not inconceivable that a method applied to the SLP could not also be extended to the DLP.

2.6.1 Exact Methods

Exact methods are solution algorithms that guarantee optimality. This is the major attraction of such methods of solution and the primary reason why they were so prevalent early in the problem's history. Optimality assurance comes at a cost though. In order to ensure optimality, an exhaustive search of the design space is required. For problems of small size such a search can be managed by partial/selective enumeration algorithms. However, as the size of the problem increases, the computational time required to achieve optimality grows exponentially until eventually becoming unreasonable. As such, exact methods are useful approaches when optimality is of the upmost importance and the LP is relatively small in size. Both of these characteristics were of major relevance initially as researchers were handling smaller LPs and most concerned with solving said problems to optimality, as observed by Kulturel-Konak [100].

A few of the more prominent exact methods implemented in the literature to solve the LP include branch and bound (B&B), dynamic programming (DP), and direct methods, with the dynamic programming being the typical method implemented to solve the DLP. Each of these methods are discussed at length in Appendix A and relevant works highlighted for context.

2.6.1.1 Applicability of Exact Methods to the Problem

As is stressed in Appendix A while presenting the such exact methods of solution, the major draw of exact methods is their ability to guarantee optimality. Guaranteeing optimality requires an extensive search of the design space however. For larger sized problems, searching this space becomes intractable. The inability of exact methods to solve problems of large size is one of its major drawbacks. The literature has demonstrated, at best, the ability to solve MIP formulated DLPs of size twelve departments by three periods within a reasonable duration of time. This was achieved by Lacksonen (1994) where solutions were generated in just over 5.5 minutes [104].

In addition to this, exact methods such as B&B and DP are better suited to handling QAP formulations of the DLP that are purely integer-based. With the exception of researchers such as Lacksonen, solving the MIP formulation of the DLP using B&B or DP has rarely been entertained by researchers as a result of this less than ideal compatibility. Even Lacksonen's (1994) research implements an initial stage that first solves the QAP formulation of the problem to reduce the size of the MILP problem that then needs to be solved by B&B and linear programming [104].

This observation brings up yet another important point. To solve the MILP formulation of the LP (static or dynamic) a direct method, such as linear programming, is required to completely solve the problem. In other words, B&B or DP alone are not capable of solving such a problem that captures layout continuity by continuous departmental locations. By definition a continuous variable has an infinite number of possible values [124]. For this reason, the full enumeration of such a continuous layout would be impossible. This is where the implementation of direct methods, such as linear programming, become necessary to solve the problem effectively. The major limitation, that has yet to be explicitly stated, of using direct methods, is their requirement that the problem be completely linear or at a minimum continuous. In each of the research cited before that addressed the continuous layout representation, the MIP formulation was either formulated directly or linearized to be a MILP formulation of the LP before solution. This presents a major roadblock to potential application to the LP of this dissertation given the need to account for flow feasibility in the model. A flow path feasibility guaranteeing distance-based objective function is anticipated to not be continuous let alone linear.

The less than capable nature of exact methods in solving larger problems is itself not a deciding factor in the applicability of exact methods to solve the proposed LP. The inability of exact methods to effectively handle the MIP formulation of the problem and potentially non-continuous functions is, however. The following assertion is then made:

> **Assertion 13:** Exact methods of solution are not capable of solving the layout problem of interest in this dissertation.

Next, the role that heuristics have played in solving the LP is observed.

2.6.2 Heuristic Approaches

Before the advent and maturation of metaheuristic techniques, heuristics provided the best alternative to exact methods. Overcoming the limitations that accompany exact methods has been the principal purpose of implementing heuristics techniques to facilitate more effective solution of the LP. Of these, the ability to solve larger problems more effectively has by far been the most prominent motivator to their implementation. This was no better demonstrated than by Rosenblatt, who himself implemented heuristics in his original QAP/S formulation of the DLP to enable larger problems to be solved by DP [146]. A complete review of heuristic techniques in the literature is presented in Appendix A, Section A.2.2.

Although heuristics enabled larger LPs to be solved in a reasonable duration of time early on, room for improvement remained. The problem-dependency that accompanies these heuristic approaches greatly limits their applicability to other variants of the problem. Furthermore, their often greedy and hyper focused tendencies often inhibit them from identifying the global optimum of the space effectively. The advent and maturation of metaheuristics to solve COPs, including that of the LP, provided researchers with a viable alternative to heuristics. As such, metaheuristic techniques become the focus of discussions going forward.

2.6.3 Metaheuristic Approaches

The more problem-independent and less greedy solution tendencies of metaheuristic approaches enable them to avoid the limitations stated before regarding the implementation of heuristics alone to solve the LP. Several metaheuristics techniques have been proposed in the literature to solve the LP. These include simulated annealing algorithms as well as intelligent approaches. Intelligent approaches consist of evolutionary algorithms such as genetic algorithms and other approaches like that of particle swarm optimization, ant colony optimization, tabu search, and colonal selection algorithms. Of the metaheuristics implemented in the literature, simulated annealing and genetic algorithms are the two most prominent methods of solution to the LP. As such, both algorithms will be observed here and further discussed thoroughly in Appendix A. In addition to these two metaheuristics approaches to the solution of the problem, hybrid approaches implementing one, both, or neither of these two metaheuristics have recently become the choice of researchers to solve the problem. Therefore, hybrid as well as other intelligent approaches will also be discussed.

2.6.4 Simulated Annealing

Simulated annealing has often been used by researchers to solve the LP. Simulated annealing (SA) is a probabilistic method, based on outcomes of statistical mechanics, for discovering a function's approximate global optimum, where said function may too contain local optima [29,59]. Simulated annealing was first introduced by Kirkpatrick et. al. (1983) and independently by Cerny (1985) to emulate the process of gradually cooling a physical system to its minimum energy state [94,44]. A comprehensive review of the literature pertaining to the application of simulated annealing to the layout problem is presented in Appendix A. The section to follow provides a summary and analysis of this review, particularly in regard to the potential viability of applying SA to the LP of interest in this research.

2.6.4.1 Applicability of SA Approaches to the Problem

Given the survey of the literature conducted on SA and presented in Appendix A, several observations and assertions can be stated. First and foremost, SA has proven consistently that it is an effective method of solution for the QAP formulated LP. Its predominate application to the QAP formulation of the LP is a testament to this. Coupling this with SA being an effective local search mechanism [137], the following assertion can be stated:

> **Assertion 14:** Simulated annealing should be implemented to facilitate the effective solution of a QAP formulated LP.

Furthermore, Tang's research demonstrated SAs capability of handling the QAP/U-SP form of the static problem. Furthermore, Sahin et al. demonstrated its ability to solve the budget constrained DLP problem. Coupling the two of these together, the following statement becomes reasonable.

> **Conclusion**: Simulated annealing has the capability of effectively solving the QAP/U-SP formulated budget constrained DLP.

Provided the earlier assertion made in the modeling section that the sequence-pair was the most suitable of the QAP models to the problem at hand, the above assertion and conclusion acknowledge that a SA approach should be implemented if a QAP/U-SP model is used to characterize the layout at any point during the solution process.

The literature survey also enabled a preferred annealing schedule for the SA algorithm to be established. Chen and Chang's proposed Fast-SA schedule demonstrated on average a 12x speedup in finding configurations of similar quality to that of the

Classical and TimberWolf schedules [47]. Figure 7 provides a comparison between the three popular annealing schedules implemented in the literature as applied to a layout problem where the goal was to minimize dead space. Its superior performance and overall robustness, as demonstrated in Figure 7, enables the following assertion to be made:

> **Assertion 15:** The Fast-SA annealing schedule should be implemented as the preferred schedule for the SA algorithm.

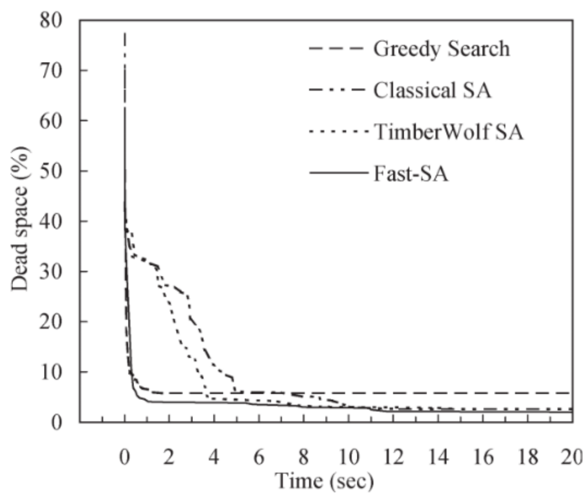


Figure 7 – Comparison of popular annealing schedules in solving a minimal dead space layout problem [47]

Furthermore, the literature provides insight into the appropriate heuristics required to perturb the configuration effectively. McKendall et al.'s perturbation scheme incorporating a look-ahead/look-back strategy demonstrated improved performance over the standard method implemented by many researchers for the DLP. Although an attractive method, alteration to the heuristics would be required in order to both account for machine addition/removal scenarios that may occur from one period to the next and further to encapsulate a sequence-pair model structure. The latter can be achieved by adoption of Tang's swapping procedure after the period of swap is chosen whereas the former will require the development of new heuristics. Furthermore, the perturbation schemes implemented by researchers, including McKendall et al., fail to account for fixed-entities and their impact on configuration feasibility, with Tang's research being one of the few exceptions to this. Although fixed entities were accounted for by his formulation, the perturbation heuristics remained unaltered. To handle the presence of fixed entities and their inhibiting effect on sequence-pair feasibility, new perturbation heuristics will be required. The inhibiting effect that fixed entities have on sequence feasibility will be detailed later. This concludes the discussion on the applicability of SA to the problem of this dissertation.

2.6.5 Genetic Algorithm

The genetic algorithm is another metaheuristic technique often implemented in the literature to solve the LP. The genetic algorithm (GA) is an evolutionary algorithm, based on C. Darwin's 1859 theory of natural selection [52] and the role it plays in biological evolution, that is capable of solving complex and difficult to solve COPs [58]. Popularized in 1989 after D. E. Goldberg's publication of "Genetic Algorithms in Search, Optimization and Machining Learning" [80], the advancement and application of genetic algorithms to engineering COPs has grown exponentially [58]. A comprehensive review of the literature pertaining to the application of genetic algorithms to the layout problem is presented in Appendix A. The section to follow provides a summary and analysis of this review, particularly focusing on the potential viability and application of GA to the LP of interest in this research.

2.6.5.1 Applicability of GA Approaches to the Problem

As can be observed from the expansive survey of the literature conducted and presented in Appendix A, the GA has proven to be a suitable approach to solving the LP of various formulations. The GA's ability to effectively handle problems involving non-linearity, non-convexity, multiple objective functions, as well as side constraints makes it an ideal choice for solving the budget constrained DLP formulated as either a QAP or MIP. The GA is also highly parallelizable. This is a major advantage of it and one that could prove essential to handling a computationally burdensome problem such as the one of this dissertation.

The literature provides several viable reproduction options for effectively evolving the population of the GA. Although observed in Appendix A have their own advantages, some present more promising behavior and applicability to the problem being addressed in this dissertation. It is evident from the literature presented that proportionate roulette wheel selection is the preferred method of selection. The major attraction of this selection operator is it provides a fitness driven approach to selecting the parents. Researchers including Conway and Venkataramanan, Ulutas and Islier, Baykasoglu et al., and even that of Mazinani et al. while solving the MILP DLP, implemented this as their selection operator. From this, it can be concluded that a roulette wheel selection operator should be implemented to facilitate the selection of parents for reproduction by the GA when solving either a QAP or MIP formulated DLP. As for the replacement operator, the approach employed by Balakrishnan et al. (2003), which ensures population uniqueness when selecting the next generation of individuals, is an entertaining option for future implementation, should a GA be used to solve the problem.

Also universal to either the solution of a QAP or MIP formulation of the problem by GA, the inclusion of a jumping gene operator (JGO) to further facilitate improved evolution of the population should strongly be considered. As observed by Tang et al. (2008) and Tang et al. (2011), the introduction of the JGO to solve the DLP improved the likelihood of achieving better convergence and diversity of the population and furthermore helps to avoid premature convergences [157,158]. These improved solution properties are a direct byproduct of the JGO's ability to better explore and exploit the search space as a result of it horizontally transmitting genes. Crossover and mutation variation operators are only capable of introducing vertical transmissions of genes from generation to generation. Therefore, regardless if a QAP or MIP formulation of the DLP is being solved, if a GA is implemented to facilitate solution, a JGO should be included as a supplementary reproduction operation in addition to crossover and mutation operations. Ripon et al.'s application of this operator to the QAP/S problem could relatively easily be altered to handle the position-pair ordered structure of the QAP/U-SP problem. On the other hand, application to the MIP DLP would require a more complex adaption of the operator. Novel procedures for performing the cut and paste and copy and paste operations on a vector string of real numbers would need to be developed. The benefits that the JGO can provide make this development a worthwhile investment though.

As for crossover and mutation operators, these are more specific to the formulation of the problem. While handling a QAP/U-SP formulation of the problem, Liu and Meller's approach to reproduction is ideally suited, unlike some of the other works reviewed, as it was specifically applied to such a position-pair structured problem formulation. Adoption of their uniform crossover and mutation technique would first require extension to the DLP. Extension of the uniform crossover operator takes little effort and would in many ways just be a fusion of Liu and Meller's crossover for the SLP and Mazinani et al.'s for the DLP. As has been done by most researchers while handling the DLP, appending a step for randomly selecting a period to perform mutation upon can enable Liu and Meller's mutation operator to be extended to the DLP. Another viable alternative to this method of mutation is Pourvaziri and Naderi's mutation operator that randomly swaps two period layouts of the offspring. Either of these options are suitable to the encoding representation of an individual characterized by a QAP/U-SP model. On the MIP side of the problem, Mazinani et al.'s use of a continuous uniform crossover operator and three independent mutation operators is an intriguing option. With their approach proving quite effective in solving both the QAP and MIP, implementing their approach in the solution of the LP being addressed in this dissertation should be strongly considered.

A summary of the preceding observations regarding reproduction strategies for solving the DLP with GA enable the following assertions to be made should a GA be chosen as the primary solution method to the DLP.

Assertion 16: Should a GA be employed, the following reproduction strategies should be implemented to effectively evolve the population and therefore solve a:

<u>QAP formulated DLP</u>

1. Ulutas and Islier's roulette wheel selection method

2. Liu and Meller's modified uniform crossover and mutation operators *adapted* to the DLP structure

3. Alternatively, Pourvaziri and Naderi's mutation operator

MIP formulated DLP

1. Mazinani et al.'s roulette wheel selection, continuous uniform operator, and tri-mutation operator approach

Either a QAP or MIP formulated DLP

1. Balakrishnan et al.'s (2003) uniqueness guaranteeing replacement scheme

2. Ripon et al.'s JGOs *adapted* to a QAP/U-SP or MIP representation of the layout respectively

In addition to potentially viable reproduction strategies, another notable observation from the literature on GAs is Liu and Meller's use of a multi-modal (QAP/U-SP and MIP) solution procedure for solving the MIP formulated SLP. Although their approach proved to be an effective method in reducing the solution space for the GA to search by leveraging the fundamental properties of SP, it also implemented linear programming to facilitate solution of the MIP for each QAP/SP solution evaluated by the GA. The ability to leverage linear programming to solve the MILP portion of the problem was this approach's major advantage. As observed at the beginning of this section on solving the layout, it was noted that direct methods such as linear programming were not a viable option for the problem at hand due to the non-linearity of the problem formulation required to adequately characterize a layout. In other words, the major advantage of the aforementioned procedure no longer pertains to such a problem as the linear programming would need to be exchanged for another heuristic solution technique. Again, although an attractive approach, said method is not viable for application to the problem formulation required in this dissertation.

2.6.6 Hybrid and Intelligent Approaches

In addition to SA and GA approaches there are hybrid approaches, which implement a combination of metaheuristic techniques or the combination of a metaheuristic technique and another solution method such as dynamic programming to solve COPs such as the LP. Every solution method has its own areas of expertise where they provide superior performance compared to others. An example of this was noted before in establishing the effectiveness of SA as a local search mechanism. Compared to GA, SA demonstrates superior local search performance. As such, one can postulate combining the two in order to leverage the superior performance of SA at searching the space locally while allowing GA to more effectively search the space globally. The result of this fusion is often a superior performing solution method with more robust characteristics as each solution method infused can be leveraged for what they are best at.

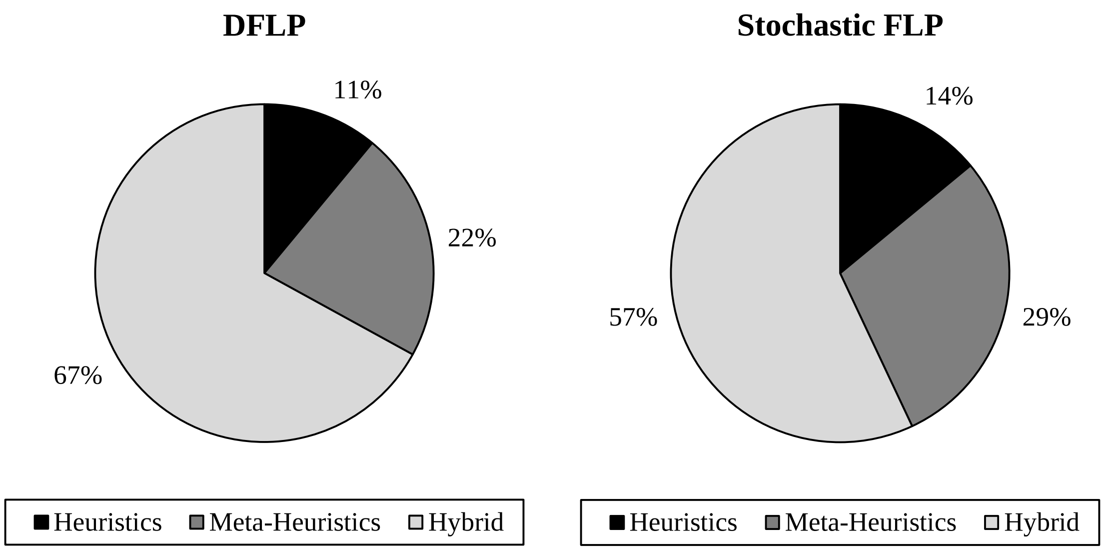


Figure 8 – Recent dominance of hybrid and meta-heuristics approaches to the solution of the problem [100]

The recognition that such an approach can provide substantial advantages, such as superior performance (solution results and times) and overall robustness, has in recent years led the large majority of researchers to employ hybrid approaches to solve the LP subject to evolution and uncertainty (DLP and RLP). Kulturel-Konak, as demonstrated in Figure 8, observed this paradigm shift in solving the DLP and RLP in his extensive review of the topic [100]. In addition to hybrid approaches, researchers recently have also entertained other intelligent approaches to solve the DLP. Provided in Appendix A is a review, brief in comparison to Konak or Moslemipour et al. [127,100] more expansive review, of some of the more notable applications of hybrid and intelligent approaches to the layout problem, specifically the DLP.

2.6.6.1 Applicability of Hybrid and Intelligent Approaches to the Problem

An assessment of the literature on hybrid and intelligent approaches revealed an assortment of viable solution methods for solving the LP. Of these though, one outshined. Although applied to just the QAP/S formulation of the DLP, Pourvaziri and Naderi's introduction of a hybrid multi-population GA with SA enhancement to the solution of the problem demonstrated highly attractive performance characteristics that should be translatable to other formulations of the problem. The robustness and ability of their approach to provide effective solution to the DLP within a reasonable duration of time is promising. Having also used GA to provide this solution performance, selection of GA to solve the problem of this dissertation becomes further justifiable.

The combination of evolving multiple populations concurrently and infusing SA are the main drivers of this superior performance. Both of these can be easily adaptable to other formulations of the problem solved by GA. To populate the initial sub-populations of Pourvaziri and Naderi's approach to the MIP formulation of the DLP, solution first to the QAP/U-SP formulation, which is far less computationally burdensome, could be performed. Adoption of Liu and Meller's GA approach extended to the DLP and simplified to just the QAP/U-SP portion of the problem (i.e. removing the internal MIP formulation solution procedure by linear programming), can facilitate not only the solution of said problem, but further enable the best region and non-promising region sub-populations to be populated. Furthermore, this GA could be hybridized by implementing SA, more specifically FSA, just as Pourvaziri and Naderi did for their QAP/S problem.

It has also since been established that should a GA be used to solve the MIP formulation of the DLP, Mazinani et al.'s GA approach should be implemented.

Mazinani et al.’s approach could presumably be enhanced further by altering the GA structure of their algorithm to encapsulate the multi-population procedure proposed by Pourvaziri and Naderi. As noted before, the sub-populations could then be populated with the results of the aforementioned solution to the QAP/U-SP formulation of the problem. Furthermore, hybridizing the GA by infusing SA to further enhance the best solution for each generational loop, as done by Pourvaziri and Naderi, could provide further improvement to the performance of the solution method.

As observed before, Liu and Meller’s approach coupled with Yang et al.’s approach to solving the MIP formulations of the problem using bi-model approaches of sorts demonstrates the potential for employing such a hybrid model approach to solve the MIP problem of this dissertation. Bi-model here is representative of either the simultaneous solution of both QAP and MIP models fused together and as done by Liu and Meller, or the sequential solution of the two to solve the problem in its original form. The latter of these two definitions relates to the discussion above that considers solving first the QAP/U-SP problem in order to then populate the sub-populations of the GA solving the MIP problem.

> **Assertion 17:** A bi-model hybrid approach, where a QAP/U-SP model is solved to some extent initially to then sufficiently populate the multi-populations, as defined by Pourvaziri and Naderi, of the Mazinani et al.’s GA approach to the MIP formulated DLP, enhanced to encompass this multi-population structure and inclusion of SA, should be implemented to solve the MIP formulated DLP most effectively.

2.7 Summary of the Literature

The preceding survey of the literature on modeling approaches, the addressment of evolving and uncertain conditions, measures of layout effectiveness, and solution methods for solving the problem have led to a series of observations and assertions that can be graphically observed in Figure 9. The assertions made prior collectively establish the overarching characterization of the problem that ensures a realistically viable layout that would remain effective in the presence of evolving and uncertain conditions could be designed. Furthermore, conclusions on potentially viable approaches to the solution of this well-established problem are made. The chapter that follows acknowledges the assertions and conclusions established in this chapter to formulate the methodology that will be required to effectively design a realistically viable layout subject to evolving and uncertain conditions. Furthermore, notable assertions, conclusions, and questions posed during this survey of the literature will be once again revisited later when recapitulation of the problem is provided before presenting the results of this work.

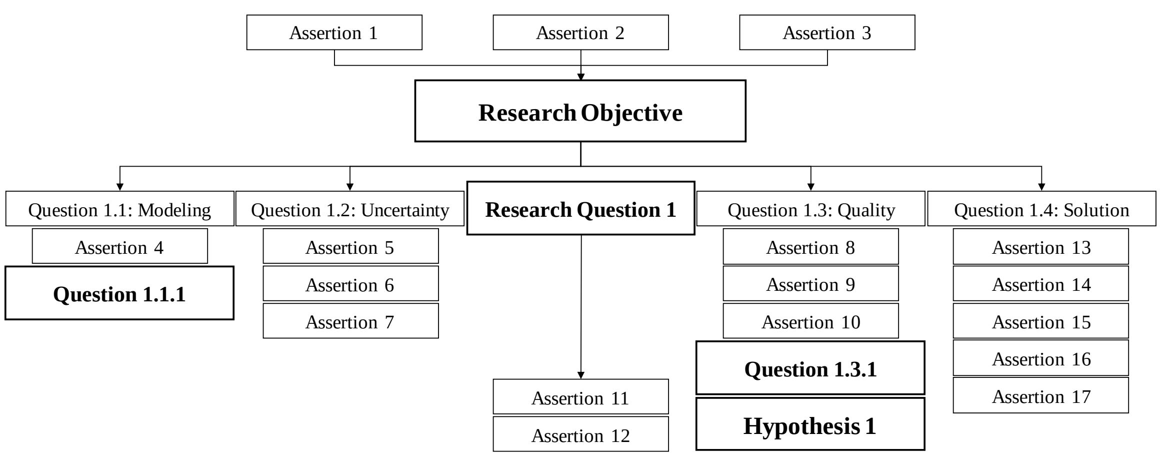


Figure 9 – Hierarchal representation of the problem

CHAPTER 3

METHODOLOGY FORMULATION

In the preceding chapter, a thorough understanding of the problem was gained and several observations, conclusions, and assertions were established as the literature on the layout problem was surveyed. These acknowledgements collectively motivate the need to solve such a robust dynamic layout problem and further address key gaps within the literature. Closure of these gaps is required for an environment subject to evolving and uncertain conditions to be designed effectively. Furthermore, these observations, conclusions, and assertions acknowledge potential formulations that adequately characterize such evolving detailed layout designs as well as viable strategies for effectively solving such an arduous problem formulation.

The proposed methodology of this dissertation provides closure to the identified gaps and leverages these potential formulations and strategies to effectively solve the problem of designing layouts subject to uncertain and evolving conditions. The proposed methodology encompasses three steps. The first step involves initializing the problem(s) to be solved, the second, then solving these problem(s), and finally third, visualizing the layout design(s) and leveraging the data generated by the implemented performance model to then make more informed and collaborative design decisions.

3.1 Overarching Methodology Framework

The overarching framework of the proposed methodology, titled **L**ayout Des**I**gn **V**isualization **E**nvironment (LIVE), is composed of three steps as mentioned before. This framework and the three steps can be observed in Figure 10, which provides a pictorial representation of the framework. The first of the steps encompasses initializing the problem(s) to be solved. In this step the overarching problem is decomposed such that a series of scenarios for which layout designs are to then be established for are defined. Each of these *N* scenarios is composed of a unique combination of market and business model conditions as prescribed by the designer.

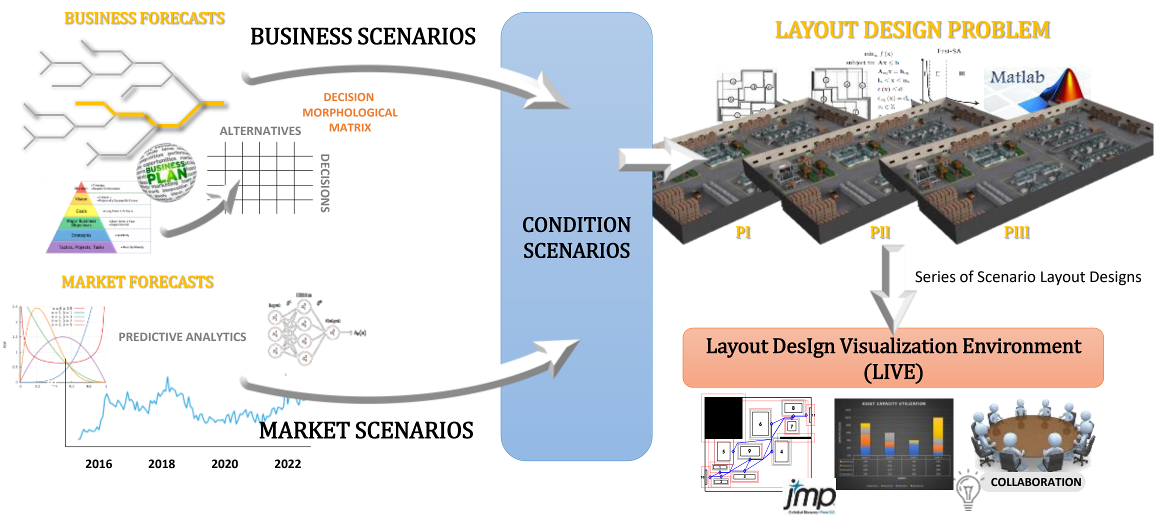


Figure 10 – Overarching framework of the LIVE methodology

In the second step, each of these *N* scenario layout design problems established by the designer are solved, producing then *N* layout designs for consideration. Taking the scenario conditions as inputs; costs, demands, rearrangement plans, etc. are defined and each (locally robust) dynamic layout problem is solved to identify the best or most locally robust layout design under the provided scenario conditions. As each of these scenarios is solved, the layout design space is then populated, and the performance results recorded.

With the layout design space completely populated and all the scenario layout problems solved, the final step of the proposed methodology is to then visualize these designs and analyze the performance results. It is proposed that an analytical cost model, which infuses operation management concepts, be implemented to provide improved insight into the performance of the layout design and the system it is a part of, where the system is defined as the layout design plus the conditions it is subject to. It is believed that the access to the additional data this model provides will enable designers to make more informed and strategic business decisions regarding the design of the system and the layout design itself. It is intended for the results to eventually be exhibited in a parametric environment. Though the development of this GUI environment is not within the scope of this dissertation, it is intended for it to be a future extension of the work. This environment would enable the designer to more easily observe the behavior of the layout design in the presence of such market and business model conditions and furthermore make more informed and collaborative decision on the final design of the layout and the system as a whole. The environment could further enable a rapid assessment of the design space, allowing designers to adjust market conditions and alter business model decisions to see the effect it has on the layout design from both a physical and quantitative business perspective (costs, production revenues, profits, utilizations, etc.). Now that the proposed methodology's framework has been presented, each of the three steps are outlined further.

3.2 Step 1: Problem Initialization

The first step of the proposed LIVE methodology is to initialize the individual scenarios layout problems. First though, the overarching problem or study to be considered must be decomposed in order to establish the combinations of market and business model conditions that define each of the scenarios.

3.2.1 Decomposing the Problem

As was observed before in the background section of this dissertation, poor problem transparency is one of the major shortcomings most existing methods that account for evolving and uncertain conditions suffer from. Most existing methods provide the designer with a single robust design that without clarity accounts for the relevant fluctuations and uncertainties of the problem. Scenario based methods for establishing robustness have the greatest potential for providing the required insight to enable more informed decisions to be made. With such approaches decomposing the uncertainty into a set of scenarios, the sensitivity associated with various market conditions can directly be observed by the designer. Unfortunately, too often these scenarios are collectively addressed within the solution procedure. As a result, a singular layout design remains the outcome without any understanding of how the conditions affect the design. As such, instead of directly incorporating fluctuations and uncertainties in the market conditions within the solution algorithm, as has often been done before, it is proposed that these be handled external to the solution procedure to an extent. The outcome of this approach then yields solutions to a series of condition scenarios, which can then be inspected manually by a designer in a layout design exploration environment. Such an environment enables a more thorough understanding of a design's behavior to the various condition forecasts.

Although this approach provides improved transparency, none of these designs themselves are technically robust, as their solution would be attained without consideration of the alternative scenarios it may encounter. Thus, it is proposed that a localized robustness approach employing Norman and Smith's method be implemented to provide robustness about the provided condition scenario. This local robustness would be established internal to the solution procedure and would therefore yield a design that is locally robust about the provided scenario's projected condition forecasts. This local robustness concept is illustrated in Figure 11, below. In these graphs, the dotted lines are representative of the projected condition forecasts for each of the M conditions composing each of the N independent scenarios being evaluated in the study. The solid lines on the other hand characterize the upper and lower bounds of uncertainty about these projected forecasts. This range of uncertainty, varying with time, is, as mentioned, to be captured by the implementation of Norman and Smith's distribution-based robustness method.

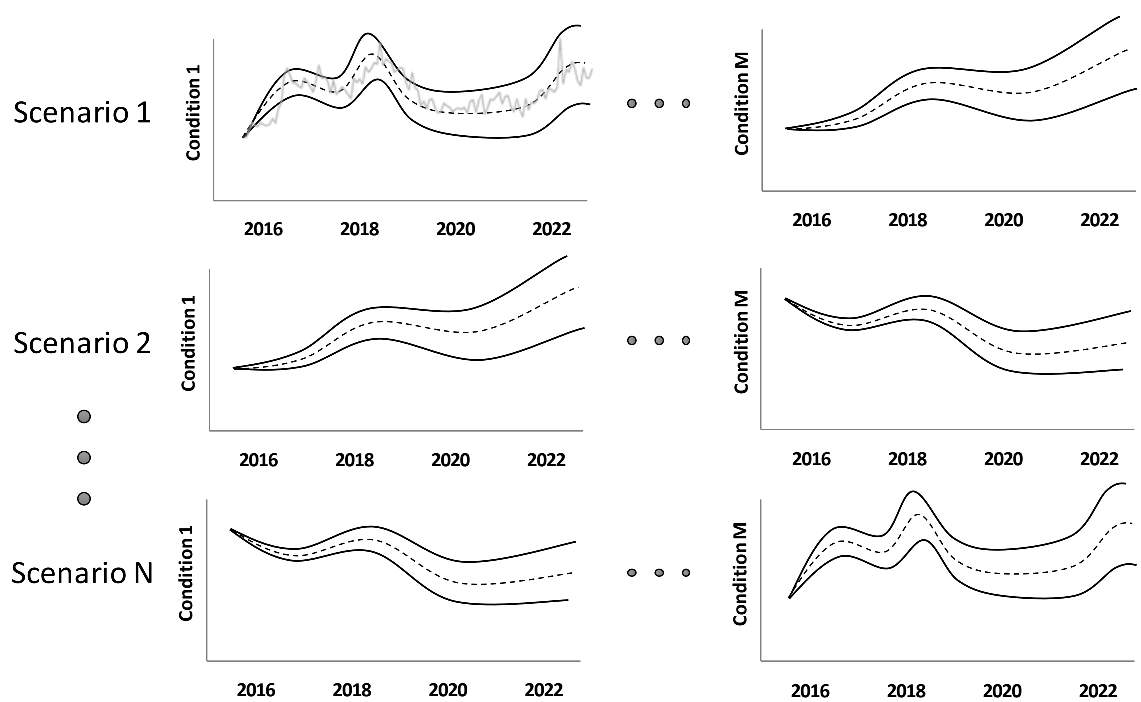


Figure 11 – Local robustness concept visualized

As noted before, the condition evolutions and uncertainties are to be handled externally *to an extent*. The general uncertainty of the projected condition forecasts are captured in the N independent scenarios, each of which generates a layout design for manual consideration in the design exploration environment. The evolution of each of the conditions is captured in the projected forecasts being time dependent, meaning the projected value can change over the layout design's planning horizon. This is demonstrated in Figure 11, where the nominal dotted line changes over the span of the planning horizon (2022 in this notional example). In other words, this is the portion of uncertainty that is handled externally, that is, external to the solution procedure. On the other hand, the uncertainty that accompanies these prescribed nominal projections is captured by the local robustness method, which is implemented internal to the solution procedure. In other words, each of the N scenario layout designs then becomes locally robust to fluctuations in the conditions about the nominal projected forecasts. The uncertainty in the expected value of each of the conditions is then captured in the definition of the scenarios whereas the uncertainty bounds about these expected values is captured by the local robustness method and thus within the solution procedure.

The benefit to this strategy is that it enables situations where the evolution of the market condition's nominal values is more likely known, but the fluctuations about these values less so. This situation is also depicted in Figure 11, where in the upper left graph the actual forecast of the condition is graphed as the gray stochastic behaving curve, which fluctuates about the nominal projected forecast, yet remains, for the most part, bounded by the uncertainty bounds prescribed by the local robustness method. Naturally these bounds will grow the further out into the future the layout is designed for and the conditions projected. This growing uncertainty is captured by the local robustness method, which defines the distributions of uncertainty about the expected or nominal condition value as a function of time. The further into the future the projection is, the more spread the distribution has and vice versa. The major advantage of this localized robustness approach is that it provides the designer with the ability to more thoroughly understand how the locally robust layout design changes as the market conditions nominally evolve over time. The sensitivity of the layout design to uncertainties in the nominal condition forecasts are now directly observable through this approach, whereas before, they were not.

Such an approach also has the added benefit of reducing the RDLP to just that of a DLP, making the problem more computationally tractable to solve. This would be the case when the local robustness strategy is not activated by the designer when studying the problem. Instead the designer may choose to completely enumerate the potential evolutions of the market conditions within the scenarios he/she defines to completely capture these uncertainties. Posterior, one may then apply Braglia et al.'s TPC function to each of the scenario layouts to establish the most robust design. Similarly, the designer may visually inspect the layout design space to identify regions in which the design does not vary significantly in order to establish robustness.

It was also acknowledged in **Assertion 6**, that existing methods of defining the planning period structure, which encourages the evolution of the environment, are limiting and often result in structured uniform planning period horizons that are poorly representative of real-life schedules. Although the aforementioned external scenario-based approach has considered just the evolution of the market conditions so far, these scenarios can too encapsulate the evolution and uncertainty associated with the business model. This provides designers with the ability to answer a vast array of design questions pertaining to the business model and operations that otherwise cannot be answered with traditional methods. For example, designers can define these business scenarios strategically to answer the following more prominent design questions:

- Which asset between a series of options would provide the most benefit to the company in the long term? Furthermore, when during the planning horizon should this integration occur?

- What is the best evolution schedule for the environment to maintain the best layout effectiveness and/or robustness?

- Is it wise to perform layout rearrangement at the onset of a period or sometime else when production demands may not be as high and a more favorable budget available?

The ability to answer questions such as these can greatly benefit a company and contribute largely to its ability to effectively design its environment with long term viability in mind. Answering questions such as (1-3) can also enable a company to maximize their potential for success in the market by making more informed and strategic business decisions. In addition to these questions, many others regarding the business model and its impact on the design of the environment can be answered. Furthermore, many useful questions with regards to the market conditions can too be answered as a result of this approach. This approach also has the added benefit of emulating Yang and Peters approach to reducing the dimensionality of the DLP that needs to be solved while avoiding the loss of transparency that accompanies their approach. For example, if the scenario only calls for one rearrangement to occur over a five-year planning horizon, then only a DLP of two periods needs to be solved, which is far easier achieved compared to a five-period problem (i.e. a period for each year of the five-year horizon).

The flexibility and insight into the problem that such an approach to capturing the evolution and uncertainty associated with the business model and market conditions has, is a major benefit. It enables designers to make more informed decisions regarding the design of environments subject to unpredictable and evolving conditions, which is the core goal of this dissertation's overarching research objective. With this decomposition of the problem established, a couple overarching assumptions regarding how these scenarios are to be defined is presented.

3.2.2 Accounting for Evolution and Uncertainty in the Conditions

As described before, evolution and uncertainty in the macro market (nominal projections) and business model conditions are to be captured by the scenarios defined a priori by the user in the first step of the methodology. In this research, two major assumptions are made. They are as follows:

1) A designer/business would be capable of populating a business model decision morphological matrix of the following:

2) Asset integrations

3) Product addition/subtractions

4) Layout rearrangement schedules

5) Product production rates

where each of the first three would have a value and time of change associated with it. Each of these may have more than one value and more than one possible time of change. The product production rates would be defined across the horizon with a range of discrete nominal values and variances. The latter property is to be leveraged by the proposed robustness method to account for uncertainty about the nominal values prescribed.

6) Market forecasts are established and known by the designer (through predictive analytics, insight, etc.) for the following:

7) Rearrangement costs

8) Product market values

9) Manufacturing costs

where each are defined across the horizon and have a range of discrete nominal values.

In other words, a series of market and business model condition scenarios would be readily available to the designer to populate a design of experiments or a matrix of condition scenarios to then study.

3.2.2.1 Notional Example of a Business Model Decision Morphological Matrix

Before continuing to discuss the second step of the framework, that is how to effectively solve the scenario layout problems that are defined in this first step, clarity on the concept of establishing a business model decision morphological matrix and how it relates to the definition of the restructuring schedule is to be provided. The idea of a business model decision morphological matrix mirrors the concept of a decision tree. A decision tree, in operations, is a hierarchal representation of decisions and their associated consequence(s) [144]. This representation resembles that of a tree in nature. At each intersection of branches a decision is made and depending on this decision a different branch path is followed. This branch path is nothing more than the resulting consequence of the decision. As more decisions are encountered, the tree continues to expand outward. This outward expansion from the origin decision (first made in the sequence) continues until reaching an outcome (leaf in the tree). This analogy is important as it depicts the various outcome scenarios and how they come to occur. Considering the possibility that decisions may overlap with one another, each of the unique scenarios can be understood to be a unique path from the root decision node (root of the tree) to that of an outer leaf node. As alluded to before, this hierarchal representation can be leveraged to identify beneficial strategies for success and further establish all possible decision paths.

In the context of this problem, the decisions of interest (branch intersections) are those relating to asset integrations (when and which), layout restructures (when and how many), and alterations to the business's production mix (additions, subtractions, and redistributions of resources). Each of these decisions then has a series of potential responses or in morphological matrix terminology, a series of possible alternatives. To best establish a thorough understanding of the concept, a notional example is presented below in Table 4.

In this notional example, it can be observed that the designer established two growth stages of the business. This is indicated by the first two decisions, which address expanding the business's operational capabilities by investing in additional machinery. Under these two decisions several potential responses are acknowledged, including the choice to not invest in a 2nd machine all together. Accompanying these are the decisions as to when each shall be purchased and integrated into the environment. The designer in this situation opted to consider a few options around the one-year mark for the first machine, while for the second machine a broader range about the second-year mark was to be considered in the study. It should be acknowledged that in situations where a new machine is integrated into the environment, product mix declarations *must* also accompany these decisions and furthermore must correlate to the same point in time that the integration occurs. This is where the compatibility of decisions becomes important and where the orange filled cells becomes relevant in this discussion.

Table 4 – Notional example of a business model decision morphological matrix

Decision	Variations of Decision			
1st Machine to Purchase	Machine A	Machine B	Machine C	
2nd Machine to Purchase	Machine A	Machine B	Machine C	No Purchase
Integration Date of 1st Machine	Month 9	Month 12	Month 15	
Integration Date of 2nd Machine	Month 18	Month 24	Month 30	
1st Layout Restructure	Month 12	Month 18		
2nd Layout Restructure	Month 24	Month 30		
3rd Layout Restructure	Month 36	Month 42	Month 48	
Product Mix @ Month 0	Option A	Option B	Option C	
Product Mix @ Month 12*	Option A	Option B		

Table 4 (continued)

Product Mix @ Month 18	Option A	Option B	Option C	Option D
Product Mix @ Month 24	Option A	Option B		
Product Mix @ Month 30*	Option A	Option B	Option C	Option D
Product Mix @ Month 36	Option A	Option B	Option C	
Product Mix @ Month 48	Option A	Option B	Option C	
Product Mix @ Month 60	Option A	Option B		

* = product mix definitions required as a result of a chosen machine integration

The orange filled cells in Table 4 represent a unique series of decisions made (a path in the decision tree), or in the context of this problem, a distinct scenario that would become a part of the design of experiments or matrix of condition scenarios to be studied. It can be observed that the designer in this notional example chose to define a product mix on an annual basis (likely to align with annual business review cycles). Additionally, the designer opted to examine a restructuring schedule involving three preordained evolutions of the layout. These three restructures need not overlap with a machine integration decision, though they can. In fact, in this distinct scenario, this is the case as the second preordained restructure aligns with the integration date of the second machine. Further, anytime machine integration occurs it is inherently assumed that restructuring must too ensue or at a minimum be entertained. Thus, in this example scenario, the true restructuring schedule for the defined scenario resembles the timeline provided in Figure 12, which as depicted indicates four distinct evolutions of the layout design (each dot on the timeline). The fourth is a result of the first machine's integration not aligning with a preordained restructuring yet requiring a restructuring to accommodate the integration of the new machine. The designer should be aware of this logic when establishing the restructuring schedules for each scenario. As one can observe, there is much design freedom in this approach, enabling the designer to tailor the study to their specific needs, requirements, and level of granularity.

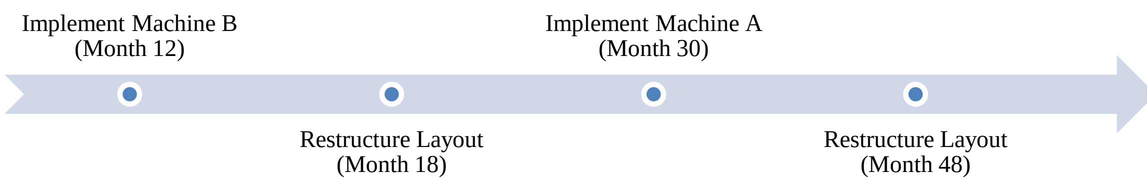


Figure 12 – True restructuring schedule for the notional example scenario

3.3 Step 2: Effectively Solving the Layout Problem

The second step of the methodology considers the actual solution of each of the scenarios, which in turn define a layout problem. As was well established in Section 2.5, solution to either a dynamic layout problem or a robust dynamic layout problem is a time intensive and difficult endeavor. **Assertion 12** acknowledged the importance of identifying an efficient solution method to such problems as a byproduct. Furthermore, the scenario approach defined before requires solution to a series of these problems and as such, the viability of this methodology becomes contingent upon being able to efficiently solve such problems. Fortunately, and as observed by Liu and Meller, layout planning is not a real-time decision process [115]. In light of this, solution quality should be the biggest concern so long as achieving said solution can be performed in a reasonable duration of time. Consequently, the following proposed solution method seeks to effectively solve the robust dynamic layout problem in a reasonable duration of time.

To achieve this, it is proposed that a bi-model multi-stage hybrid solution approach be implemented to solve these budget constrained (locally robust) dynamic layout problems. **Question 1.1.1** which questions whether a more simplified model in that of a QAP could be leveraged during the solution process, is entertained by implementing a bi-model approach (QAP/MINLP), where the more tractable QAP/U-SP model can be leveraged to refine the solution space and enhance the solution of the MINLP formulation of the problem, thereby reducing computational times and improving solution quality. Furthermore, the implementation of a hybrid optimization approach enables the superior niche capabilities of different solution techniques to be leveraged. As observed in Section 2.6.6, a byproduct of this is improved solution performance, which is highly favorable in this application.

The proposed bi-model multi-stage hybrid approach is to be partitioned into two distinct stages. Inspired by the work of Pourvaziri and Naderi and those who have attempted to solve similar MIP formulated LPs, solution to the MINLP formulated RDLP is assisted by the initial solution of a more tractable form of the problem. In this case, that would be a QAP/U-SP formulation of the problem. In the proposed approach, the first stage is then to provide solution to this formulation while in the second stage, the outcome of the first is leveraged to stimulate its solution to the overarching MINLP formulation of the problem.

3.3.1 Stage One: Solution to the QAP/U-SP RDLP

To populate the initial populations of Stage Two's GA, it is proposed that the QAP/U-SP formulation of the problem be solved to some extent, which fully acknowledges

Assertion 17. A by-product of this proposition is the second of the major research questions. It is as follows:

> **Research Question 2**: To what extent does the QAP/U-SP formulated problem of Stage One need to be solved to adequately populate the initial populations of Stage Two, such that the MINLP formulated problem can be solved most effectively?

It is anticipated that answering this question will require a balance between computational time and the quality of the designs provided to Stage Two to initialize the populations. It may not necessarily be useful to solve stage one to convergence given that the goal is to only populate the initial populations with good solutions, not necessarily converged solutions. With that said, such an approach enables the designer to neglect Stage Two all together in favor of a more rapidly attainable result procedure and a more "conceptual" understanding of the environment's design if he/she chooses. This can be achieved by solving just the QAP/U-SP formulation of the problem in stage one to convergence and neglecting the second stage completely. Although not an original intention of this work, this is a ancillary benefit of this approach.

3.3.1.1 Mathematical Representation of the Layout

The proposition that stage one solves the QAP/U-SP formulation of the problem was well established before by **Assertion 4**. As observed, it is the most capable of the discrete models at adequately characterizing the environment especially when Tang and Wong's version is adopted. Furthermore, such an approach has the major advantage, when coupled with the appropriate buffer added to each object, of providing solutions that are guaranteed to be feasible in the MINLP formulation of the problem. This implies that solutions are both feasible on the basis of all constraints (including overlap constraints which are typically troublesome to deal with stochastically for a continuous formulation) and path flow feasibility. In this case, the inability of the QAP formulation to characterize continuous layouts is an advantage. A further advantageous by-product of this outcome is that the advanced flow method could potentially be replaced with a rectilinear method without sacrificing too much accuracy, yet gaining substantial reductions in solution time. As was the case before, Tang and Wong's formulation will require alteration to encompass the (locally) robust dynamic nature of the problem and the objective function unique to the developed performance model of this dissertation.

3.3.1.2 Solution Approach

It is also proposed that this QAP/U-SP RDLP be solved using a hybridized GA with SA, specifically FSA implemented to enhance the solution procedure. McKendall et al.'s perturbation scheme incorporating a look-ahead/look-back strategy is to be implemented in the FSA algorithm, but with new heuristics to account for the unique nature of this dissertations layout problem formulation.

To provide effective evolution of the population, it is proposed that the strategies provided in Table 5 for a QAP problem formulation be implemented. Liu and Meller's unique application of GA to the QAP/U-SP data structure [115] makes their strategies ideally suited as they already account for the position-pair nature of the data structure. They will however require slight alteration to account for the DLP structure of the problem. This will involve the implementation of a preceding random selection of a period to perform the operations on. Ripon et al.'s jumping operations [143] will also require adaption despite their QAP application. The position-pair data structure of the SP representation differs from QAP/S data structure considered by Ripon et al., which was used to construct the original jumping gene operations.

An anticipated challenge of handling this problem formulation involves the effectiveness of these reproduction strategies and also the perturbation scheme of the FSA technique to generate feasible sequence-pairs. As has been observed before by the author, ensuring sequence-pair feasibility in the presence of constrained objects in the space is an arduous task that can greatly inhibit the effectiveness of the perturbation scheme and therefore the performance of the solution method. Liu and Meller also observed this difficulty of identifying feasible sequence-pairs when handling a highly constrained layout (i.e. high area utilization) with a GA. To combat this, they incorporated a 5% increase in the facility area along with a penalty function to account for the sequence-pairs with marginal violation of the true boundaries. This proved to greatly reduce solution times [115]. As such it is proposed that a novel method incorporating new heuristics be developed to handle this situation and better ensure sequence-pair feasibility during the perturbation and genetic reproduction processes.

Finally, after having performed some preliminary tests of the SP formulation subject to constrained objects in the space and additionally attained a more thorough understanding of the SP representation's fundamental principles, it was discovered that a strong correlation existed between the placement of the constrained object in the space and its placement in the sequences of the feasible sequence-pairs. This correlation was a by-product of the fundamental construct of the representation's sequence ordering relative to the corresponding diagonal line that bisects the space. It is therefore proposed that a method exploiting this behavior be implemented to promote the more efficient discovery of feasible sequence-pairs by these perturbation schemes and reproduction strategies. This method will be expanded on in the implementation chapter to follow.

Table 5 – Proposed genetic reproduction strategies

		Model	
		QAP/U-SP	MINLP
Selection Operators	Selection:	Liu and Meller's / Ulutas and Islier's roulette wheel selection [115,167]	
	Elitism:	Liu and Meller's *k* best with revised improvement heuristics [115]	Traditional *k* best
Variation Operators	Crossover:	Liu and Meller's modified uniform operator *adapted* to the DLP structure [115]	Mazinani et al.'s continuous uniform operator [118]
	Mutation:	Liu and Meller's mutation operator *adapted* to the DLP structure [115]	Mazinani et al.'s tri-mutation operator approach [118]
	Jumping Gene:	Ripon et al.'s cut and paste and copy paste operations *adapted* [143]	

3.3.2 Stage Two: Solution to the MINLP RDLP

3.3.2.1 Mathematical Representation of the Layout

To solve the MINLP formulated RDLP for each condition scenario, the MINLP formulations itself must first be established. As observed before in Chapter 1, the more accurate the model captures the real-life behavior and general characteristics of the environment, the more realistically viable the design will be in practice. This is in turn is imperative to effectively designing an environment such that the largest benefit from performing the layout design process can then be realized, which encompasses a major goal behind this dissertation. To achieve this goal and adequately model the environment it is proposed that Barbosa-Póvoa et al.'s (2001) non-linearized MIP formulation be adopted with some alteration (**Observation 2**) in Stage Two. These alterations will be outlined in the subsequent section when the proposed approach to evaluating the performance of each layout design is presented.

3.3.2.2 Solution Approach

To solve this MINLP formulated RDLP, it is proposed that a GA be adopted as the evolutionary algorithm due to its prior proven success in solving the general LP, its ability to effectively handle problems involving non-linearity, non-convexity, multiple objective functions, as well as side constraints (e.g. budget constraints), and its parallelizability. The latter of these will become essential to solving the problem in a reasonable duration of time.

To facilitate effective evolution of the population and thus solution, it is proposed that the reproduction strategies established in **Assertion 16**, for MIP formulated problems, be adopted. For completeness and clarity these strategies, acknowledged in **Assertion 16,** are provided in Table 5. As can be observed, for the most part Mazinani et al.'s GA reproduction approach is adopted. Except for Ripon et al.'s jumping gene operations, the other genetic operators are directly applicable given their original application to such a MIP formulated problem. The jumping gene operations however will require adaption to the MIP data structure as Ripon et al.'s original application was to that of a QAP/S data structure of the DLP.

It is further proposed that Pourvaziri and Naderi's tri-population GA structure be adopted. The observed robustness and performance improvements of its solution to a QAP/S formulated DLP are encouraging. The adoption of Pourvaziri and Naderi's tri population approach however, presents a major challenge of establishing an effective method of populating these three populations for a MIP formulation of the problem. This challenge is overcome however by the inclusion of the proposed Stage One solution to the simplified form of the problems. As observed, this stage solves, to an extent, a more fundamental version of the problem such that Stage Two's initial populations can then be effectively populated and moreover done so reasonably fast.

3.4 Step 3: Evaluating the Layout Design

To evaluate the performance and feasibility of the layout designs generated during the solution procedures outlined before, it is proposed that a cash-based performance model be developed to determine how well each layout design performs and that a constraint model, encompassing a wide range of constraint factors, be developed to determine its feasibility. In the end, it is these models that determine which design is considered to be the best solution to the provided problem(s) by the solution algorithms outlined before.

It is proposed that the performance model be comprehensive and yet not overly so such that defining the necessary input costs, revenues, etc. proves too difficult for a designer. It is also proposed that a cash-based objective function be leveraged to provide a performance metric that all stakeholders involved in the design decision-making process can easily comprehend. This model shall consider three distinct cost categories. They include indirect and direct costs of production as well as capital expenditures.

The indirect costs of production component can be decomposed further into RCs, and other indirect costs associated with production. Using Barbosa-Póvoa et al.'s (2001) formulation as a baseline, alterations to it are required to encapsulate the evolving nature of the layout and furthermore the budget constraints associated with this evolution. Barbosa-Póvoa et al.'s (2001) formulation is of a SLP; therefore, extension of it to the DLP by the inclusion of RCs in addition to the MHCs is required. These RCs are to be defined according to a distance-based variable horizon method along with a loss of production cost method as established by **Assertion 8**. In addition to these RCs, budget constraints are to be incorporated such that financial resource restrictions on layout evolution can be considered as well.

The RCs are only one portion of the indirect costs of production component. Other indirect production costs, which do not fall under these two cost umbrellas, need to be estimated as well. To provide estimates of the other indirect costs, a method that allocates indirect costs on a product-basis and according to a percentage (value based on the nature of the processes involved in manufacturing the products) of the product's direct costs of production is proposed. The direct cost of production must first be known for each of the products in order to establish these estimates. As such, how these direct costs are estimated in this research is addressed next.

In addition to the indirect costs of production component, there are also the direct costs of production that must be considered. Establishing estimations of the direct production costs is essential. Not only does it enable estimations of the other indirect costs as noted, but it also provides closure to the cash-based performance model's formulation which seeks to establish the performance of a layout design on a cost and profit margin-basis. Assessing a layout design on these two fronts is beneficial as these quantitative figures are often those managers base their decision from, but are also those that can be leveraged to design a layout from a mathematical programming perspective. As observed in Section 2.4.4**,** SEER-MFG provides the necessary direct production cost granularity needed to complement the indirect cost formulations (MHCs, RCs, and other) defined before. An intimate knowledge of each process involved in the manufacturing of the various products is required by this approach. A SEER-MFG model of the manufacturing environment has to be constructed and the process information noted prior used to define each of the individual process sub-models that compose it. Once the SEER-MFG model has been constructed, direct production cost estimations can then be generated, subsequently enabling the other indirect costs to also be defined as noted before. Though a promising approach, it is proposed that instead a generative-analytical model developed by the author be developed. This model shall encompass all the same costs that SEER provides, but do so with fewer inputs and more simplified models. Models in which would be validated on a per case-basis. Instead, linear cost models based on designer estimated cost forecasts will be leveraged to define the direct production costs. The motivation behind this approach, rather than leveraging SEER, is that it allows the LIVE tool to remain open-source and does not require access to the SEER software package. Furthermore, it reduces the upfront effort for the designer of having to learn SEER. It also reduces the effort by no longer requiring an elaborate SEER model to be built. It was for these reasons, that it is proposed that the analytical model be developed rather than going the SEER model direction. This is not to say that SEER could not be implemented in an extension of this dissertation.

It is also important to note that the SEER software package also does not characterize the MHCs to the level of detail that is required in this dissertation. Now although Barbosa-Póvoa et al.'s (2001) formulation, adopted as a mathematical programming foundation, includes a MHC method, it too requires not only modification to account for the dynamic nature of the problem, but also to account for flow path feasibility considerations. As established by **Assertions 9** and **10**, their rudimentary rectilinear method needs to be substituted for a method that defines the flow distances such that flow feasibility is guaranteed. As such, the novel flow distance method previously developed by the author and discussed briefly in Section 2.4.2.5 is to be leveraged. Furthermore, infusion of Norman and Smith's robustness parameters into the MHCs and also the RCs noted before is required to allow for the formulation to encapsulate the local robustness method proposed earlier. If the designer chooses not to incorporate such an approach, a switch to be implemented can effectively render these parameters inactive within the formulation.

Two of the three cost components have since been established leaving just the capital expenditures component undefined. This cost component is included to account for capital expenditures that occur during the planning horizon. For example, purchasing a new machine in year two would become a capital expenditure of that period in the planning horizon. This expenditure and that of the cost to rearrange will be compared to the budget of the firm to evaluate feasibility. To establish a value for the acquisition and installation cost of a machine, an estimation provided by the designer for this cost is required under this proposed formulation. A major advantage of considering the capital expenditure costs, in addition to the other cost, is that it enables designers to begin to evaluate potential investment opportunities for their firm with this methodology and performance model.

To provide closure to the performance model, estimations of the product market values will be required. Provided that the designer can supply estimates of these, the revenue for the system can then be identified. With the revenue known and costs also known, the performance of the layout design and the system as a whole can be evaluated with its profitability at the core of the cash-based performance model.

In addition to the performance model, several constraints are to be considered as part of the proposed constraint model. As mentioned before, budgetary constraints are to be considered, which will provide restrictions on the evolution of the layout design. In addition to these, boundary, access to the I/O points of the stations, and object overlap avoidance constraints are all to be considered to evaluate layout designs for feasibility. Many of these are encapsulated in Barbosa-Póvoa et al.'s (2001) formulation; however all will require alteration to encompass a, unique to this dissertation, multi-spacing interaction as well as the DLP structure. The multi-spacing interaction will attempt to inherently capture the appropriate degree of spacing as specified by safety guidelines for moving about the objects in the space as well as for required maintenance procedures that need be performed on each object without obstruction. The former will be instrumental in dictating the MHCs whereas the latter will be more relevant in determining how closely packed the objects can be without violation of the overlapping constraints. It is also proposed that a cash-based penalty function be implemented to account for any violations of the boundary constraints and budgetary constraints.

3.5 Concluding Remarks on the Proposed Methodology

The outlined approach to solving the individual layout problems defined by the scenario set, synthesizes the best performing strategies proposed in the literature coupled with new novel techniques for enhancing further the solution process and the formulation. It is this synthesis and further enhancement that makes the following hypothesis possible:

> **Hypothesis 2:** **If** the proposed bi-model multi-stage hybrid solution approach is implemented to solve the MIP formulated RDLP, **then** the problem will be solved most effectively, in terms of solution quality.

Additionally, the overarching LIVE methodology proposed, which encapsulates the aforementioned proposed approach to the formulation and solution of the layout problems, further improves upon the existing methods of capturing evolving and uncertain conditions. The proposed implementation of a local robustness method and an external handling of the scenarios describing the evolution of the market and business model conditions provide designers with the ability to answer many important questions regarding the design of an environment. The improved problem insight and transparency that this methodology as a whole can provide leads to the following overarching hypothesis of this research:

> **Overarching Hypothesis: <u>If</u>** the problem of designing an environment subject to evolving and uncertain market and business model conditions is solved with the proposed LIVE methodology, **<u>then</u>** designers will be capable of making more informed and collaborative decisions on its design.

This overarching hypothesis directly addresses the research objective of this dissertation. It acknowledges that a highly involved methodology is required to achieve an improved layout design process.

CHAPTER 4

METHODOLOGY IMPLEMENTATION

With the LIVE methodology's overarching formulation established in the preceding chapter, attention turns now toward the technical implementation of the methodology. The chapter is broken into four sections. Combining the middle two sections, the format mirrors the three steps to the methodology. In the first section of this chapter, the first step of initializing the problem(s) is discussed. Following this, the second section focuses on Stage One and how layout designs are first defined and subsequently altered in pursuit of alternative, potentially superior, designs. In other words, an overview of the Stage One solution procedure is discussed. The third section is much like that of the second, the difference being that now the focus is on Stage Two. The composition of the second and third sections encompasses step two in the methodology. The fourth and final section details the models developed to evaluate the performance and feasibility of the layout designs established as a by-product of the procedures outlined in sections two and three. It is these models that can then be leveraged to make more informed decisions on the design of the layout. This provides a brief overview of the proceeding sections of this chapter and with that a more in-depth outline of the first section is provided.

Before though, a few terms require definition. For future reference when referring to a *layout* it implies a singular arrangement of the objects in the space. A *layout design* on the other hand refers more broadly to the compilation of layouts that form a series of object arrangements. The term layout design can be synonymous to that of the term layout in the case where such a series consists of only a singular period and thus layout.

Now when referring to an *environment* this refers to a *layout* arrangement *and* the conditions it is subject to. These conditions include operational properties such as capacities, labor availability, production rates, processes, costs, etc. A *system* differs from an *environment* in that it is a compilation of the *layout design* as well as the operational properties across the entire planning horizon of the layout design. In other words, a system is a compilation of environments. Like before with the layout design, a system can be synonymous to that of an environment when the layout design consists of only a single arrangement of the objects. These subtle distinctions are important to understand before going forward.

4.1 Step 1: Problem Initialization

Initialization of the problem is not only the first step in the methodology, but it is also one of the most critical. In the LIVE methodology, the initialization of the problem constitutes more than just establishing the unique scenarios (i.e. market and business model conditions) to run as was outlined before in the preceding chapter. It also constitutes establishing the approaches/methods to deploy during the solution procedures and further the definition of the optimization parameters, which directly impact the solution performance. Moreover, it also establishes the physical properties of the layout such as the objects present (stations and regions) and outer mold line (OML) of the layout in each of the scenarios.

4.1.1 Defining the Structures of the Scenarios

Before any of these inputs (properties, parameters, or conditions) can be defined for a scenario, the structure of each of the scenarios must first be established. In this LIVE methodology, these inputs are all dependent on this structure. This structure is a composition of the planning horizon length, forecasting intervals to be leveraged, and the restructuring schedule to be examined in the scenario.

4.1.1.1 Establishing the Planning Horizon

First the planning horizon length and its segment composition are to be established by the designer. The segment composition is nothing more than the forecasting intervals. These intervals include both the frequency at which the designer desires to establish the forecasts of the market conditions, but also the timing of strategic business decisions relating to the production rates (i.e. when the production rates should change). This could be the result of a new machine and thus process-line or product added to the system or simply a desire to redistribute the production mix. Therefore, if a three-year horizon is to be analyzed in a yearly decomposition of the market condition forecasts and production rate decisions, then the horizon would be defined by a monthly scaled timeline resembling that provided below in Figure 13.

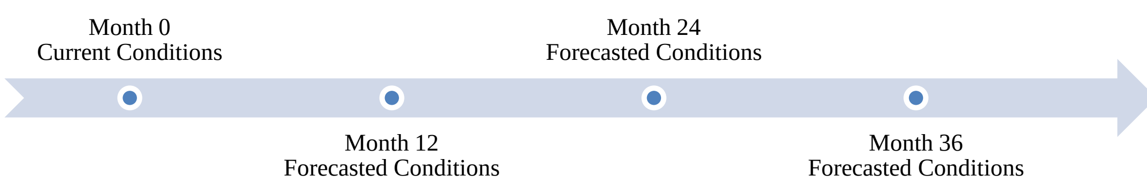


Figure 13 – Example of a planning horizon

4.1.1.2 Establishing the Restructuring Schedule

In parallel to the planning horizon's definition, the restructuring schedule to be examined in the scenario is also established. The restructuring schedule, as defined by the designer, constitutes the specific timings for when restructuring of the layout design is to be performed. In addition to these prescribed restructures, any time in which a new or existing asset is added or removed from the environment, a restructuring must at least be considered. In other words, a restructuring must accompany such an event. The concept of a business model decision morphological matrix, outlined before, can be leveraged by the designer to appropriately define the scenario's restructuring schedule given this logic. Like the planning horizon, the restructuring schedule is defined on a monthly scale. An example of a restructuring schedule whereby the designer requires for a restructure to occur in year two and that it is anticipated that a machine be purchased and added to the environment 6 months from now yields the restructuring schedule shown in Figure 14. Establishing the restructuring schedule also establishes the nature of the problem to then be solved. In this case, since two restructures are to occur, it establishes that a three period DLP is to then be solved in this scenario.

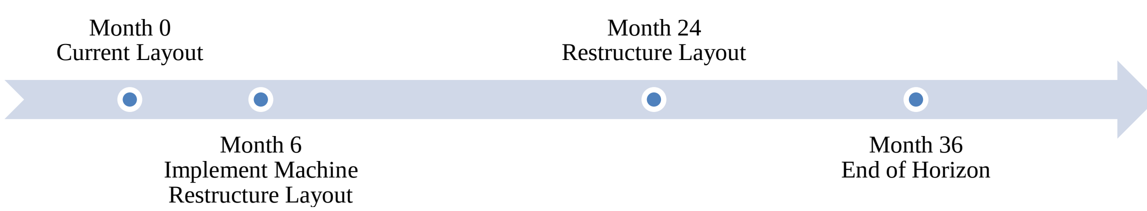


Figure 14 – Example of a restructuring schedule

4.1.1.3 Establishing the Scenario Structure

Once the planning horizon and restructuring schedule have been defined for a specific scenario, the two are then joined to establish the unique points in time correlating to a restructuring, a condition forecast, business decision, or all three. Joining the two together then yields the scenario structure. Continuing the examples from before, the planning horizon and restructuring schedule when joined resembles that demonstrated in Figure 15. Colors on the timeline have been added to distinguish the three distinct periods of the DLP. The first of these periods spans from the start to the 6^{th} month point where the machine is to be implemented thereby requiring a restructure to occur, the second from month 6 to month 24 where just a prescribed restructuring is then to proceed, and finally the third period running from this 24^{th} month point to the end of the horizon (i.e. 36 months out). With the inclusion of the 6^{th} month restructuring event, an additional forecast point for the conditions at this 6^{th} month point must be defined by the designer. The simplest way to define the conditions at this extra forecasted point would be to use linear interpolation. As will be highlighted later, this aligns with the assumed behavior of these conditions across the forecast segments.

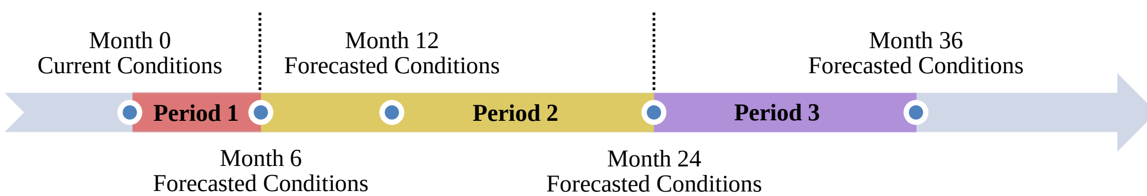


Figure 15 – Example of a scenario structure

Now with this scenario structure established the inputs are then able to be defined by the designer. The order in which these are then defined is not important in this implementation as they are independent of one another. The remainder of these scenario inputs will be presented in an order in which the author believes to be the most comprehensible and logical, however.

4.1.2 Defining the Physical Properties of the Layout

Defining the physical properties of the layout is the first action following the establishment of the scenario structure. These properties include the OML of the layout, the relevant spacing properties, and the stations and regions composing the objects to be placed in the environment each period along with their associated properties. With a rectangular OML assumed in this dissertation, the OML is defined by its lower-left corner (defaulted to be 0, 0) and its upper-right corner (xmax, ymax). In addition to these, the walking and maintenance spacing for the boundaries of the space are defined.

The next task is to establish the objects present in each of the periods (i.e. restructuring segments; three in the earlier example). All the region and station objects implemented across the entire scenario set are to be established by the designer and imported in the form of csv files (one for the regions and one for the stations). It is from these collections that the specific regions (if any) and stations to be included in the current scenario can be selected from. For each period of the scenario the relevant regions and stations are defined by indexing the appropriate one in the imported datasets. These datasets encompass a variety of object specific properties including, but not limited to, their dimensions, maintenance spacing, and installation/uninstallation times when applicable. One may refer to Appendix C for a comprehensive list of these properties for each of the datasets. Once the necessary regions and stations for each of the periods of the scenario have been indexed, all the data provided in the datasets is automatically populated into data structures which can be easily referenced later when evaluating the performance of the layout designs and determining their feasibilities with the performance and constraint models.

Now that the regions (if applicable) and stations have been indexed for each period, their placements are to be defined. The regions, by default are fixed in the space as they represent pillars, walls, or other inaccessible areas. As such, the designer must physically establish their positions in each period layout. Often, this will be as simple as defining the positions and orientation in one period and then copying them over to the other periods as these regions are likely to remain throughout the planning horizon. That is, unless it is expected that a structural pillar or wall will be removed, which as one can imagine, is unlikely. The stations on the other hand can have one of two distinctions. Stations can either be movable or fixed within the period layouts and they need not be one distinction throughout the horizon. For each period the indexed stations are labeled as one of these two distinctions. For those labeled as being fixed, they require their positions and orientations to be established. As for the movable objects, the designer can establish their initial position if desired, though it is not necessary. If evaluating an existing layout however, these initial placements are required in order to eventually evaluate the rearrangement costs in the first period as the layout would need to be rearranged from its existing state to the one of the first period. It is recommended that these initial placements be established with what is believed to be a good configuration since the solution algorithms implemented consider this initial provided one while searching for the best solution.

For each of the stations, additional data is also to be provided by the designer. Like before with the station and region datasets, another dataset is imported and provides a complete list of all personnel or potential personnel. The composition and format of this dataset is elaborated on in the Appendix C. Now for each of the stations, the relevant number of workers manning those stations are assigned from this personnel list to the stations. This assignment also brings along their wage rates, whereby if there is more than one worker their average rates can be automatically established. These labor rates for each of the stations and number of workers manning the stations are later leveraged when computing direct production costs in the performance model. Additionally, the number of dedicated material handlers of each period must be defined by the designer. This will later be leveraged by the performance model to establish the material handler utilizations in the system.

4.1.3 Defining the Market and Business Model Conditions

Defining the market and business model conditions is the next step in the process of initializing the scenario problems. These conditions encapsulate a variety of inputs that will be later leveraged by the performance model to compute the manufacturing costs, revenues, and utilization levels of the system. These inputs have a range of different formats. These formats will be noted as the inputs are presented.

4.1.3.1 Period-Based Conditions

The first of these inputs to be defined by the designer is the processes present in the system, which is analogous to saying the products to be produced in each period. These processes or products are defined on a period-basis to account for the common occurrence of a restructuring being triggered by the inclusion of a new, or exclusion of an old, station (i.e. a machine for example) in the system. With such an event, the processes present inherently change as any associated with this new or old station are then added or removed from the system respectively. First the designer is required to establish all potential processes present in each period of the scenario. This entails establishing for each process its process flow. In other words, the ordered sequence of station objects that each product visits as it passes through the environment going from that of a raw input to a finished good. Once this collection of processes is established, the designer can then index the relevant processes to include in each period of the scenario.

The other two period-based inputs to be defined are the capital expenditures and budget constraints. The capital expenditures are defaulted to zero unless an event such as a new station is purchased in a specified period. In the case where a new station is purchased, the capital expenditure for the relevant period is established as the total cost to purchase and install the station. This is a positive value to match the definition of expenditures. If instead a station is removed from the space and say sold, its estimated salvage value would be recorded as a negative capital expenditure to indicate a gain as opposed to a capital loss. The budget constraint parameter is defined as a percentage of the previous period's cash flow. Given that the first period does not have such a reference cash flow, an initial income flow must also be defined by the designer.

4.1.3.2 Horizon-Based Conditions

In addition to the period-based conditions, there are several conditions that are horizon-based. Horizon-based here denotes the modified horizon that results from the union of the original horizon and restructuring schedule. In other words, the scenario structure of before. These conditions are defined across the entire horizon and as such at each of the unique event time stamps of the modified horizon, [0, 6, 12, 24, 36] in the example from earlier. This means that at each of these time stamps, the conditions are to be defined by the designer. This data structure is where the nominal evolution of the conditions is captured. This is often where the scenarios will greatly deviate from one another assuming the designer desires to consider different conditions across the scenario set. Figure 16 below provides a graphical depiction of the production rate's definition, by the designer, for the example scenario structure presented earlier.

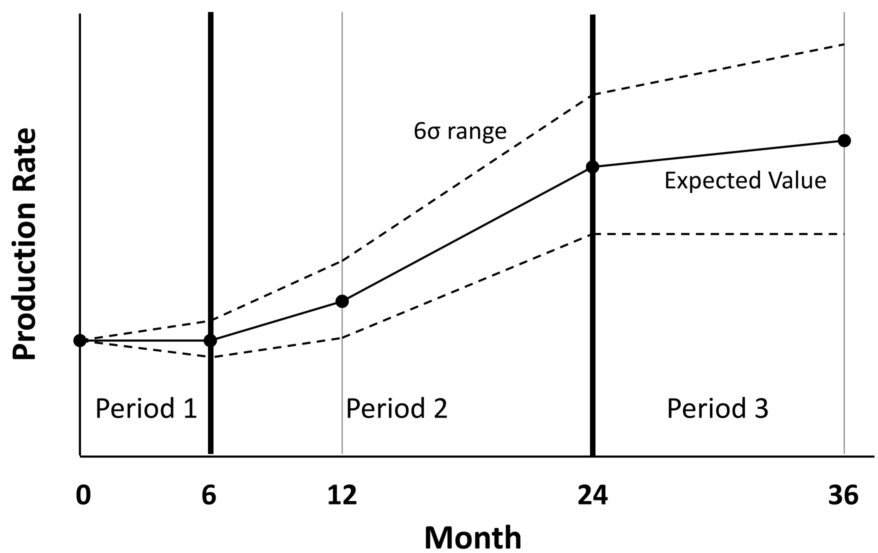


Figure 16 – Horizon-based condition definition example

In this case, the standard deviation of the production rate condition has been overlaid to illustrate how the independent definition of these two conditions combines to define how the local uncertainty about the production rate is captured by the developed robustness model. The solid dark vertical lines indicate period, or evolution, boundaries while the dotted lines sandwiching the solid expected forecast line depict the 6σ ranges for the production rates. The other horizon-based conditions do not have such ranges as there are no standard deviations defined for them in the developed performance model. As demonstrated by the solid dots, at each of these forecasting points, expected values for the conditions are prescribed by the designer, thereby defining the conditions across the entire horizon. The linear solid lines between the values depict how the production rate and all other conditions behave across the forecasting segments. In this research, it is assumed that, apart from a few conditions, the conditions behave linearly across the segments. This assumption was implemented to simplify the mathematical integration of these conditions across the horizon, which will be discussed when the performance model is presented later. The conditions being the exception to this assumption are the following: the labor cost adjustment factor, work days per week, and work hours per day. These conditions are assumed to be discrete across the segments where the value at the beginning of the segment defines the value across the segment.

As for those conditions that do behave linearly, they also are defined on a product-basis. In other words, for each product-process included in the scenario and as defined before, the following conditions are to be defined by the designer across the entire horizon: the desired expected production rates, the coefficient of variations of the production rates (i.e. ratio of the standard deviations to the expected production rates), the setup rates (if applicable), the market value, the estimated total manufacturing cost, and the direct consumable cost of producing the product. Again, for each product present in the defined scenario, a forecast such as that depicted in Figure 16 is required. An explanation of each of these conditions, along with their unit definition can be found in Appendix C. Definition of the production rates in this fashion enables the designer to control not only the total production rates of the products, but also the relative mix of the products and moreover which products are to be produced. Additionally, it is worth noting that the production rates coefficient of variances establishes a core input of the local robustness method to be highlighted later in this chapter while presenting the performance model developed in this research.

There are also three additional conditions that are defined on a horizon-basis. These conditions however are defined on a station-basis as opposed to a product-basis. Moreover, these conditions are defined discretely across the segments, just like that of the work days, hours, and labor adjustment factors. These conditions include the fixed costs of installing a station, the cost of displacing a station by a unit distance, and the cost of a station's support conduit. These conditions are to be later leveraged when establishing the costs of rearranging the layout. An explanation of each of these conditions can be found in Appendix C**.**

4.1.3.3 Process-Based Conditions

There are several process-based conditions that must also be defined by the designer for each scenario. These conditions are defined for each unique process present across the entire scenario (i.e. all periods). These conditions are decomposed into two types, those associated with the between-station segments of the process and those at the stations of the processes. The first type, those associated with the between-station segments, or handling segments of the processes, are defined on a segment-basis. For each segment in each process the following inputs require definition: the handler flow-rate capacity, average handler labor rate, number of handlers, and other per unit handling costs. As an example, if a process consists of a product visiting three stations, then two segments would need definition by the designer for the above input conditions.

As for the second type, those associated with the stations themselves, these are defined on a process-station-basis. These inputs include the following: the capacity of each station to produce each relevant product and the setup capacity of each station. Remember, these are defined on a unique process-basis which correlates to a specific product. As a result, the designer under this approach has the capability to define the setup capacity and station capacity more-or-less on a product-basis too. For a more expansive coverage of these conditions the reader can refer to Appendix C.

With the designer's definition of these process-based conditions, the market and business model condition inputs encapsulated in the LIVE methodology are complete. Moreover, the definition of the scenario itself is completed. Though it may seem like several inputs are required, most of these inputs should be able to be defined relatively easily by the designer. Many of the inputs are high level and easy to quantify by observing the system or the market.

4.1.4 Analysis Parameter Definition

Before the scenario problem can become completely defined, how the analysis of the system should be performed needs to be established by the designer. In the LIVE methodology, the designer has the choice of several different options pertaining to how the analyses of the performance and constraint models should be performed. Many of these enable the designer to consider additionally business-strategies while establishing the scenarios that will ultimately be leveraged to decide on the layout design to implement and moreover the operational approach.

The first option available is whether to run an analysis of the system or an optimization for the defined scenario. As a designer it may be useful to analyse the baseline layout design for the expected conditions before running any sort of optimization. Perspective can be gained by doing so and moreover a baseline for comparison can be established. Insights on utilization levels and layout performance can also help inform the need for additional scenarios. As such, this option was implemented to provide the designer with this capability. It is also one that will be leveraged later in this dissertation during the experimentation.

In the case where the optimization mode is chosen by the designer, which is the default one for running the scenarios, the designer also has a choice as to whether to run the first optimization stage only, the first and second stages sequentially, or even the option to load already established Stage One results and then proceed to just run Stage Two. As will be observed later, it may be useful for a designer to only run Stage One to more quickly solve the set of scenarios and populate the design space.

The designer also has the option to select whether the analysis should assume an existing layout, whereby it must be restructured to achieve the first period layout, or if no such layout exists and a completely new facility is being designed. This option provides the designer with the ability to solve what the literature refers to as either a brown or green layout problem.

The LIVE methodology also provides the designer with the ability to consider different analysis methods and solution techniques. The first and most relevant one is the ability to define the material handling costs by either a traditional rectilinear or the novel advanced flow distance method which will be covered later in this chapter. Moreover, the designer has the flexibility of prescribing which to deploy in both solution stages of the methodology. As such, the designer can choose to leverage the rectilinear approach in the first stage while employing the advanced method in the second stage. This combination also happens to be the default and moreover recommended as will be established later during the experimentation. In addition to the option to deploy different MHCs methods, the designer may also choose how to handle budgetary and boundary constraint violations. These options will be elaborated on more when the developed constraint model is discussed. Lastly the designer also has the option of defining how the performance model should handle situations where the system cannot sustain the defined production rates of before. This option relates to how the developed model dynamically adjusts the production rates to account for this. This will too be elaborated on later when the performance model is presented.

4.1.5 Optimization Parameter Definition

The last of the inputs that requires definition by the designer are the optimization parameters. These include the parameters associated with the genetic algorithm of the first stage and the tri-population genetic algorithm of the second stage along with the fast-simulated annealing algorithm implemented in the Stage One genetic algorithm to enhance its performance. Recommendations on how these parameters should be set is provided in a later chapter. Two experiments provide context on how to best establish these parameters under different problem characteristics and design choices. The full list of the optimization parameters can be found in Appendix C.

With the conditions for each scenario defined along with the analysis and optimization parameters, which will be used in solving each of the scenario problems, the next step in the methodology is to solve each of the scenario layout design problems as defined by the designer. The algorithms developed to achieve this solution and identify the best layout designs for each of the scenarios is presented next; starting with the first stage of the bi-model multi-stage approach developed in this dissertation.

4.2 Step 2: Solution Procedures of Stage One

This section outlines the developed algorithms and methods of Stage One. It begins with a brief overview of the mathematical model leveraged in Stage One to represent the layouts, followed by an understanding of how the physical layout is established from this model's data structure. The design variables of Stage One are then established followed by a detailed discussion on how the model's data structure was exploited to improve how the solution procedure searches through combinations of these design variables such that feasible layout designs are identified more efficiently. This discussion details the novel feasible sequence-pair promoting method (FSPPM) and its construction. Lastly, an expansive discussion on how improved layout designs are sought and subsequently discovered through the manipulation of the design variables is presented. This more broadly encapsulates the GA solution procedure developed to solve the layout problem of Stage One. This discussion consists of the following: how the GA's population is initialized by leveraging the novel FSPPM, how this population is then evolved through the employment of genetic operators tailored to the mathematical model employed, and lastly how FSA is implemented to provide improved solution performance. With the model chosen to mathematically define the layout being at the core of how all other elements of Stage One were constructed, the mathematical model deployed in Stage One is first presented.

4.2.1 Mathematical Representation of the Layout

To represent the layout mathematically (i.e. establishing the position of the objects in the space) in Stage One, Tang et. al.'s Fast Sequence Pair (Fast-SP) QAP mathematical model is implemented [160,161]. As mentioned before in the background section of this dissertation, such a discrete representation of the layout can be leveraged in Stage One since the primary objective here is merely to adequately populate the initial populations of Stage Two's GA solution procedure, rather than accurately capturing continuity in the layout. To follow, a brief outline of Tang's Fast-SP model is presented, which highlights only the elements necessary for a fundamental understanding of the model deployed. These fundamentals will become important to understanding later derived methods, such as the FSPPM, employed in this research. For a more thorough expansion of the formulation, one may refer to [159-161].

4.2.1.1 Mapping a Sequence Pair to a Physical Layout

As mentioned in Section 2.2.1.2, the sequence-pair representation is a QAP formulation of the layout problem employing a meta-grid data structure. As the name implies, this data structure consists of a pair of sequences of *n* objects, with each representing a unique object to be placed in the space. The following encoding relationships are imposed upon this structure to establish the relative positioning of blocks to one another and ultimately their positions in the physical space:

$$
\begin{array}{lcl}
(\langle ..b_i..b_j..\rangle, \langle ..b_i..b_j..\rangle) & \rightarrow & b_i\ is\ left\ of\ b_j \\
(\langle ..b_j..b_i..\rangle, \langle ..b_i..b_j..\rangle) & \rightarrow & b_i\ is\ below\ \ b_j
\end{array}
\tag{1}
$$

where the first bracketed sequence in the parenthesis denotes the positive sequence and the second, the negative sequence of the order-based sequence-pair ($\langle \Gamma_+ \rangle, \langle \Gamma_- \rangle$). Such an encoding scheme in turn creates relationships among the objects that follow horizontal $G_h(V,E)$ and vertical constraint graphs $G_v(V,E)$. As an example, construction of the horizontal constraint graph is as follows where (V: vertex set, E: edge set):

- $V = \{s_h\} \cup \{t_h\} \cup \{v_i | i = 1, \ldots, n\}$, where v_i corresponds to an object, s_h is the source node representing the left boundary and t_h is the sink node representing the right boundary

- $E = \{(s_h, v_i) | i = 1, \ldots, n\} \cup \{v_i, t_h | i = 1, \ldots, n\} \cup \{v_i, v_i | \ i \text{ is left of } j\}$

where the weight of the vertex in the graph is equal to the width of the object i for vertex v_i, but zero for s_h and t_h. The vertical constraint graph can similarly be constructed, the only difference being the weight would instead be the height of the object rather than its width. Furthermore, both constraint graphs then become vertex weighted, directed, and acyclic in nature.

As a result of this outcome of the constraint graphs, it can then be shown that the length of the longest path to each object node from the source in each of the constraint graphs defines the coordinate positions of the objects. It can further be shown that the weighted sequence-pair enables more efficient placement of the objects to be achieved by leveraging a longest common subsequence (LCS) algorithm [160,161]. The length of the identified longest common subsequence of the SP in each dimension (i.e. x and y) then establishes the coordinate position of the objects. An example of this for the horizontal constraint graph is shown below in Figure 17. The oblique grid of the sequence pair $(\langle 4\ 3\ 1\ 6\ 2\ 5 \rangle, \langle 6\ 3\ 5\ 4\ 1\ 2 \rangle)$ is graphed in Figure 17 along with all possible common subsequence paths from the source node to the sink node. In this situation, the highlighted path $s_h \rightarrow 4 \rightarrow 1 \rightarrow 2 \rightarrow t_h$ corresponds to the common subsequence $\langle 4\ 1\ 2 \rangle$ of the sequence pair. Identifying the longest of these common subsequences enables the coordinate positions of each object to then be established for the provided sequence-pair. This process of identifying the LCS, and as a by-product the coordinate positions, is known in the literature as the placement algorithm.

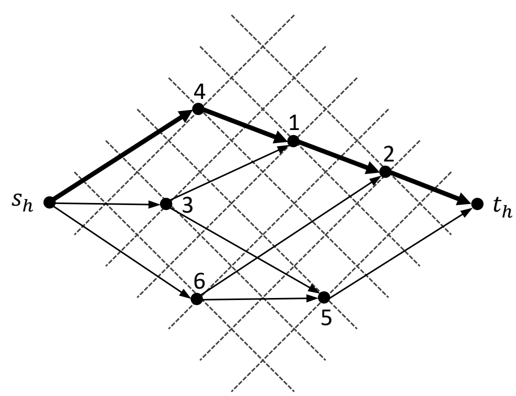


Figure 17 – Horizontal constraint graph with the path for the common subsequence $\langle 4\ 1\ 2 \rangle$ of the SP = $(\langle 4\ 3\ 1\ 6\ 2\ 5 \rangle, \langle 6\ 3\ 5\ 4\ 1\ 2 \rangle)$ highlighted

This approach to mapping a sequence-pair to a physical layout using LCS was first theorized and subsequently proved viable by [160,161]. Titled Fast-SP, Tang et. al.'s algorithm can determine the *x* and *y* coordinates of each object in the physical space in a literature best, *O(n log log n)* time. Furthermore, the Fast-SP algorithm, as they formulated it, is capable of handling constraints such as boundaries and fixed placed objects in the space. It is for these reasons that Tang et. al.'s Fast-SP approach was deployed in this dissertation to establish the coordinate positions of each object in the space from that of its sequence-pair. For a detailed mathematical decomposition of Tang et. al.'s placement algorithm one may refer to [160,161]. A slightly modified version of their placement algorithm was deployed in this dissertation. The algorithm was adjusted to enable extra control of where the bottom-left stacking origin position was located. In Tang et. al.'s formulation they assumed this to be the absolute origin (0,0), whereas in the formulation of this dissertation it can be placed elsewhere if desired. Moreover, the widths and heights supplied to the algorithm were defined not as the physical boundaries of the objects, but instead as a boundary that accounts for additional spacing about them. These adjustments were made to implicitly account for spacing restrictions about objects as well as boundaries. The importance of the latter adjustment will be acknowledged later when the performance and constraint models of this dissertation are presented.

4.2.1.2 Design Variables of Stage One

<u>Position Variables:</u>

With the coordinate positions established relative to a provided sequence-pair, the sequence-pair itself then becomes the positional design variable in this stage. It is then the sequence-pair itself that is manipulated by the optimizer in Stage One to perturb the design of the layout in pursuit of the best available design(s). Remember, the sequence-pair is composed of two distinct sequences of length *n*, where *n* equals the number of objects in the space. Each object present in the layout constitutes two variables, one in each sequence. Thus, the optimizer has control over a total of 2(*n-1)* position design variables, where the minus one is a result of the last object to be placed, having no choice but to be placed in the last remaining position in the sequence. As such, increasing the number of objects in the space effectively increases the number of design variables in a linear (slope of two) fashion. For example, adding one additional object increases the number of design variables by two (as anticipated), two objects increase it by four, three by six, and so on and so forth. Though the constrained objects have fixed positions in the space, positioning in the sequence-pair is defined on a relative-basis. Thus, the constrained object variables must too be controlled by the optimizer. Assurance that these constrained objects fall appropriately in the space is a task partially handled by the assignment of dummy blocks in the placement algorithm that act in artificially shifting such objects, when allowed, to their fixed position in the space and also by the posterior constraint evaluation, which will be discussed later in section three of this chapter where the modeling of these constraints are detailed.

In general, after the optimizer has manipulated the position variables of the sequence-pair, the placement algorithm can then be executed to establish the coordinate positions of the objects in the physical space. This process of manipulating the sequence-pair and subsequent construction of the physical layout through the employment of the placement algorithm is notionally demonstrated in Figure 18 below. In this example, the two objects in the positive sequence-pair, highlighted red, are exchanged by the optimizer to produce the resulting sequence-pair shown to the right of the optimizer block in the figure. This effectively alters the oblique grid, though marginally as demonstrated. With the exchange occurring only in the positive sequence, the two objects exchange positions in the upper-left to bottom-right diagonal direction while remaining on the same grid line in the bottom-left to upper-right direction. In turn, when the LCS is discovered by the placement algorithm for this new graph, the resulting placement of the objects in the physical space is that shown in the bottom right of the figure. In this scenario, by exchanging the two objects in only the positive sequence, it can now be observed that in both sequences the number five object precedes the number two object. From the established encoding relationships defined earlier, it is known that when this occurs, the object that comes first in both sequences is to the left of the other object. In this case, object five comes first, therefore this relationship establishes that object five is then left of object two. As demonstrated in the figure, this is the case as object two has shifted from atop object five to now being right of it in the physical space.

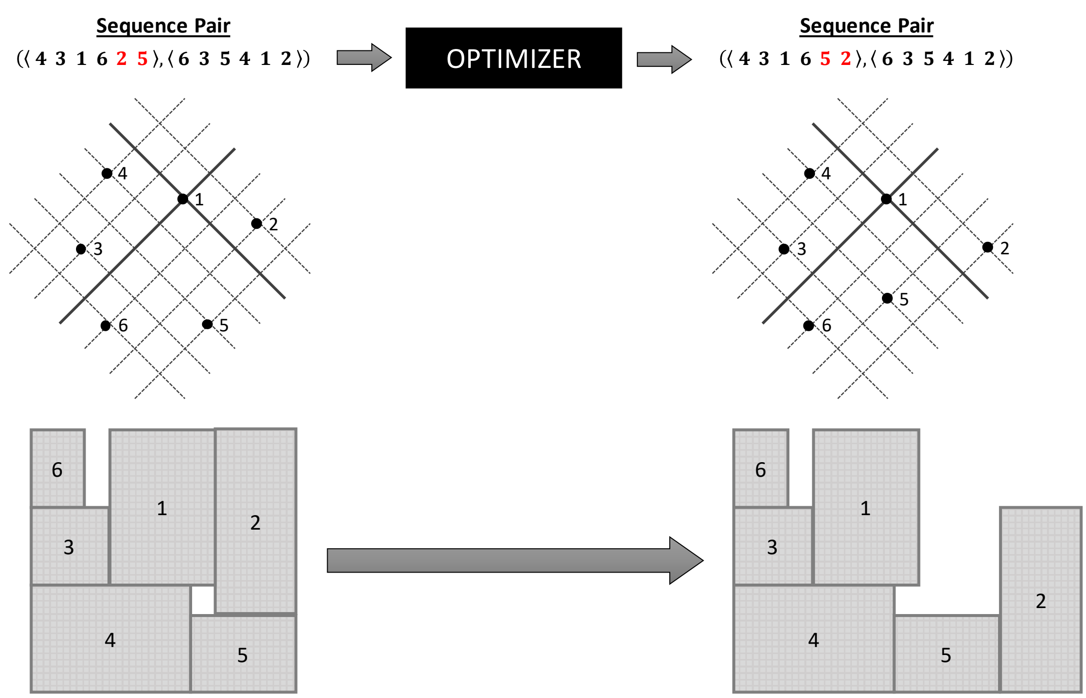


Figure 18 – Notional example of altering the layout by sequence-pair manipulation

Orientation Variables:

In addition to the optimizer in Stage One having control over the relative positions of the objects in the space through the manipulation of the layout's sequence-pair, it also has control over the orientation of the objects. Further, since the orientations of the objects define their dimensions in the coordinate directions, the orientation of the objects must first be established before then deploying the placement algorithm detailed before. If rotated, the width and heights could be interchanged depending on the orientation of said objects.

As for the orientation of the objects, each object has four distinct orientations under the formulation of this research. In addition to the default orientation, which corresponds to a rotation of zero, there is a 90-degree, a 180-degree, and finally a 270-degree rotation of the object that is possible. To characterize these four possible orientations, each object has an orientation-pair, consisting of two binary variables, representing its physical orientation. Unlike that of the sequence-pair, the orientation-pair consists of variable pairs (two binary variables for each object) that are independent of one another. As such, objects whose orientations are constrained (i.e. constrained objects), need not be a concern of the optimizer's. Therefore, though there are $2n$ orientation variables describing the orientations of the n objects, the true number of orientation variables controlled by the optimizer is equal to just twice the number of free/movable objects in the space. At most there could be $2n$ for the optimizer to control, this being the case where all objects are free/movable in the space.

An assumption of this formulation is that the I/O point of each object is assumed to be on the bottom edge of the object in its default 0-degree rotation state, as shown in the top-left illustration in Figure 19. As the object is rotated through the orientation variables, this point remains on this original edge. As such, each orientation is unique. For example, although the top-left orientation is identical to the bottom-right orientation in terms of its dimensions in each coordinate direction, the I/O points fall on opposite sides of the object. In the absence of including I/O points, only a single orientation variable would be sufficient in characterizing these orientations as the two diagonally opposite orientations would collapse into a single unique orientation. This is not the case in this formulation, thus why a pair of orientation variables for each object are required. How the I/O point positions and the object properties of height and width are derived from that of the design variables, one may refer to Appendix D.

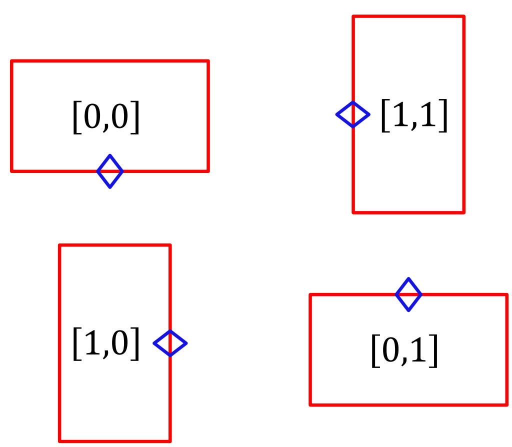


Figure 19 – Possible orientations of the objects and their corresponding orientation-pairs

The sequence and orientation pairs collectively constitute the design variables of the Stage One optimization. Manipulation of these variables by the optimizer enables the layout to be altered in pursuit of alternative and ideally better performing layout design(s). In total, and at most, there are $2n+2(n-1)$ variables that the optimizer must handle. An addition of any objects to the space increases the dimensionality of the problem linearly by a factor of four, though it could be by just two if these added objects are to be constrained in nature. This concludes the discussion of the design variables that define the layout design geometrically in Stage One. With that the discussion turns now towards how the optimizer can more frequently establish the sequence-pairs such that the resulting translated design is feasible from a constrained object placement perspective.

4.2.1.3 The Difficulty of Identifying Feasible Sequence Pair Designs

Traditionally when handling the sequence-pair formulation of the problem, unguided (i.e. random) search methods are employed to perturb the design (i.e. alter the sequence-pair). For example, Tang in his research uses a purely random placement method when generating neighboring sequence-pair designs [159]. What is most problematic with such approaches is that as the layout white space decreases, the number objects with fixed placements increases, and the total number of objects increases, these approaches start to labor in discovering the very limited number of feasible designs that are available. This in turn leads to them becoming extremely inefficient as will be demonstrated through experimentation presented later in this research. Again, this is intuitive as the procedure is effectively attempting to discover a few needles in a haystack while at the same time being blind folded. Without a systematic approach to more frequently identifying feasible sequence-pairs, such approaches lack an informed direction and therefore waste ample time searching without discovering. As a result, excessive search times are observed. Furthermore, failure to discover feasible designs efficiently can lead to subpar optimization performance, especially when subject to restrictions in run time.

It is for these reasons that an improved method of discovering feasible sequence-pairs more effectively was originally sought. It was believed that a method, which could promote the discovery of feasible sequence-pairs more frequently, would both improve convergence properties (e.g. reduced search times) and optimality discovery. This belief was the root motivation behind the development of the Feasible Sequence-Pair Promoting Method, or FSPPM, to be outlined next.

In the preceding section, placement of the objects in the physical space, from that of the sequence-pair, was discussed from a high-level perspective, intentionally avoiding the more mathematical details behind the process. The reverse process of mapping a physical layout to a sequence-pair is also important to understand, as in doing so one gains insight into how the Feasible Sequence-Pair Promoting Method, FSPPM, was derived. This process is detailed in Appendix B.

4.2.2 A Novel Feasible Sequence-Pair Promoting Method

To construct the Feasible Sequence-Pair Promoting Method (FSPPM), the outcomes of the gridding rules and preliminary observations outlined in Appendix B were leveraged. With the constrained objects in the space being one of the root causes which limits the number of sequences that are feasible, the FSPPM's emphasis is on first assigning these objects to the SP before then placing the free, or movable, objects. Considering the observations cited in Appendix B regarding the placement of said constrained objects, a statistical distribution approach was adopted in order to establish the placement of each constrained object in the sequences of the SP. The question that needed answering was then, how were these placement distributions (i.e. expected values and variances) to be defined for each constrained object? To answer this question, it was first assumed that said placement distributions would behave normally, an assumption supported by an earlier noted observation and one in which will be validated in a later experiment. A normal distribution avoids inducing any placement bias about the expected placement position in the sequences making it a favorable assumption. With the normal assumption in place, methods for defining the expected placement and variation about this expected placement were all that were then required to define said distributions. It is these distributions that will then be later leveraged to generate feasible sequence-pairs more efficiently.

4.2.2.1 Establishing the Distribution Variations

To define the variation about the expected placement, a global standard deviation parameter was implemented that would apply to all constrained objects in the space. This parameter is user defined and can be altered as desired, enabling the user to retain control over the diversity of the method. Prescribing a lower sigma value will force the distribution to be tighter about the expected value and therefore diminish diversity while a larger sigma will do just the opposite, instead promoting more diversity and exploration. A recommended value for this sigma parameter will be provided later during the experimentation, which analyzes how to define this parameter appropriately. With the method for defining the variation of the distributions since established, the method developed for defining the expected placement positions of the constrained objects within the sequences is now outlined.

4.2.2.2 Determining the Expected Placement of the Constrained Objects

The method developed for determining the expected placement positions of each of the constrained objects within the sequences leverages the observations and implication of the gridding rules highlighted in Appendix B. In light of these, it was concluded that the distribution's expected value (i.e. expected placement position in the sequences) be a function of the constrained object's normal distance to the appropriate bisecting diagonal of the physical space. Further, it was observed that this function should be structured such that as an object's normal distance to the appropriate bisecting diagonal approaches that of the absolute corner normal distance, the probability of the object being placed at the associated end of the corresponding sequence becomes greater and vice versa. Additionally, placement in the negative sequence should be based upon the normal distance to the upper-left to bottom-right corner bisecting diagonal line where as the placement in the positive sequence should be relative to the bottom-left to upper-right bisecting diagonal line. This relationship is notionally demonstrated below in Figure 20, where SPP and SPN denote the positive and negative sequences of the SP respectively.

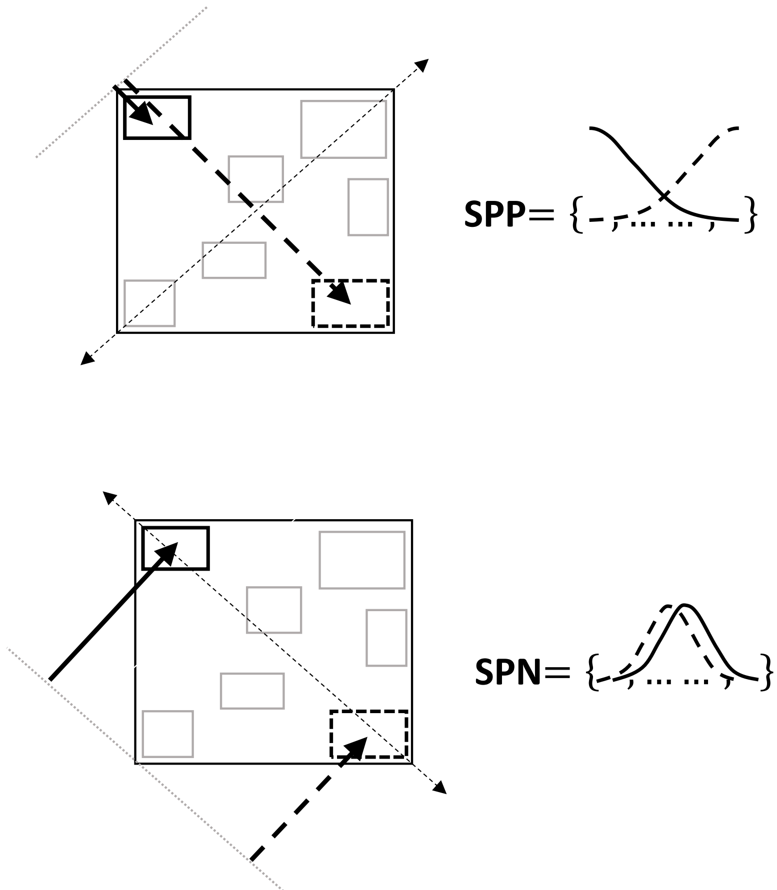


Figure 20 – Physical placement versus placement in the sequence-pair

Now that the general behavior and characteristics of the expected placement function have been revisited, the actual method developed to emulate this behavior is discussed. The method for defining the expected placement position in the sequences leverages the user-prescribed coordinate centroid positions (established during the initial problem setup process) of the constrained objects as reference points for defining the normal distances for each object to the respective bisecting diagonals of the space. With the corner points of the space known (C0, C1, C2, C3), equations for the two bisecting diagonal lines can easily be formed as a function of these points. Then, the explicit equation for the distance between a line and a point can be leveraged to calculate the normal distances. This equation can be generalized to be a function of the end and start points of the bisecting line (P, Q respectively) and the coordinated centroid position (R) of the constrained objects as demonstrated below in Equation (2).

$$ND = \frac{\left|(Q_x - P_x)\left(P_y - R_y\right) - (P_x - R_x)\left(Q_y - P_y\right)\right|}{\sqrt{(Q_x - P_x)^2 + \left(Q_y - P_y\right)^2}} \tag{2}$$

where P and Q would be set equal to C3 and C1 respectively when calculating the normal distance to the negative bisecting diagonal line and to C2 and C0 respectively when calculating the normal distance to the positive bisecting line. The definition of these corner points relative to the physical space is as follows: C0 is the bottom-left corner, C1 the top-left, C2 the top-right, and C3 the bottom-right.

With the normal distances from each constrained object's centroid position to both diagonal lines of the space computed, yielding a total of $2N_f$ normal distances where N_f is the number of constrained objects present, the mostly likely to appear position of each object in the space from that of the two diagonals is effectively known. In order to transform these into expected placement positions in the sequence pairs, which are non-dimensional ordered sequences, these dimensional distances needed to then be normalized and subsequently translated before the expected placement positions in the sequences can become known.

Normalization of these distances is achieved by determining the normal distances from each bisecting diagonal to the two remaining and opposite corner points of the space (i.e. those two points not a part of the bisecting lines definition). For example, if computing the negative bisecting diagonals corner point distances, the corner points of C2 and C0 would become the two opposite corner points and in turn the reference points used to compute the normal distances using Equation (2). Note, R would then become the corner points of the space rather than the constrained object centroid positions. Due to the nature of the rectangular space assumed in this research, these corner points are also effectively the maximum normal distances from the bisecting diagonals, in other words the absolute corner normal distances. Adding the normal distances on each side of the diagonals together then yields the maximum normal distance (MND) between the two corners. This maximum normal distance becomes the normalizing constant which reduces the constrained object normal distances found earlier to non-dimensional quantities. Before normalization can be performed though, the constrained object normal distances earlier need to be modified to align with the data structure of the sequence array variables.

The sequences of the SP are arrays of gene positions ranging from 1 to N. For the negative sequence, the first gene position can be thought of as aligning with the bottom-left corner of the space and the last with the top-right corner as notionally demonstrated in Figure 21 where the middle of the sequence array aligns with the negative bisecting diagonal line. This overlay is an accurate visualization of the conclusions made earlier regarding the relationship between the normal distances to the bisecting diagonal lines to that of the placement of the constrained objects in the sequences. For example, a constrained object placed in the top-right corner of the space would likely fall in the fifth position of the negative sequence, likewise objects falling about the negative bisecting diagonal line would be expected to appear in the second through fourth positions, i.e.

middle positions. Now with the first position of the negative sequence being representative of an object placed in the bottom-left corner (positive sequence being the top-left) of the physical space, the normal distance-based function needed to be anchored accordingly before normalization could be performed.

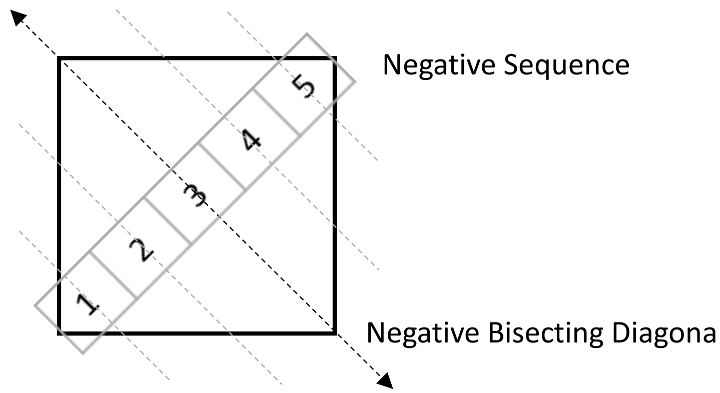


Figure 21 – Placement in negative sequence relative to negative bisecting diagonal

To anchor appropriately, the constrained object normal-distances computed prior are used to then compute the normal distances to the anchor points. The FSPPM determines this distance by using the normal distances to the bisecting diagonal lines, which are absolute values, along with a method of defining the direction the object is located relative to the diagonal line and combines it with half the maximum normal distances just found. The method of defining the direction the object is relative to the diagonal line leverages the sign function below (Equation (3)), which effectively produces a value of negative one when the object is placed on the side of the anchor point and a value of one when not.

$$S = sign\Big((Q_x - P_x)(R_y - P_y) - (Q_y - P_y)(R_x - P_x)\Big) \tag{3}$$

The reason for this sign function is such that when combining the constrained object normal distances with that of half the maximum normal distances, the constrained object normal distances would be appropriately added or subtracted from half the maximum normal distances thereby yielding the correct normal distances from the object's centroid position to that of the anchor points. When the object is on the anchor side of the diagonal, the sign is negative and thus the normal distance would then be subtracted from half the maximum normal distance as it should be and vice versa.

Conversion of the constrained object normal distances to that of the normal distances to the anchor point is done for two reasons. First, the normal distances now span from zero (when an object is constrained at the anchor point corner) to the maximum normal distance values (when an object is constrained at the opposite point of the anchor point, or anti-anchor point. The major advantage of this is that by then dividing the object normal distances to the anchor points by that of the respective maximum normal distances (i.e. normalization constants), the resulting quantities are then normalized normal distances (NND) that range from zero to one. The second reason is that objects constrained near the anchor or anti anchor point will yield a normalized normal distance of zero or one, meaning they would fall at the beginning or end of the range. As is understood from earlier, this is analogous to saying the beginning or end of the sequence which aligns with earlier observations. This conversion of the constrained object normal distances (ND) to that of the normalized normal distances (NND) is mathematically depict below in Equation (4).

$$NND = \frac{\frac{1}{2}MND + S \cdot ND}{MND} \tag{4}$$

where the appropriate MND and combination of S and ND are used for each constrained object and sequence.

Before the expected placement positions of the constrained objects can be determined though, there remains one final step. The normalized normal distances, ranging from zero to one must be converted to range from one to N so as to align with the sequence data structure noted before. This is achieved with the equation shown below whereby the by-product is the expected placement position in the sequences.

$$\mu = NND(N-1) + 1 \tag{5}$$

Where (N-1) provides the span of positional values that make up the sequences and the +1 provides a shift such that the beginning of the range starts at one, i.e. at the first gene positions of the sequences. For each constrained object there are two NNDs, one for the positive and one for the negative sequence. This in turn yields a μ for both the positive and negative sequence.

4.2.2.3 Defining the Placement Distributions

Now that the expected value and variation of the constrained object placements in the sequences are known, the FSPPM generates normal distributions for each constrained object’s placement in both sequences of the SP. These distributions are further modified by the FSPPM in order to improve its ability to generate feasible sequence-pairs more frequently. After establishing the likelihood of placing each constrained object in each position of the two sequences, the placement probabilities are modified in two ways by the FSPPM when applicable.

These modifications seek to emulate two prior observations. Both pertain and therefore are only applicable to, objects appearing at or near the absolute corners. In these scenarios, the expected position would fall at or near the ends of the appropriate sequence. When near, but not at the absolute corner, a portion of the distribution will extend beyond the bounds of the sequence-pair. In other words, given the provided expected value and variation, the object could have a 25% probability of placement outside the bounds of the sequence pair (i.e. below a position of 1 or above a position of N). Since this is infeasible, it is remedied by placing this 25% likelihood of placement then on the nearest feasible position (which would either be at the 1 position or the N position). This emulates the observation that the closer an object is to the absolute corners, the more likely it is to appear at the ends of the appropriate sequences.

The second modification to the distributions relates to situations in which a constrained object falls at the absolute corners or the space. In this scenario, it was earlier observed that in the appropriate sequence, the constrained object would appear at the appropriate end of the sequence with 100% probability. For example, if a constrained object was positioned in the absolute bottom-left corner of the space, then it should appear at the beginning of the negative sequence 100% of the time. As such, for such constrained objects, the FSPPM further modifies the placement probabilities by placing 100% of the probability in the appropriate end position (would be the first position of the negative sequence in the previous example).

4.2.2.4 Placing the Constrained Objects in the Sequences

With these newly modified placement distributions at its disposal, the FSPPM can then place the constrained objects in the sequences. By placing said objects in the sequence first and moreover in positions within the sequences that they are more likely to appear in, there isn’t the chance that movable, or free, objects could occupy said positions. As a result, it has the expected advantage of enabling feasible sequence-pairs to be discovered more frequently. Once the constrained objects are placed only then are the movable, or free, objects assigned to the remaining sequence positions of the sequences. The process of fully generating a sequence-pair from scratch by leveraging the FSPPM will be outlined when the section on initializing the population for the hybrid GA implemented in this research is presented in a subsequent section. Provided that the computations are dependent on only properties known from the problem initialization step, it is important to note that the determination of the constrained object placement distributions can be performed directly following the problem initialization and more importantly prior to performing an exhaustive search of the design space using the hybrid GA. This saves substantial computational effort whereby the distributions can be computed once upfront rather than having to be continually computed each time a new sequence-pair is to be formed. This concludes the discussion on how the mathematical model’s data structure was exploited to construct the FSPPM with the goal of improving how the GA solution procedure eventually searches through combinations of the sequence-pair and orientation design variables such that feasible layout designs are identified more efficiently.

4.2.3 Architecture of the Implemented Hybrid Genetic Algorithm

Until now the focus has been on discussing how the sequence-pair is converted into a physical layout design and further how the sequence-pair data structure was exploited to construct the FSPPM, which can form these sequence-pairs with a greatly likelihood of them being feasible once converted to a physical layout design. Attention is now turned towards how these feasible sequence and orientation-pairs, i.e. designs, are formed by leveraging the FSPPM and further evolved to discover the most optimal design. In this research and as was noted in the previous chapter, a hybrid GA was implemented in Stage One to achieve this search for optimality.

The goal of the implemented hybrid GA solution procedure of Stage One, and any GA, is to form and search for new and feasible designs in the pursuit of more optimal, or in other words, superior performing designs. The sections that follow outline the procedures developed to initialize the population, subsequently evolve the population, and finally identify the most optimal solution, i.e. design. Before diving into each of these phases that compose the hybrid GA developed, the general process flow of the hybrid GA of this research is presented.

As is the case with any GA, the first phase in the GA is to initialize the population, or put alternatively, populate the initial population that will then be evolved. In the hybrid GA of Stage One the same is true. The FSPPM method is leveraged to better generate an initial population that is rich with feasible designs. Once the initial population is generated, the population is evolved through an evolutionary process. The developed evolutionary process deploys three genetic operators to evolve the designs of the population. To further improve the performance of the evolutionary process and GA, a fast-simulated annealing (FSA) technique was implemented. The FSA is applied to the fittest design in the evolved population to further alter it in the pursuit of further improvement of the design. It is the implementation of the FSA that makes the GA a then hybrid GA. Once the FSA is applied, the final phase in the developed hybrid GA involved establishing the presence of convergence. This phase considers time constraints, generational limits, and solution improvement to establish the state of convergence. When a state of convergence has been met, the process of evolving the population from one generation to the next is terminated and the current best solution is then identified as the "optimal" solution. Now that a high-level understanding of the developed hybrid GA's sequence of events has been provided, each of these three phases will be elaborated on in more detail.

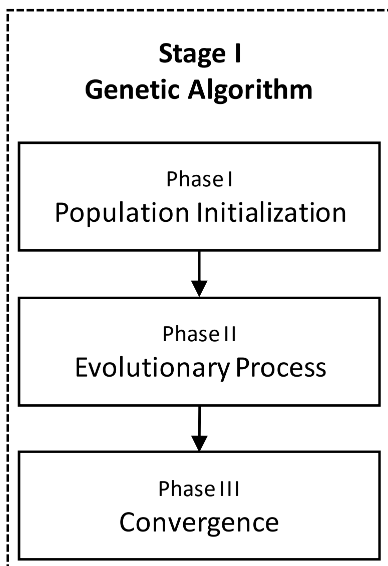


Figure 22 – Stage One genetic algorithm solution procedure

4.2.4 Initializing the Population

The first phase of the hybrid GA is to generate an initial population. Provided that the process of discovering the best solution via the hybrid GA originates from this initial population, it is imperative that it be populated with designs that will best enable the implemented hybrid GA to perform well. As a reminder, the importance of populating this initial population to the GA's performance was well established in the background section of this document. It was identified that the characteristics of this initial population can have a substantial impact on the performance of the GA. The most important characteristics cited were diversity and optimality. For this application a third, in that of feasibility, was also important to consider. As such, the method implemented to populate the initial population, for the Stage One hybrid GA, sought to balance simultaneously the diversity, optimality, and feasibility of the designs comprising this initial population.

To achieve this, the method developed leverages the FSPPM discussed earlier, random assignment techniques, and other measures to ensure this balance. The implemented process for generating this population consists of two segments. The first segment attempts to populate a user-defined percentage of the population with only feasible designs (those abiding by the spatial constraints, constrained placement constraints, among others) given a time restriction. The time restriction was implemented to avoid situations where an excessive, and potentially endless, amount of time could be spent in the population initialization phase of the hybrid GA process. This was implemented after observation of such situations during experimentation.

The second segment then populates the remainder of the population with designs that may or may not be feasible. In this segment the FSPPM is still leveraged; however, instead of requiring that the design be feasible in order to be assigned to the population, any design feasible or not is allowed. This effectively results in the first p designs being placed into the population, where p is the number of designs remaining to be assigned to the population following the first phase. Some of these designs may be feasible by chance, but there is no guarantee. Under conditions where the space is highly constrained, a large portion of these designs are likely not to be feasible. While the first segment ensures enough feasibility in the initial population, the second ensures diversity.

Now, regardless of the segment, a method for generating designs that could then be considered for assignment to the initial population was required. Up until this point, it has only ever been noted that the FSPPM was leveraged to generate said designs. How the FSPPM is leveraged in this research to generate new sequence-pairs is now finally discussed. Additionally, how the orientation-pairs are generated by the developed method to completely form a new design is also examined.

4.2.4.1 Generating Sequence and Orientation-Pair Designs

The method developed to generate a new design leverages the placement distributions generated by the FSPPM to establish the sequence-pair portion of the design's definition while the orientation-pair is established more generically using a simple random assignment technique. The method deployed for assigning the objects to the sequence and orientation-pair design variables is as follows:

1) First, a random sampling method is used to assign the constrained objects to the sequences of the sequence-pair using the placement distributions generated by the FSPPM, which establishes the probability of placement at each of the positions in the sequences

2) Then, the remaining sequence positions, or gene positions, are filled randomly with the movable objects

3) Lastly, the sequences of the orientation-pair are generated by random binary sequence generation

The usage of the FSPPM method to assign the constrained objects to the sequence-pair has the benefit of improving the discovery of feasible sequence-pairs thereby helping to sufficiently populate the initial population with feasible designs. At the same time, the random assignment of the movable objects and the binary sequences comprising the orientation-pair has the advantage of promoting diversity within the population. This concludes the discussion of how the initial population of the hybrid GA is generated and further how the FSPPM was leveraged to do so.

4.2.5 The Evolutionary Process of Stage One

Once the initial population has been formed, the developed hybrid GA then progresses into its evolutionary process, also often referred to in the literature as the generational loop. This generational loop constitutes the second phase in the developed GA process outlined earlier. In this phase, the initial population is evolved using several genetic operators. Given that the designs of Stage One are represented by a sequence and orientation-pair where the former is order-based, the genetic operators deployed needed to be designed to accommodate this chromosome representation, or data structure.

The evolutionary process of the developed Stage One GA consists of a sequence of six operations. Five of these operations are genetic operators, while the sixth is the

FSA technique noted earlier. Of the five genetic operators, two are selection while the remaining three are variation operators. The sequence of operations comprising the evolutionary process begins with first the execution of an elitism selection operator. Inclusion of this operator is meant to provide global generational selection, by transferring, unaltered, a specified number of the fittest designs from the previous generation to the current one.

Following the execution of the elitism operator the reproductive process begins. This process consists of the remaining operators (one selection and three variation operators) and continues until the next generation has been fully populated. First in the process the reproduction selection operator is executed whereby parent designs (i.e. sets of sequence and orientation-pairs) from the previous generation's population are selected for evolution via the variation operators. With parent designs selected a novel adaptation of the jumping gene operator is first applied to these designs to alter their compositions. Once applied these altered parent designs are then further modified via the more traditional GA variation operators of crossover and mutation. Not always are all three of these variation operators applied to the parent designs. Sometimes just one will be applied while other times two or all three may be applied. Each has a user-defined probability of occurrence. A study regarding how these probabilities should be prescribed will be presented later. Once the variation operators have or have not been applied, the new designs are evaluated for their feasibility and performance, both of which will be discussed in detail later. Designs that satisfy the feasibility property, i.e. constraints of the problem, are added to the current generation's population. This reproductive process continues until the generation is fully populated of feasible designs.

Once the reproductive process finishes the developed GA applies FSA to the fittest, i.e. best performing, design of the current population. After this fittest design has been potentially further improved by the FSA technique and then placed back in the current population, the evolutionary process for the current generation ends only to be repeated in the following generation. This complete sequence of operations comprising the implemented evolutionary process is depicted visually in Figure 23 below. Each of these operations and their applications to the problem considered in this dissertation are now detailed in ordered succession.

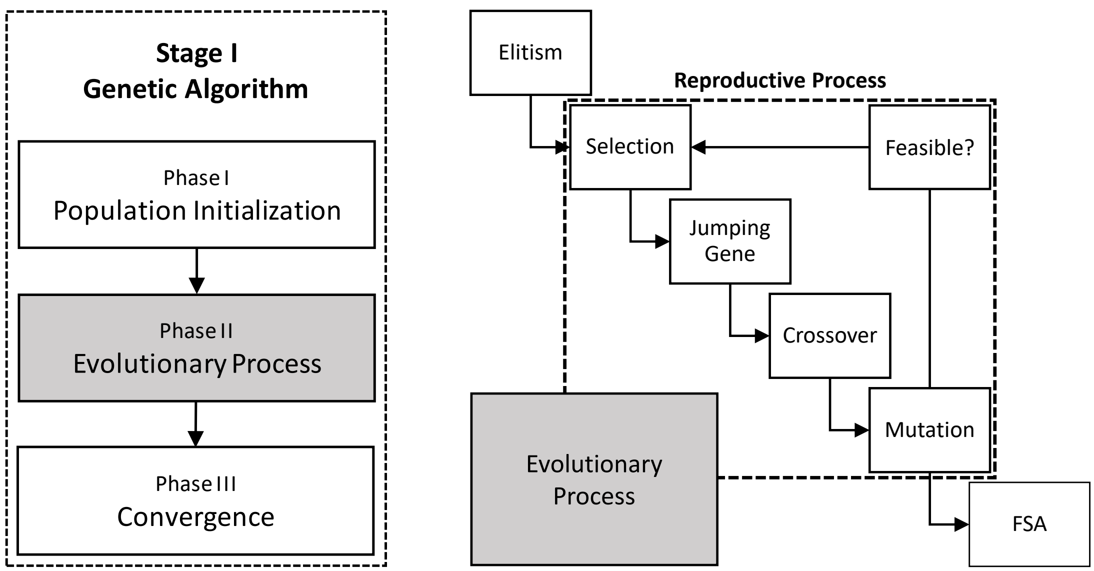


Figure 23 – Evolutionary process of the Stage One genetic algorithm

4.2.5.1 Elitism Operator

As mentioned before in the background, the elitism operator's primary function is to ensure the best individual(s) survive from one generation to the next. Since De Jong's (1975) original introduction of elitism, others such as Mitchell (1999) have established its ability to improve the GA's performance [126,127]. It is for this reason it was included in the developed hybrid GA of Stage One. As discussed and established in Table 5 of the preceding chapter, and represented in Table 6 below, to employ the elitism concept, Liu and Meller's *k* best pair-wise exchange heuristic approach applied to the SLP was *adapted* to the DLP for use in the Stage One GA's evolutionary process.

Table 6 – Stage One implemented genetic reproduction strategies

		Model
		QAP/U-SP
Selection Operators	Selection:	Liu and Meller's / Ulutas and Islier's roulette wheel selection [115,167]
	Elitism:	Liu and Meller's *k* best with revised improvement heuristics [115]
Variation Operators	Crossover:	Liu and Meller's modified uniform operator *adapted* to the DLP structure [115]
	Mutation:	Liu and Meller's mutation operator *adapted* to the DLP structure [115]
	Jumping Gene:	Ripon et al.'s cut and paste and copy paste operations *adapted* [143]

4.2.5.1.1 Elitism Process

Liu and Meller's elitism operator initiates by first selecting the *k* best individual(s) from the previous generation (if the first generation, then these come from the initial population) before then applying the pair-wise exchange improvement heuristic to the best individual of these *k* individual(s). This potentially improved best individual and the other *k-1* best individual(s) are then copied to the next generation [74]. The elitism operator deployed in this research mirrors this process employed by Liu and Meller. The difference between it and Liu and Meller's lies in how the pair-wise exchange heuristic proceeds in improving the most fit individual.

4.2.5.1.2 Pair-wise Exchange Improvement Heuristic

Liu and Meller's approach was originally applied to the SLP and as such utilized a gene-based pair-wise exchange improvement heuristic. After attempting all possible exchanges, the most improved of these becomes the basis for which all possible exchanges are then again considered, constituting yet another pass of the exchange procedure. This process continues until no further improvement of the design is discovered. Adopting this same approach to the DLP requires significantly more overhead however, as gene-based exchanges would then need to be performed across each of the *P* periods comprising the DLP. The number of exchanges required given a DLP of *D* departments and *P* periods is equivalent to the equation provided in the second column of Table 7 for the exchange method labeled "Liu and Meller's." With this number becoming exponentially larger as the number of periods, *P*, increases, an alternative approach was required given the potential overhead associated with evaluating each of the designs yielded by these exchanges.

To adapt this procedure, several alternative exchange methods were considered. Option one being, to perform gene-based pair-wise exchanges on each of the periods independently. This approach reduced the number of exchanges from scaling exponentially with the number of periods to scaling just linearly. Although a noticeable improvement in comparison to Liu and Meller's approach the number of required exchanges to be evaluated remains quite high, however. The second alternative, option two, was to perform gene-based pair-wise exchange on a period selected randomly or based on each period's proportional contribution to the cost function with the larger contributor being selected. This exchange approach effectively reduces the application of the operator to that of an SLP thereby matching the number or required exchanges Liu and Meller's original formulation had to perform for the SLP they considered.

The latter selection process has the advantage of focusing efforts where they are most needed making it the preferred of the two variants noted thus far. This focus is also its major drawback. Such a focus can prevent it from providing global improvement of the design. This leads to the last option, option three, which attempts to provide more global improvement of the design compared to option two. Option three was to perform a period-based pair-wise exchange procedure. Operating on a period-basis effectively interchanges the D's in option two's equation with P's. Since it is more likely for the period count, typically five, to be fewer than the department count, this approach further reduces the overhead associated with the elitism operator. This approach is independent of department count size which is advantageous at it will not scale as the problem size increases in this dimension, which is more likely to be the case. Although better at more effectively improving the design in a cross-period global sense compared to option two, it lacks the ability to improve the design beyond that of the period level and is therefore limited in that sense.

Table 7 – Overhead of elitism exchange approaches

Exchange Approach	Number of Exchanges	Nominal Example: Exchanges	Nominal Example: CPU Time (s)
Liu and Meller's	$\left(\frac{D(D-1)}{2}\right)^P$	> 184.5 million	> 1.8 million
Option 1	$\frac{D(D-1)P}{2}$	225	2.25
Option 2	$\frac{D(D-1)}{2}$	45	0.45
Option 3	$\frac{P(P-1)}{2}$	10	0.1

For a notional example of five periods and ten departments, a relatively small sized problem, the number of exchanges required at each pass of the various exchange heuristics are provided in column three of Table 7. Additionally, assuming evaluation of an exchange takes one thousandth of a second, the total CPU overhead that would result is provided in column four. As can be observed, Liu and Meller's approach would be intractable and therefore impractical to implement. Option one is far more reasonable, however, with more than one exchange pass procedure being required and further this operator being executed at each generation of the GA, even it would be impractical to implement. Ultimately, a procedure employing the combination of both options two and three was deployed. Combining the two enabled the superior global improvement of option three to be leveraged while also retaining option two's ability to improve the design where improvement was most needed. The process for the exchange improvement heuristic deployed in this research for Stage One is as follows:

1) The basis is set to be the best individual of the k best individual(s)

2) A pass of period-based pair-wise exchanges is first performed with the basis as the baseline individual, accepting the exchange with the most improvement

3) The basis is then set to be this newly accepted individual

4) Steps 2-3 are repeated until no further improvement is identified

5) Using the result of Steps 1-4, the basis is set to be this resulting individual

6) The period that contributes most significantly to the cost function is then determined

7) A pass of gene-based pair-wise exchanges on this period with the basis as the baseline individual is performed, accepting the exchange with the most improvement

8) The basis is then set to be this newly accepted individual

9) Steps 7-8 are repeated until no further improvement is identified

The resulting improved design of this pair-wise exchange improvement heuristic procedure then replaces the original best design of the *k* best individual(s) as noted before. This completes the discussion of the developed process employed by the elitism operator of this research for the Stage One's GA. Following this operator, the evolutionary process enters the genetic reproductive cycle, or reproduction process as established before.

4.2.5.2 Selection Operator

In the evolutionary process flow, the reproduction process follows the elitism operator and within this reproduction process the selection operator initiates the reproduction cycle. The selection operator's function is to select the individuals from the population generated in the previous generation that will then become parents for genetic reproductive purposes. As established before and presented in Table 6, Liu and Meller's, and similarly Ulutas and Islier's, roulette wheel proportionate selection technique employing linear scaling was deployed in this research to select the parents for reproduction. The method deployed to perform the selection process is as follows:

1) With the population established by the preceding generation as the basis, the best performing design from this population is identified and its fitness value, i.e. objective function, retrieved

2) The fitness values of the basis population's designs are then linearly scaled by subtracting each design's fitness value from that of the best performing design's fitness value identified in Step 1

3) Next, the summation of all the designs scaled fitness values is obtained

4) Then, using this summation, each scaled fitness value is divided by it in order to establish each design's likelihood of selection via a roulette wheel approach

The preceding portion of the selection process is executed once each generation and just prior to the reproductive process outlined before. This is to avoid unnecessary and redundant executions of these computations each time new parents are to be selected within the reproductive process loop. The second portion of the selection process, and one that is executed each time at the start of the reproductive process, leverages the probabilities established in Step 4 above to then deploy the roulette wheel selection process to select two new parents from the basis population (i.e. previous generation's population). This selection is achieved by randomly selecting two parents from the basis population according to the probabilities of selection established in Step 4. Those designs of the basis population with superior performance characteristics (i.e. a higher fitness value), have a higher probability of selection and vice versa.

4.2.5.3 Jumping Gene Operator

With two parents selected for reproduction, the reproduction process continues with the deployment of a jumping gene operator (JGO) before then deploying the more conventional genetic operators of crossover and mutation. The JGO is executed with a user-defined probability. In other words, it may not always be applied to the parents to promote evolution.

Since McClintock's first observation of the jumping gene phenomenon in nature, researchers, having observed its usefulness in promoting diversity and population evolution, have constructed GA operations that emulate this phenomenon. Its promotion of genetic diversity and evolution of the population through the horizontal transmission, in addition to the conventional GA's vertical transmission, of genes amongst two parents [94] enables a larger portion of the design space to be searched within a single generation. This in turn improves the evolution of the population and therefore performance of the GA. It is for this reason that the jumping gene operator was implemented in this research.

4.2.5.3.1 Jumping Gene Process

To emulate the jumping gene behavior, Ripon et. al.'s JGO applied to the QAP/S DLP was adapted to the QAP/U-SP formulated DLP subject to evolutionary changes in genome length (i.e. unequal number of objects present from one period to the next or put alternatively a scenario involving the introduction of a new asset into the environment). Ripon et. al.'s operator employs two jumping gene operations, cut and paste and copy and paste, both of whose concepts are adopted in this research for deployment in the Stage One GA. The general execution process is also adopted directly from Ripon et. al. and deployed as follows:

1) A copy and paste *or* cut and paste operation is randomly selected to be executed

2) The appropriate variant of the chosen operation, based on the parents selected for reproduction by the selection operator process, is then performed

As indicated in step two, two variants exist for each operation. One of these variants is employed if by chance the two parents selected are identical to one another, while the other is employed when the parents are found to be different from one another.

4.2.5.3.2 Assumptions

Before elaborating on the deployed operations, a few overarching assumptions and terminologies first need to be presented. The assumptions that govern the behaviour of the redeveloped operations of this work are synonymous to those presented by Ripon et. al., but with the first assumption having a subtle difference in wording to account for the potential evolutionary changes that are unique to the problem of this research. The assumptions are as follows:

1) A single transposon represents a single period

2) Transpositions can be made within the same chromosome (i.e. parent individual) or a different one AND can be more than one period in length

3) Transposition inserting positions are restricted to the starting gene of the period genome, genome being the collection of genes that make up a design

Assumptions (1) and (3) combine to result in the cut and copy operations working on a period-genome-basis not a gene-basis. In other words, period layout designs are swapped, shifted, etc. as a whole. This is an important concept to understand going forward.

Furthermore, in presenting the operations, the format will look much the same as Ripon et. al.'s formulation; however, there is one important difference. Unlike that of Ripon et. al.'s formulation, which was applied to the QAP/S formulation of the DLP, the letters presented in the diagrams represent sequences of both position pairs and orientation pairs (i.e. design variables) that define the layout design of that given evolution (cell). For complete understanding going forward, the first cell of the string of cells (i.e. a chromosome) is equivalent to the first evolution, the second cell equivalent to the second evolution, and so on and so forth. Similarly, the letters in the cells represent the sequences of both position and orientation-pairs defining the layout design, which is also referred to as a genome, and as defined in their origin chromosome. The latter will become clearer as examples of the individual operations composing the deployed JGO are demonstrated.

4.2.5.3.3 Cut and Paste Transposition Process

The following section details the cut and paste transposition operations implemented in this research. These processes are performed only when the cut and paste operation has been selected by the jumping gene process to provide variation.

Same Chromosome:

In the scenario when the selected parents are identical, the following process is performed to just one of the chromosomes:

1) First, the transposon length, whose max length can be up to *T-1*, where *T* is the number of period evolutions for the problem, is randomly selected

2) Then, a collection of sequential evolutions in the chromosome are randomly selected to represent this transposon

3) Next, an insertion position not encapsulated by the collection (choosing one within the collection will just reinsert the transposon in the same position it was originally, resulting in no alteration) is randomly selected

4) Then, the transposon is removed from the chromosome and the genomes between the transposon and the insertion position are shifted accordingly to fill the created gap made by the removed transposon

5) Finally, the transposon is inserted into the chosen insertion position

This altered chromosome, along with the other unaltered parent chromosome, then become the new parents. These new parents are those then used by subsequent genetic operators (i.e. crossover and mutation) to provide genetic variation within the population.

Figure 24 below demonstrates the above process for a chromosome consisting of seven periods, labeled *a* through *g*. In this example the transposon length was randomly chosen to be of length two and consisting of the sequential evolutions starting with *d*. In other words, evolutions *d* and *e*, as demonstrated in the first string of cells labeled PRT1 and denoting parent one, became the transposon. Next, the insertion point was randomly chosen to fall between evolutions *a* and *b*, but could have alternatively been chosen as preceding *a*, between *b* and *c*, between *f* and *g*, or following *g*. It could not however, have been chosen to be between *c* and *d*, between *d* and *e*, or between *e* and *f*, as it would result in no alteration as one can intuitively visualize. Then the transposon (*d* and *e*) is removed from the chromosome. To fill the gap created between it and the insertion point, the designs of *b* and *c* are shifted to the right by the transposon length (two here and coincidentally into the *d* and *e* design's original position). If the insertion point were to lie right of the transposon, then the shift would occur in a leftward or upstream direction. Finally, the newly formed gap created by the shifted designs by inserting the transposon *d* and *e* as demonstrated by the brackets and arrow from PRT1 to PRT2 below is filled in.

The result of this is the altered parent, now defined as PRT 2. PRT 1 would then just be the original chromosome, or in other words the other identical parent of the pair selected by the selection operator.

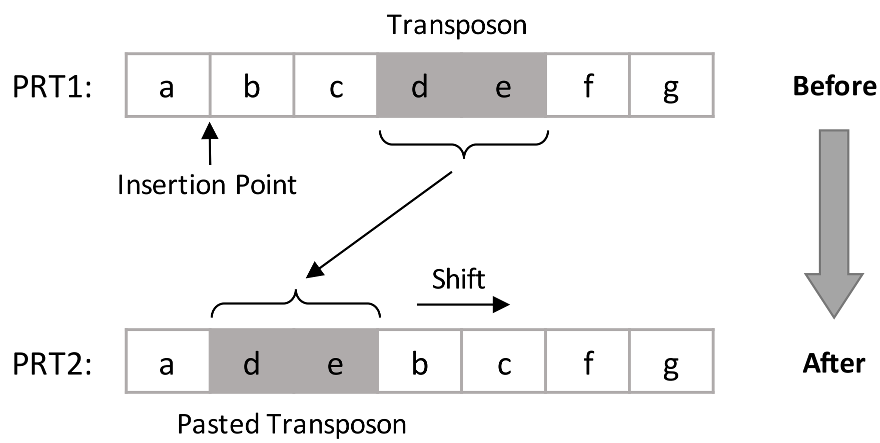


Figure 24 – Cut and paste transposition process for identical parent chromosomes

Different Chromosomes:

In the scenario where the selected parents are different from one another, the process is much the same as that described before, the major difference being that it operates across two parent chromosomes. The process is as follows:

1) Like before, the transposon length, whose max length can be up to *T-1*, where *T* is the number of period evolutions for the problem, is randomly selected

2) Then, a collection of sequential evolutions in each parent chromosome are selected randomly to represent the transposons

3) Next, insertion positions, not encapsulated by the respective collections in each chromosome (the preceding insertion point is acceptable though here), are randomly selected

4) Then the transposons from each chromosome are removed the genomes between the transposons and insertion positions of each chromosome are filled accordingly to fill the gaps created by the removed transposons

5) Finally, for each, the transposon of the other chromosome is inserted into its own insertion position

Just as before, these altered chromosomes become the new parents for future genetic operators to operate upon.

The figure below demonstrates the above process for the same problem as before. Note that here each evolution of the two parents has their own unique letter identifier, PRT1 spanning from *a* to *g* and PRT2 spanning from *t* to *z*. In this example, once again a transposon length of two was coincidentally chosen. Collections of *b* and *c* in PRT1 and *v* and *w* in PRT2 were chosen as well as insertion points preceding *f* and following *z* in PRT1 and PRT2 respectively. Here the insertion points are right of the transposons in each chromosome so the designs between the transposons and the insertion points are shifted leftward, or upstream, as was revealed in the previous example. In PRT1, *d* and *e* shift into the original positions of the transposon designs of *b* and *c* whereas in PRT2 x, y, and z all shift to the left by two evolutions, with *x* and *y* replacing the original transposon designs *v* and *w*. Then as the arrows demonstrate, the transposon from PRT1, *b* and *c*, is inserted into the insertion point of PRT2, following the shifted *z* design.

Likewise, the transposon of PRT2, *v* and *w*, is inserted into the insertion point of PRT1, that being preceding *f*. These altered parents then become the new parents going forward.

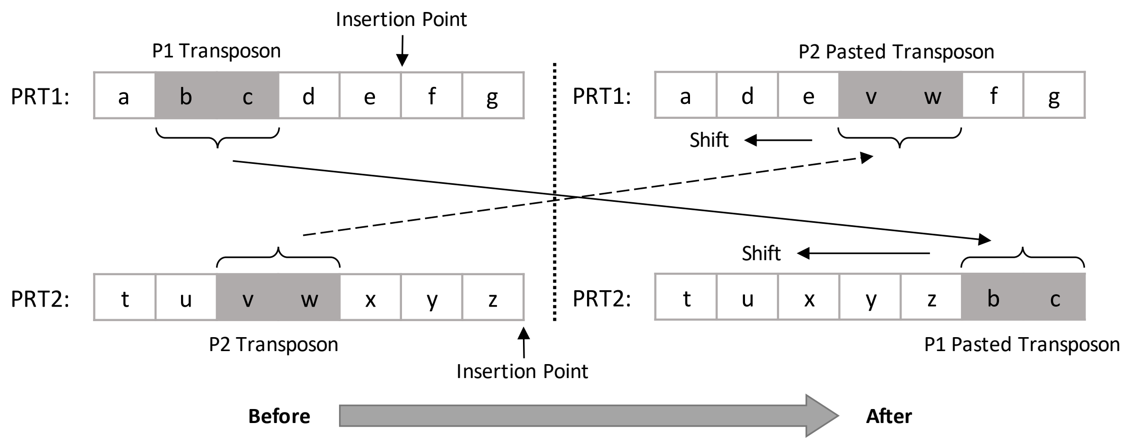


Figure 25 – Cut and paste transposition process for different parent chromosomes

4.2.5.3.4 <u>Copy and Paste Transposition Process</u>

The following section details the copy and paste transposition operations deployed in this research. These processes are performed when the copy and paste operation has been selected, over that of the previously detailed cut and paste operation, to provide variation.

Same Chromosome:

In the scenario where the selected parents are identical, the following process is performed to just one of the chromosomes:

1) Same as before, the transposon length, whose max length can be up to *T-1*, where *T* is the number of period evolutions for the problem is randomly selected

2) Then, the collection of sequential evolutions in the chromosome, that will represent the transposon, is selected randomly

3) Next, an insertion position in the chromosome that is not exactly encapsulated by the collection (choosing one within the collection will just reinsert the transposon in the same position it was originally, resulting in no alteration) is selected at random

4) Finally, the transposon is inserted at the insertion position, overwriting those genomes (or designs) and maintaining the transposon at its original location unless otherwise overwritten as part of the pasted transposon

The resulting altered chromosome and the original then become the new parents for use by future genetic operators.

The figure below, Figure 26, demonstrates the above process visually. Since the first few steps remain the same as those detailed in previous explained examples, to avoid redundancy they will not be restated here. In this example *f* and *g* were chosen as the transposon of length two and the insertion point as preceding genome *b*. The last step is then to paste this chromosome into the insertion point as demonstrated, overwriting the b and c genomes (i.e. those designs for evolutions two and three) with *f* and *g* genomes while retaining the *f* and *g* genomes in their original locations. In this example, this is analogous to saying that after pasting, evolutions two and six have the same layout design and so too do evolutions three and seven. This PRT2 along with the unaltered (i.e. other identical parent) chromosome become the new parents going forward.

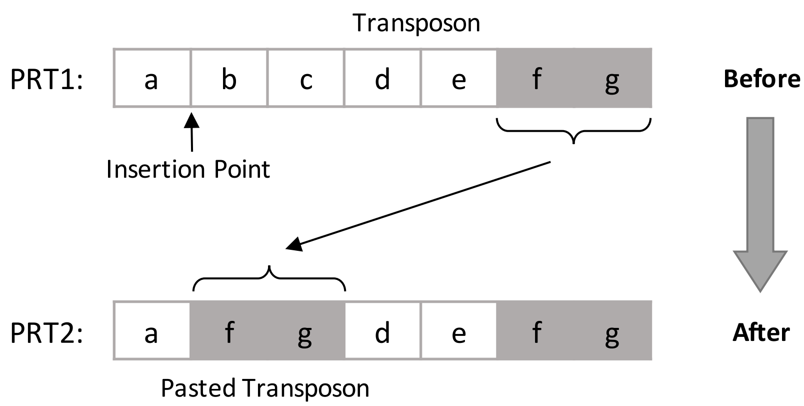


Figure 26 – Copy and paste transposition process for identical parent chromosome

Different Chromosomes:

In the scenario where the selected parents are different from one another, the process is much the same as that described before, the major difference being that it pastes the transposon of one parent into the second. The process is as follows:

1) Again, the transposon length, whose max length can be up to *T-1*, where *T* is the number of period evolutions for the problem, is randomly selected

2) Then, it is randomly selected from which of the two chromosomes the transposon will be taken from

3) Next, the collection of sequential evolutions in the selected chromosome identified in the previous step is randomly selected to represent the transposon

4) Now, the insertion position in the other chromosome (no restriction on those points not encapsulated by the transposon as pasting in the same position, but in the other chromosome will still result in alteration) is selected at random

5) Finally, the transposon is inserted at this insertion position overwriting those genomes while leaving the transposon derived chromosome unaltered

These two chromosomes then become the new parents going forward just like before.

The above process is demonstrated visually in Figure 27 for the same example problem defined several times before. Here the transposon was chosen once more to be of length two and moreover to be taken from PRT1. The insertion point was also chosen to be preceding *x* (i.e. evolution five) in PRT2. As demonstrated the transposon consisting of *b* and *c* was copied to that of PRT2 replacing *x* and *y*. In this situation, PRT2 now has the layout designs of evolutions two and three from PRT 1 in its (PRT 2) evolutions five and six, while its other evolutions remain unaltered. This chromosome (altered PRT2), along with the unaltered chromosome (PRT1) become the new parents going forward.

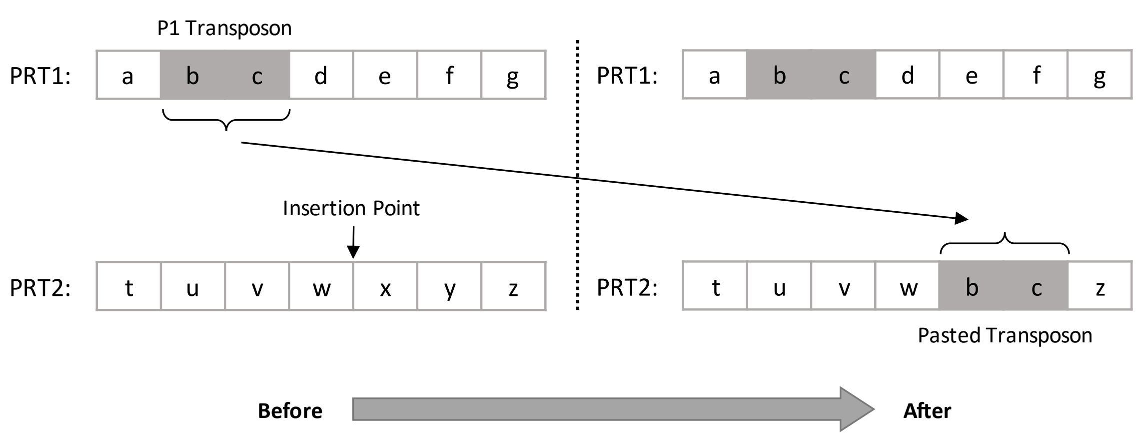


Figure 27 – Copy and paste transposition process for different parent chromosomes

4.2.5.3.5 <u>Genome Repair Process</u>

Under the problem formulation originally solved by Ripon et. al., the above processes, which have been adapted to the QAP/U-SP formulation, would alone suffice. In Ripon et.

al.'s formulation, it was implicitly assumed that no evolutionary changes in genome length occurred throughout the planning horizon. This establishes that each period evolution consists of the same number of objects. In other words, the length of the SP defining the *a* genome is the same length as the *b* genome, *c* genome, … , *t* genome, *u* genome, etc. This assumption, however, no longer applies to the unique formulation of the problem in this research where changes in these genome lengths can potentially occur. For example, if in the third evolution the decision was made to purchase a new asset, which would enable the business to expand their capabilities, this would result in the *a* and *b* genomes having a length one less than that of the *c*, *d*, *e*, *f*, and *g* genomes in PRT1. Same goes for *t* and *u* genomes versus the others in PRT2.

As one can foresee, these implemented jumping gene operations, which elicit alteration in the parents on a period evolution-basis, can potentially result in occurrences of inconsistent period genome length. This would be the case if during one of the transposition processes; two genomes of inconsistent length were exchanged. In the previous example, if the *b* genome was pasted over or in place of say the *d* genome or *y* genome of the other chromosome (arbitrary selections for demonstration), this would result in the genome of that period evolution having a length one fewer than what is required to sufficiently describe all objects present in the layout design for that period. In this case, where the transposon genome length is shorter than the original genome, the scenario of insufficient genetic material arises. On the contrary, the opposite can too occur, where the transposon genome length exceeds that of the original genome. This case characterizes the scenario of additional genetic material.

Again, these scenarios arise as a unique by-product of this research considering the assessment of evolving business models in parallel with an evolving layout design. To the best of the author's knowledge, no such application of the JGO to a QAP/U-SP DLP, or any DLP for that matter, with such evolutionary changes has been performed. As such, the development of a novel genome repair process was required to handle such occurrences. The process developed and subsequently deployed consists of two sub-processes, one for each repair scenario mentioned before. Before continuing, it should be observed that only the pasted genomes of the copy and paste operations and all shifted and pasted genomes of the cut and paste operations need be inspected for repair. Amongst these, only those resulting in inconsistent genome length need be repaired by the appropriate repair process that follows.

Additional Genetic Material Genome Repair Process:

The additional genetic material repair process is invoked in scenarios where excessive genetic material results as a by-product of a pasted transposon genome exceeding, in length, what is required to sufficiently define the objects present for that period evolution. Repairing this genome to be consistent in length with the original genome is a relatively straightforward process as all the necessary genetic material is already present, it merely needs to be systematically trimmed of unnecessary genes to match with what is required for that evolution. When repair of such a genome is required, the following process is performed for each period:

1) First, the additional genes in each of the genome sequences are identified (i.e. object indicator numbers greater than the original genome's maximum object indicator value in both the positive and negative sequences)

2) Then, these additional genes from the genome sequences are removed

3) Next, the remaining genes are shifted leftward in each sequence of the sequence pair to fill the created gaps formed by the removal of the additional genes in the previous step

4) Finally, removal/shift of the same genes within the orientation pair sequences is performed

The resulting genome is one that is consistent with the original genome length and thus with what is necessary to adequately define the layout design of the current period evolution. Further, the shifting in a leftward fashion aligns well with the general concept of packing a layout into the bottom-left corner of the space.

An example of this genome repair process is demonstrated in Figure 28, where the transposon genome was discovered to be of length six, two more than that of the original genome (i.e. length required to sufficiently define the period layout design). The object indicator numbers greater than the original genome's maximum object indicator value (genes with greater values defining them), four for four objects, were first identified in both the positive and negative sequences of the position pair. These include object pointers of values five and six as shown highlighted below in gray. Next, these genes were removed, and the remaining genes shifted leftward. For example, the first gray gene with the value of five is removed resulting in a gap in which all genes right of it then shift to the left one place. The shift results in the one-numbered gene now being assigned to the first evolution of the new genome, the two-numbered gene to the second evolution, and so on and so forth. The outcome of each removal and subsequent shift in the sequences, results in the new genome, as shown, now consistent with what is required to sufficiently describe the layout design of that period evolution. Extension to the orientation sequences follows in parallel.

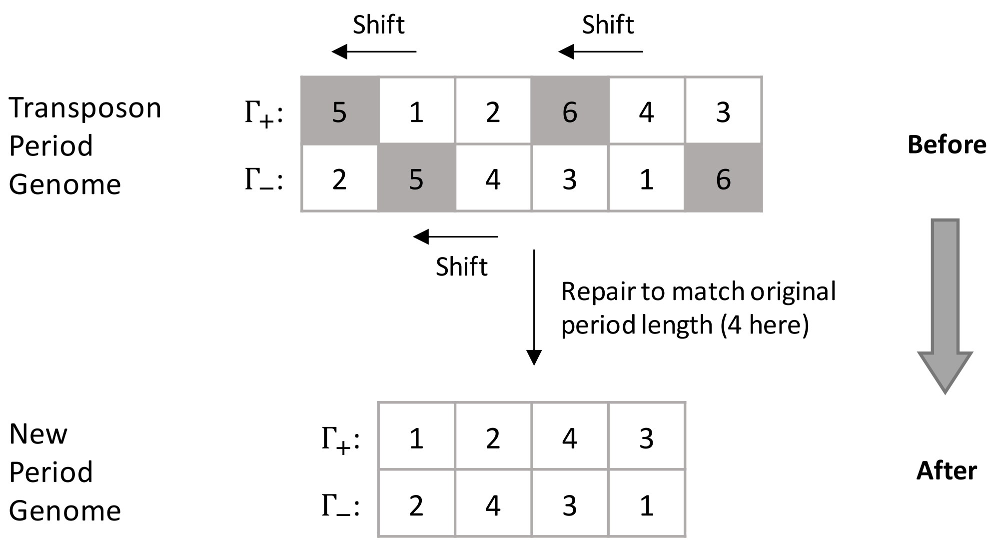


Figure 28 – Example of the additional genetic material genome repair process

Insufficient Genetic Material Genome Repair Process:

The insufficient genetic material repair process is invoked in scenarios where insufficient genetic material results as a by-product of a pasted transposon genome lacking, in length, what is required to sufficiently define the objects present for that period evolution. Repairing this genome, unlike that of the previous repair process, is far more difficult as additional genetic material must first be transferred from another source before the genome can become adequately populated to define all the objects present in the given period evolution. The question that then arose was, from what external source does one transfer this additional genetic material from? The answer to this question was to retrieve the additional information from the original genome. This was the most logical choice as the original genome would not only have the necessary genetic material to do so, but also the most up-to-date information for those additional objects. The repair process to rebuild the transposon genome of insufficient genetic material is thus as follows:

1) First, the required additional genetic material, genes, from the original genome are transferred into the new genome according to the position these genes appear in each of the original genome's position pair sequences

2) Then, the remaining genes of new genome are filled with the transposon genetic material, slotting the genes into the new genome according to the relative order they appear in the transposon genome

3) Finally, the same process is repeated for the orientation pair sequences

The resulting genome, once again, will have now become consistent with the original genome length and thus with what is necessary to sufficiently define the layout design of the current period evolution. Further, the strategic transfer of the additional data required from that of the original genome has the advantage of transferring the data that is most crucial to ensuring feasibility of the SP design. Since the constrained objects are those indexed last in each evolution, these are likely to be the missing indices when insufficient genetic material is experienced. The major benefit of then transferring and subsequently placing the additional genes in the new genome first is that it maintains that these, which are likely to be constrained objects, remain placed in a position they are most likely to appear in the sequences. Provided the original genome is one that is feasible, which is ensured given the construction of the GA solution procedure implemented, the subsequent genome would then too have a high probability of also being a feasible design.

An example of this genome repair process in practice is demonstrated in Figure 29, where the transposon genome was discovered to be of length four, two fewer than that of the original genome (i.e. length required to sufficiently define the period layout design). Starting with a new genome that is empty and of length six, the five and six genes, those that are absent in the transposon genome are transferred from the original genome into the new empty genome according to their original positions as demonstrated by the grayed genes. Next, the remaining genes are transferred from the transposon genome into the new genome slotting them into the open positions. As demonstrated in the positive sequence, positions three and five have already since been filled leaving positions one, two, four, and six open. The order of the genes in the transposon genome is two, four, one, and three. As such, gene two is slotted into position one of the new genome, gene four into position two, gene one into position four, and finally gene three into position six. The same process is repeated for the negative sequence as well before then also repeating both this and the one before for the orientation pair sequences using the same transferring and slotting orders. The result of this is then the new genomes of both position pairs and orientation pairs that are consistent in length with that which is required to sufficiently describe the layout design of the period evolution.

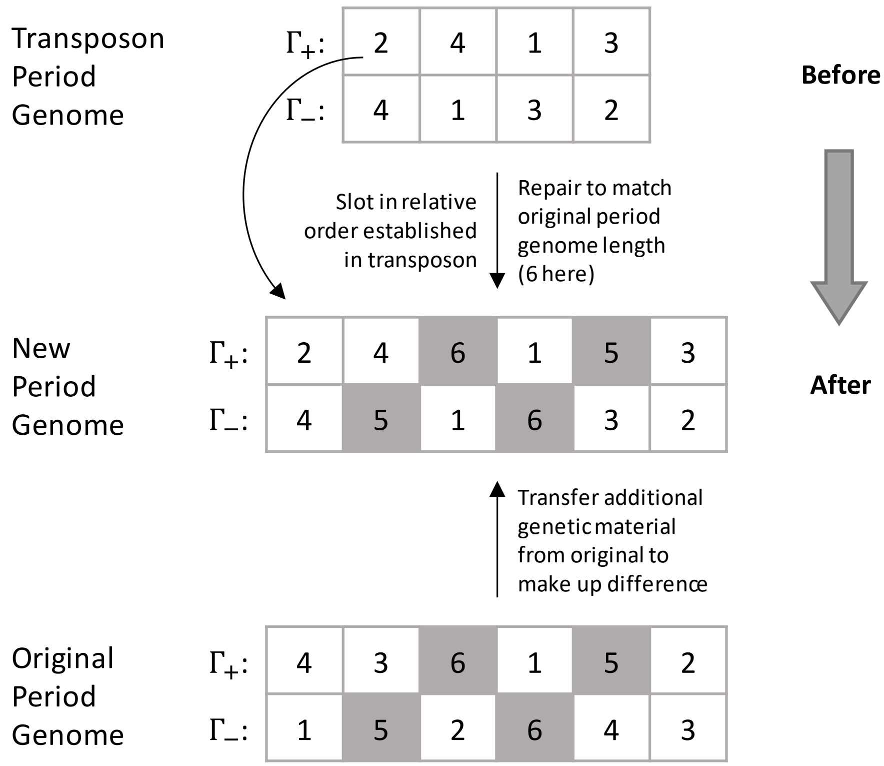


Figure 29 – Example of the insufficient genetic material genome repair process

4.2.5.4 Crossover Operator

After the probabilistic execution of the jumping gene operator, the developed reproduction process, as noted, then turns to more traditional genetic operators, the first being crossover, to induce further variation in the population. As mentioned back in the background chapter, the crossover's primary function is to vary the genetic composition of the population from one generation to the next by transferring different segments of genetic material from each parent to the offspring the process reproduces. Of the various crossover methods employed in the literature, the uniform crossover method is the most commonly applied method in the literature [129]. Thus, such a method was also deployed in this research. As discussed, and established in Table 6, Liu and Meller's modified uniform order-based crossover method, already designed exclusively for the QAP/U-SP data structure (order-based), was *adapted* to the DLP given its original application was to that of a SLP. To adapt their method to the DLP, extension of the method simply required performing the processes to be observed next for each period evolution of the DLP.

4.2.5.4.1 Sequence-Pair Crossover Process

The crossover process for the sequence-pair design variables is decomposed into two stages. In Stage One the appropriate genetic material is transferred from the parents to the offspring genomes. Stage Two then performs a relative order assignment procedure to assign the remaining unassigned genes in each offspring. These two stages are explained below and follow the same procedure employed by Liu and Meller.

Stage 1 – Transfer of Genetic Material from Parent

1) First, a binary bit string (BS), equal to the genome length, is generated for position pair selection

2) Then, position pair genes aligning with "1" bits in the BS are copied from PRT1 to OSP1, which denotes offspring one

3) Next the position pair genes aligning with "0" bits in the BS are copied from PRT2 to OSP2

This process is demonstrated in Figure 30, where a binary bit string, BS, for a six gene genome was first randomly generated. The result was a BS of [0,0,1,1,1,0]. With this BS, the position-pairs of the *c*, *d*, and *e* genes where transferred from PRT1 to OSP1 as they aligned with the "1" bits in the BS. Then, the positon-pairs of the *a*, *b*, and *f* genes were transferred from PRT2 to OSP2 as they aligned with the “0” bits in the BS. This process is no different from the conventional uniform crossover method employed in the literature. Following this though, Stage Two of the crossover process deviates from the conventional method in order to accommodate the order-based nature of the sequence-pair data structure. Before continuing though, one should understand that the (3,5) in the *a* gene of PRT1 establishes that object three lies in the first gene position of the positive sequence and likewise, object five lies in the first gene position of the negative sequence of PRT1. This terminology is important to understand before proceding.

	a	b	c	d	e	f
PRT1:	(3,5)	(6,3)	(2,2)	(4,6)	(1,4)	(5,1)
BS:	0	0	1	1	1	0
OSP1:			(2,2)	(4,6)	(1,4)	

PRT2:	(2,4)	(4,3)	(1,6)	(5,1)	(6,2)	(3,5)
BS:	0	0	1	1	1	0
OSP2:	(2,4)	(4,3)				(3,5)

Figure 30 – Stage One of the sequence-pair crossover process

Stage 2 – Relative Order Assignment of Remaining Unassigned Genes

1) First, the relative order of the unassigned gene positions (i.e. letters) of OSP1 are identified as they appear in that of PRT2 for both the Γ_+ and Γ_− sequences independently and per the sequential ordering of the object indicator numbers, or simply SP numbers, found in those genes (this will be made clearer with the example that follows)

2) Next, the missing SP numbers in the Γ_+ and Γ_- sequences of OSP1 are identified, then placed in sequential order

3) The remaining unassigned SP numbers in both sequences of OSP1 are assigned per the relative orders of the unassigned gene positions defined in step one

4) Steps 1-3 are repeated for OSP2 and PRT2 to populate OSP2 completely

The above process can be confusing to understand at first, so an example of the steps detailed above is provided in Figure 31 to make things clearer. The example provided is a continuation of the example described before in demonstrating the procedures of Stage One. First, one identifies that the *a*, *b*, and *f* gene positions are unassigned in OSP1, or in other words, empty in OSP1. Next, by examining the Γ_+ sequence of PRT2, and more specifically the unassigned gene positions just identified, it is found that the SP number two is found in the *a* gene, four in the *b* gene, and finally three in the *f* gene. Coupling these pairs together (i.e. *a* with two, *b* with four, *f* with three) and then sorting by the SP numbers from smallest to largest, the relative order of the unassigned gene positions then becomes *a*, *f*, and then *b* as demonstrated. Likewise, for the Γ_- sequence, the ordering becomes *b*, *a*, and then *f*. This completes step one of the process. Next, attention is turned back to the OSP1 genome, where it is identified that the Γ_+ sequence already contains SP numbers of one, two, and four, meaning that the SP numbers of three, five, and six are missing from OSP1. Similarly, the SP numbers of one, three, and five are missing from the Γ_- sequence. Finally, aligning this order of missing SP numbers to the ordered unassigned gene position found in step one, the SP number of three is assigned to the *a* gene, five to the *f* gene, and six to the *b* gene of the Γ_+ sequence. The same is done for the Γ_− sequence and once finished, the result is the completely populated OSP1 genome shown at the bottom of the figure. This process is then repeated to populate OSP2 using PRT1 in place of PRT2 to complete the crossover operator process. Note the definition of PRT1 and PRT2 doesn't necessarily need to be that of the parents selected during the selection operator procedure. If the JGO has been executed PRT1 and PRT2 would be then the resulting altered parents of this process.

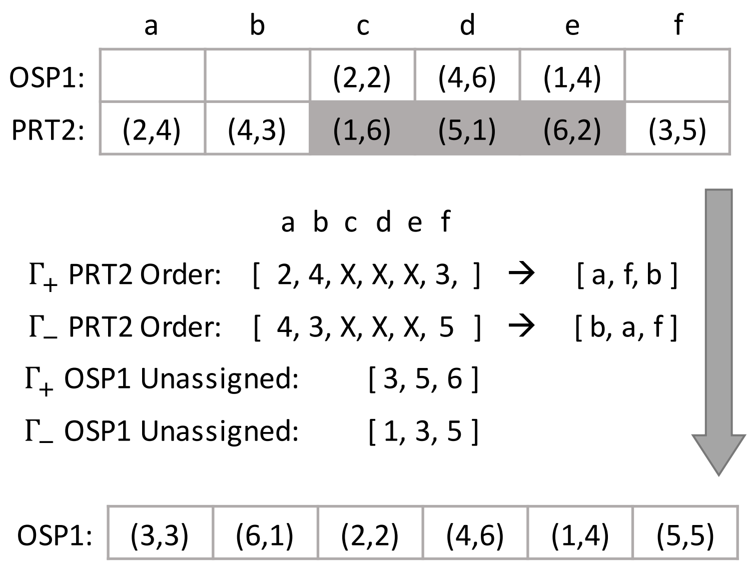


Figure 31 – Stage Two of the sequence-pair crossover process

4.2.5.4.2 Orientation Crossover Process

In Liu and Meller's formulation, rotation of the objects in the space, in addition to their positions, was not considered and therefore their method did not encompass a means of providing crossover amongst the orientation pair genes of the parents. As such, a method of performing this crossover needed to be constructed. Unlike the sequence-pair crossover process defined above, the process developed follows the traditional method of uniform crossover given the non-order-based nature of the orientation pairs. The process is as follows:

1) First, a binary bit string equal in length to the genome length is randomly generated

2) Next, if the bit is a “1,” PRT1’s orientation pair is transferred to OSP1 and likewise PRT2’s orientation pair to OSP2

3) Then to completely populate the offspring genomes, the orientation pairs for the “0” bit genes from PRT2 to OSP1 and similarly PRT1’s to OSP2 are transferred over

The result after both processes have been performed is then two reproduced offspring genomes. As mentioned before, this process as well as the one discussed before for the sequence-pair design variables are performed for each genome of the chromosome to provide crossover across the entire chromosome. In other words, these processes are performed for each layout design period evolution of the DLP. This concludes the discussion of the uniform crossover method deployed in this research to provide effective evolution of the population. Furthermore, this complete crossover process is performed probabilistically per a user-defined probability.

4.2.5.5 Mutation Operator

The last genetic operator to be implemented is mutation. Implementation of the mutation operator is important to maintain genetic diversity within the population from one generation to the next. This is often achieved through the random alteration of an individual, in some capacity, of the population. As discussed, and identified in Table 6, Liu and Meller's application of mutation to the QAP/U-SP SLP is once more leveraged. In this research's deployment of mutation, their pair-wise exchange mutation method is *adapted* to the DLP through the adoption of Ripon et. al.'s initial random period selection process. The deployed process is as follows:

1) First, a period evolution genome is randomly selected to mutate (Ripon et. al.), this effectively reduces the problem to an] SLP for which Liu and Meller's approach can then be applied

4.2.5.5.1 Sequence-Pair Process:

2) Then, select two position-pair genes in the SP randomly, without bias as to whether the pairs contain moveable or constrained objects, to exchange

3) The two position-pair genes are then exchanged

Selection without prejudice as to the nature of the objects encapsulated in the position-pairs is important to promote exploration outside of the more confined placement of constrained objects in the space that occurs as a by-product of the developed FSPPM. This further aids the algorithm in avoiding becoming trapped in local minimums. The sequence-pair mutation process is demonstrated for a two period DLP consisting of genomes of length six in Figure 32. As demonstrated, between the first step and second, the first period genome is chosen, followed by the random selection of two genes, position genes two and six. These selections are denoted by the grayed cells. Then the two genes are swapped resulting in the generation of a mutated offspring individual.

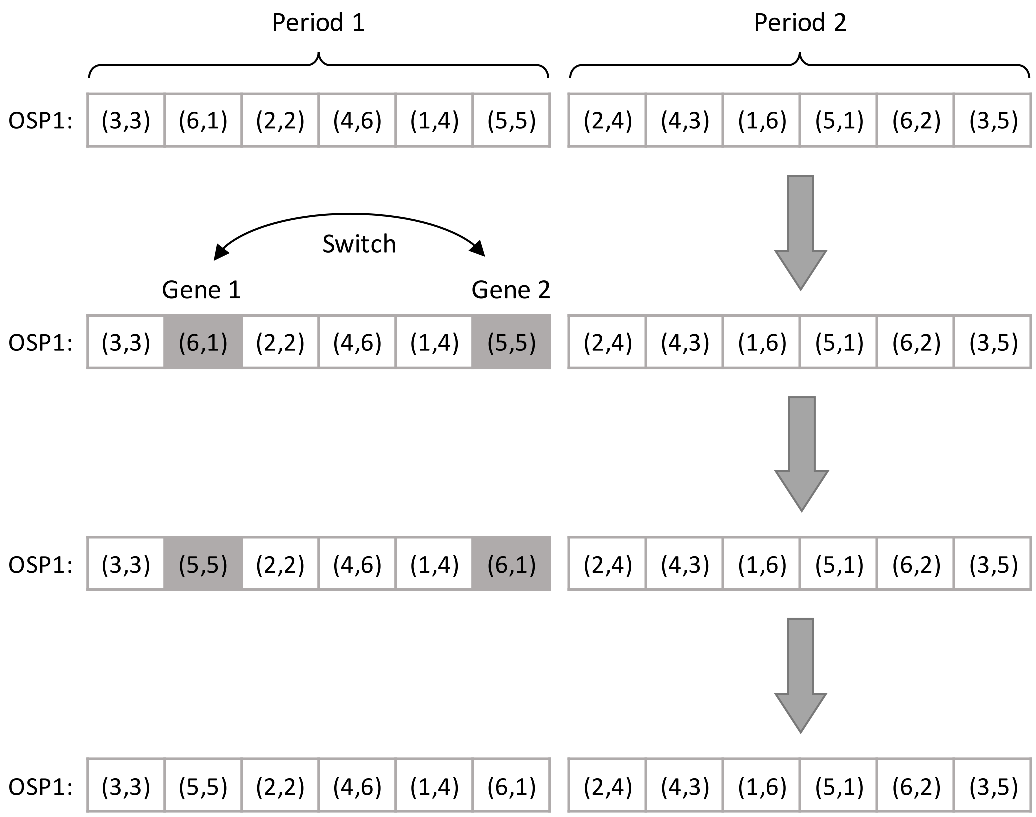


Figure 32 – Sequence-pair mutation process

Since neither Liu and Meller nor Ripon et. al. considered the rotation of the objects in the space in addition to their positions, a process for mutating the orientation-pairs of the randomly selected genome was defined and subsequently developed. The orientation mutation process developed and subsequently deployed is as follows:

4.2.5.5.2 <u>Orientation Process:</u>

2) With the genome since selected, a movable object in the space is randomly selected to rotate

3) Then, the object is rotated to one of the other three possible positions without bias as to which one

The selection of only movable objects to rotate was strategic. The reason for this is as follows, rotation of a constrained object is inherently useless as its inherent fixed nature prevents its rotation. Therefore, orientation mutation is only considered for movable objects in the space. As mentioned before, the above processes of sequence-pair and orientation mutation are applied probabilistically to each offspring individual generated by the upstream reproductive process operations, described in length before. With the completion of this operator, the reproduction process, or reproductive cycle, deployed in this research in the Stage One GA is complete. As established before, this reproductive cycle continues until the current generation's population has been completely populated with feasible individuals. Once this occurs, the genetic algorithm then proceeds to seeking further improvement through the implementation of FSA applied to the current generation's most fit individual.

4.2.5.6 FSA Improvement

With the fittest individual generated from the elitism process and reproductive cycle outlined before (i.e. the current generation) as the initial layout design configuration, FSA can then be applied to further improve this individual's fitness. The FSA algorithm deployed in this research leverages Chen and Chang's original FSA annealing schedule. Since, their annealing schedule was discussed in length in Chapter 2; one can refer to this earlier discussion for a complete understanding of the annealing schedule. The other key component to the FSA is the perturbation scheme for generating neighboring layout design configuration. The perturbation method developed is a synthesis of several approaches from the literature along with newly developed heuristics to account for unique problem setup of this research. Additionally, the FSPPM is leveraged within the algorithm to improve its effectiveness. The perturbation method developed and subsequently deployed to generate a neighbor design is as follows:

1) First, the current design is set as the origin design

2) Second, an evolution (i.e. period) of this design is randomly selected

3) Third, between the positive and negative sequence, a sequence of the selected evolutions sequence-pair is selected at random

4) Then according to a user-defined probability of reassignment, the positions of the fixed objects in the selected sequence are reassigned leveraging the placement distributions generated by the FSPPM (step mirrors that of Step 1 in Section 4.2.4.1)

5) Next, the remaining positions of the sequence are filled by slotting in the movable objects according to their pervious order, i.e. in the order they appear in the origin design

6) Then according to a user-defined probability of swapping, two movable objects are selected at random to have their positions in the sequence pair swapped

7) Next, according to a user-defined probability of rotation, a random movable object is selected for rotation

8) Now, of the two binary orientation bit variables defining this object's orientation, one is selected at random to be altered

9) Finally, the current binary value is identified and subsequently switched to its counter value

This procedure is performed each time a neighboring solution is to be generated for evaluation by the algorithm. Note instead of randomly placing the movable objects into the remaining position of the sequence-pair as was done before during the population initialization and outlined in Section 1.17.4.1, the movable objects are assigned according to their original order. This strategy helps to maintain that the design changes only marginally, i.e. remains a neighbor of the original design. Now when deploying the modified version of McKendall et al. [78] look-ahead / look-back strategy (LA/LB) in this research, this perturbation method is virtually identical except that the evolution, sequence, object selections, and occurrence of the various alterations are mandated by the overarching LA/LB procedure. In other words, the evolution is no longer selected at random, nor the sequence, nor whether the fixed objects are to be reassigned, nor whether swapping occurs or which are swapped if so, whether rotation occurs, and furthermore which movable object is rotated and by which binary bit variable. All these decisions are controlled no longer by chance, but rather by the LA/LB procedure which establishes each of these. Recall that this is the basis of the LA/LB. This strategy, outlined in detail in the background, looks to consider applying the adjustments made by the perturbation method outlined above to the other evolution period layout designs and not by random selection. This is meant as a means of propagating the adjustment throughout the other evolutions of the design to further improve it. In order to fully understand how the LA/LB strategy is deployed in the algorithm, the general sequence of operations is presented next.

4.2.5.6.1 Architecture of the Implemented FSA

The implemented FSA begins by first determining the initial cooling temperature for the annealing schedule. The initial cooling temperature is determined by deploying the standard perturbation method outlined before for a user-defined number of samples. Based on the number of uphill moves and the total uphill change for this sampling along with the user-defined probability of accepting an uphill move, the initial temperature is then established. With the initial temperature defined the annealing process then commences. Using the best design from this initial sampling, which includes the originally supplied fittest individual from the GA population, the annealing process perturbs this design to form a new neighbor for a user-defined number of samples. The new neighbor is formed by first deploying the standard perturbation method described before to the current best design. After being evaluated, if and only if it is accepted by the algorithm (either a downhill move or per the metropolis criteria), the modified LA/LB strategy is deployed. Using this perturbed design as the basis, the LA/LB method then applies the same perturbation to each of the other evolutions of the design accepting only those that produce a down-hill move, i.e. improvement. It is important to point out that the original formulation by McKendall et. al. also applied the metropolis algorithm here, thereby accepting inferior solutions by chance. Observations during implementation proved this approach to be disadvantageous. The chance of accepting an inferior solution often negated prior improvements, or worse yet, produced a design that was then infeasible. As such it was decided that the metropolis criteria not be implemented and moreover only designs that remained feasible could then be considered for selection.

Following the execution of the FSA, the operations of the current generation and further the second phase of the hybrid GA concludes.

4.2.6 Convergence Criteria

The final phase, phase three of the developed hybrid GA, is tasked with determining the algorithm's convergence. The method developed and deployed considers three criteria in order to do so. These criteria encapsulate time constraints, generation limits, and the continual improvement of the solution. When either one of these criteria are satisfied, a state of convergence is established. Once established the process of evolving the population from one generation to the next is terminated and the current best solution is then identified as the global best solution.

The first criteria deployed to establish convergence by the algorithm is an overall time constraint. This time constraint has two functions. The first is that it acts in preventing the algorithm from running for eternity; second it enables the user to dictate a specified duration of execution. The latter may be relevant when computational time limits are encountered and only a finite amount of time is available to solve each layout problem. The duration of the timing spans from the initiation of the GA's reproductive cycle, in other words just after the completion of the population initialization phase.

The second criterion is a limit on the number of generations the algorithm performs to evolve the population. A counter was implemented to track the number of generations executed by the algorithm. At the end of each generation this counter is compared to the user-defined, maximum number of generations, parameter. Once the counter exceeds this limit a state of convergence is met, and the reproductive cycle terminated.

While the first two criterions establish convergence based on limits in time or the number of generations executed, the third and final criterion implemented focuses on the algorithm's identification of the global optimum to establish convergence. Convergence in this sense, is established when no further solution improvement is possible by the algorithm. One way of establishing this is to enforce what is called a stall limit. This stall limit dynamically counts the number of sequential generations for where the best solution has not changed. If the algorithm is to find a better solution, the count is then restarted. Once the algorithm encounters the situation where it has not found improvement in the solution for a user-defined number of stall generations, a state of convergence is established by this criterion.

After each reproductive cycle, or generation, the three-criterion described above are assessed for convergence. If any one of them is met, the algorithm terminates the evolutionary process, thereby establishing the optimum solution for the problem as the current best solution in the population. This concludes the discussion on the third and final phase of the developed hybrid GA for Stage One. Now a summary and a few closing remarks on the Stage One hybrid GA are presented.

4.2.7 Summary of Stage One

As outlined in preceding discussions, in Stage One a hybrid genetic algorithm that incorporates a FSA technique to enhance its performance, to solve a QAP/U-SP formulation (i.e. designs defined by a sequence and orientation-pair) of the problem was developed. A novel FSPPM was also developed and leveraged throughout to aid the algorithm in discovering feasible designs more frequently, thereby reducing computational time and potentially improving solution quality. Additionally, novel methods were developed and implemented for many of the various genetic operators of its evolutionary process as well as for the perturbation method deployed by the FSA technique. This novelty and innovation was required to handle the unique nature of the mathematical representation, i.e. model, deployed in Stage One to characterize the physical layout design.

4.2.8 The Link Between Stage One and Stage Two

Before proceeding into the next section on the solution procedures developed for Stage Two, it is important to revisit the overarching goal of this first stage in the LIVE methodology. As was established in the preceding chapter on the formulation of the methodology, the overarching goal of this stage was to solve this slightly simplified model, in that of the QAP/U-SP representation of the layout, to then adequately and efficiently populate the initial populations of the GA implemented in Stage Two. This is achieved in practice by the implemented algorithm retaining all feasible designs generated during both the population initialization and evolutionary process phases. This collection includes that of the best solution and is passed to Stage Two to define the initial populations of its GA.

Though in the LIVE methodology, Stage One, as described, can be considered nothing more than an advanced population initialization method for Stage Two, the implementation in this work is robust enough that if the designer so chooses, the result of

Stage One can become the final solution while Stage Two becomes then inactive. This decision may be the result of limited computational resources (Stage One takes a lot less time to solve than Stage Two), the need for only an initial more conceptual layout design (e.g. one that does not provide a continuous layout design solution), or the desire to more rapidly visualize the design space topography whereby a specific region may then be identified for further exploration using Stage Two.

Regardless of the reason, this decision requires a few user-defined parameters to be defined more appropriately for this goal. For starters, it is recommended that the convergence related parameters be adjusted accordingly. This means extending the time limit and increasing the generation limit as to ensure the algorithm completely converges on the best solution. Remember the goal of Stage One by default is not necessarily to find this best solution, but rather to provide Stage Two with a good sampling of designs to initialize its populations with. As such, Stage One by default is likely to sacrifice some optimality for speed. This not only comes in the form of limiting the extent to which the problem is solved via the convergence parameters, but also through that of the population size. A smaller population is likely to be used by default to expedite the Stage One process; however, it is recommended that this population size be increased to enable better solution performance albeit at the expense of a longer solution time. A study on the extent to which Stage One should be solved and how to establish the population size will be presented when the experiments performed are presented later. With the link between Stage One and Stage Two reestablished, the focus now turns towards the solution procedures of Stage Two and how these procedures leverage the results of Stage One.

4.3 Step 2: Solution Procedures of Stage Two

This section begins with a brief overview of the mathematical model deployed in Stage Two to characterize the layout design along with the design variables that define it. Then, an expansive discussion on how improved layout designs are sought/discovered through the manipulation of the design variables is presented. This more broadly encapsulates the GA solution procedure implemented to solve the layout problem of stage two. This discussion consists of the following: how the GA's population is initialized by leveraging the collection of designs produced by Stage One, how this population is then evolved through the deployment of genetic operators tailored to the mathematical model leveraged, and how convergence of the algorithm is established. With the model chosen to mathematically define the layout being at the core of how all other elements of Stage Two are constructed, the mathematical model deployed in Stage Two is first presented.

4.3.1 Mathematical Representation of the Layout

As was established in the preceding chapter, a MINLP model, resembling that of Barbosa-Póvoa et al.'s (2001) non-linearized MIP formulation of the static layout problem variant, was implemented in Stage Two to geometrically model the layout. This model differs from the model of Stage One in one critical way; the continuity property. Unlike that of the Stage One QAP/U-SP model, where placement is discrete and stacking-based, the model of Stage Two is continuous. This is because Stage Two operates directly on the coordinate centroid positions of the object rather than on that of sequence-pairs, as was done in Stage One.

4.3.1.1 Design Variables of Stage Two

In Stage Two, placement of the objects is instead defined as the synthesis of the object coordinate centroid positions and their orientations. Just as was the case in Stage One, the orientation design variables remain as being defined by an orientation-pair, or pair of binary bit sequences. Contrary to Stage One though, the position design variables, before an order-based sequence-pair, now become the non-order-based coordinate centroid positions, defined by an x and y coordinate, of the objects. These are further normalized by their ranges (defined according to the OML of the layout). This was done as to avoid issues during the reproductive process of the GA where positions were being exchanged and modified by the evolutionary genetic operators.

This synthesis of continuous and binary integer variables is what makes the model MIP, or mixed-integer, in nature. Provided this representation, manipulation of the design is controlled in Stage Two by directly altering the x, y coordinate positions of the objects along with the object orientations. While the sequence-pairs of Stage One were discrete in nature, the coordinate centroid positions are continuous. By directly operating on these centroid positions to place the objects in the space, a continuous layout design is then able to be considered. As was noted before, this is critical to being capable of adequately characterizing the layout and therefore why such a MIP model was deployed here. Now that the design variables of Stage Two have since been established, the algorithm implemented to manipulate these variables in search of the optimal layout design is now discussed.

4.3.2 Architecture of the Implemented Tri-Population Genetic Algorithm

To perform this search for the optimal layout design to the provided problem, a tri-population genetic algorithm, mirroring the one originally developed by Pourvaziri and Naderi, was deployed in Stage Two. The sections that follow outline the solution procedures deployed by the tri-population GA to achieve this optimal layout design discovery. How the implemented tri-population GA initializes the population, subsequently evolves this population, and finally how it identifies when the most optimal design has been discovered is detailed. Before diving into each of these, the general process flow of the implemented tri-population GA is first presented.

As is the case with any GA, the first phase in the implemented tri-population GA is to initialize the population or put alternatively, populate the initial population that will then be subsequently evolved. In this implementation the same is true, though instead of initializing a single population of designs, three distinct populations are constructed. These three populations are formed from the collection of designs generated by Stage One. This three-population structure is what gives the tri-population GA its name. Once these three populations have been formed, each is independently evolved through an evolutionary process. The evolutionary processes performed on each of these populations are fundamentally identical. These populations are evolved for a user-defined period of isolation, unless solution convergence occurs first. The isolation period can represent the time or number of generations each population is evolved for before being then merged to form a single population. The convergence criteria deployed inherently account for this isolation period consideration along with the solution convergence. Therefore, the convergence block presented in Figure 33 encapsulates both these. This process of simultaneously evolving the three distinct populations until convergence comprises the second phase of the algorithm. After the isolation period, the three populations are merged into a single population through a migration process. This migration process constitutes phase three of the algorithm. This single merged population is then evolved through the same evolutionary process deployed in phase two to evolve the three initial populations. Similarly, the same convergence criteria in phase three are leveraged in phase five to identify convergence for the merged population. This tri-population procedure mirrors Pourvaziri and Naderi's implementation in the literature.

Now that the developed tri-population GA's procedures have been outlined, the unique concepts of the sequence will be presented in detail. Since the evolutionary processes of phase two are identical to that of phase four and the convergence criteria of phase two are identical to that of phase five, this leaves only three unique concepts to discuss. These include the population initialization procedure deployed to construct the three initial populations, the evolutionary process leveraged to evolve the four populations, and finally the criteria used to establish when the algorithm has converged.

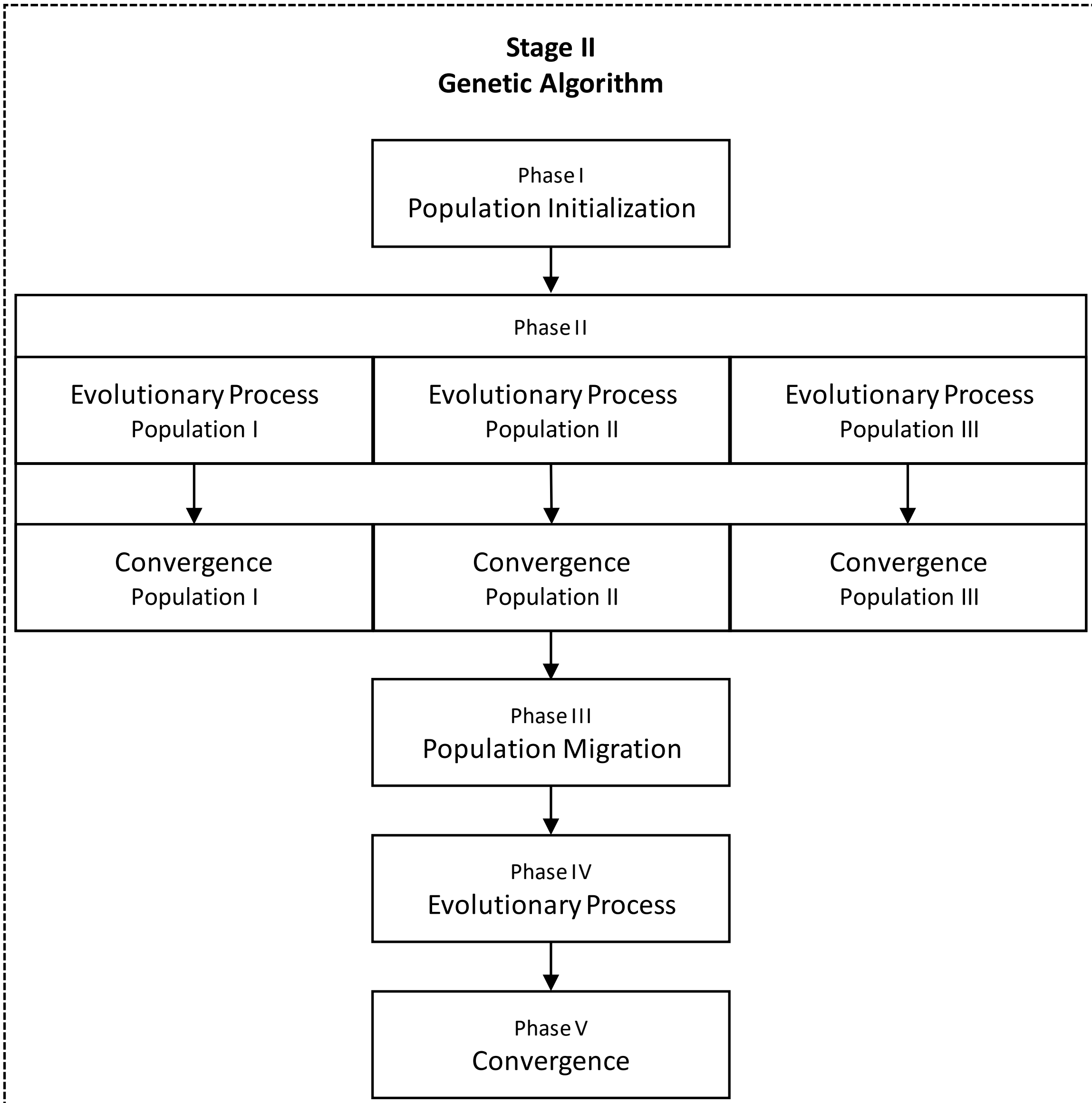


Figure 33 - Architecture of the implemented Stage Two genetic algorithm

4.3.3 Initializing the Populations

The process of initializing the populations in Stage Two is significantly different from that of Stage One. Beyond the obvious in that three initial populations are to be formed rather than just one, the method implemented to initialize the populations also does not require any designs to be generated. This is because such designs are already available as a by-product of the collection of designs generated by Stage One. It is from this collection, or design pool, that designs are selected for assignment to each of the three populations. It was decided, like that of Pourvaziri and Naderi's formulation, that these three distinct populations would be initialized as follows. The first population is composed of the best designs from the collection, while the second population is the antithesis of this in that it is composed of nothing but the worst designs of the collection. The third and final population is constructed by randomly selecting designs from the provided collection. Provided that the collection is insufficient in size to completely populate any one of the populations, duplicate designs from the pool are selected at random to make up the difference. Furthermore, each of the three populations can be sized differently. The population sizes of each are defined by three independent user-defined population size parameters. This enables the user to have control over the size of each and furthermore enables a study to later be performed to determine the best combination of sizes to deploy for different problem types.

It was believed that this best, anti-best, and random structuring of the three populations would provide a healthy balance of elitism and diversity in the algorithm and that this balance would then propagate downstream into the merged population following the isolation period where the three populations are evolved independent of one another. Having now established how the collection of designs generated by Stage One are leveraged to establish the three initial populations of Stage Two, a discussion on how the implemented evolutionary process evolves these populations as well as the merged population is now discussed.

4.3.4 The Evolutionary Process of Stage Two

The evolutionary process deployed, in phases two and five of the algorithm, to evolve the populations of Stage Two leverages the same genetic operators that were deployed in Stage One to do so. Given that the designs of Stage Two are now represented by the coordinate centroid positions of the objects, which are continuous and non-order-based, and an orientation-pair, the genetic operators implemented to vary the designs in Stage Two needed to be designed to accommodate this new chromosome representation, or design variable composition.

The evolutionary process of the developed Stage Two GA is structured identically to that of the Stage One process, with the exception being that Stage Two does not deploy FSA to enhance the most elite design of the population. Provided that the sequence and composition of operations is the same in Stage Two as it is in Stage One, only a brief overview of the sequence will be presented here. If one desires, an in-depth discussion of the sequence can be found in Section 4.2.5 on page 152. The sequence of operations comprising the evolutionary process begins with first the execution of an elitism selection operator. Following its execution, the reproductive process begins. This process is comprised of the following operators in order of their execution in the algorithm: reproduction selection, jumping gene, crossover, and finally a mutation operator. Once execution of these has occurred, the new offspring design is evaluated for its performance and feasibility. Just like before, designs that satisfy the feasibility property, i.e. constraints of the problem, are added to the current generation's population. This reproductive process continues until the generation is fully populated of feasible designs. This complete sequence is visually depicted in the right image of Figure 23 provided before. The only difference being that the FSA block in the figure would be removed.

Table 8 – Stage Two implemented genetic operator strategies

		Model
		MINLP
Selection Operators	Selection:	Liu and Meller's / Ulutas and Islier's roulette wheel selection [115,167]
	Elitism:	Traditional *k* best
Variation Operators	Crossover:	Mazinani et al.'s continuous uniform operator [118]
	Mutation:	Mazinani et al.'s tri-mutation operator approach [118]
	Jumping Gene:	Ripon et al.'s cut and paste and copy paste operations ***adapted*** [143]

4.3.4.1 Elitism Operator

As was the case before in Stage One, the concept of elitism was also implemented in the Stage Two algorithm to improve its performance. As established in the preceding chapter and re-presented above in Table 8, to employ the elitism concept, a traditional *k*, best transfer method was deployed to ensure that the best designs of the preceding generation survived. The method selects the most elite *k*, unique designs from the previous generation population and then assigns them, unaltered, to the current generation's population.

4.3.4.2 Selection Operator

Following the execution of the elitism operator outlined above, the evolutionary process enters the reproduction process where the remaining $N_{pop} - k$ designs are then assigned to the population. These designs are formed by first selecting designs from the previous population to act as parents and then combining/modifying these parents to form offspring designs for consideration. Selection of these parents is the function of the selection operator. As outlined in Table 8, a roulette wheel proportionate selection method was deployed to systematically perform this selection. This is the same selection method deployed in the evolutionary process of Stage One and outlined in Section 4.2.5.2 on page 160. To avoid redundancy, one may refer to this cited section for a complete understanding of how the selection operator is constructed and how it selects parents for reproduction.

4.3.4.3 Jumping Gene Operator

Now that two parents have been selected for reproduction, the process continues with the deployment of a jumping gene operator (JGO). Same as before, the implemented JGO is executed with a user-defined probability. In other words, it may not always be applied to the parents to promote evolution. As highlighted in Stage One, the JGO's ability to promote genetic diversity and furthermore evolution of the population, is the core reason behind its deployment once more here in Stage Two. The JGO deployed in Stage Two employs the same general process and assumptions as in Stage One. Furthermore, the transposition processes deployed by it are also identical to those in Stage One. To avoid redundancy, neither the overall process, assumptions, nor transposition processes will be outlined here. Instead, how the JGO of Stage Two differs from that of the one deployed in Stage One will be examined and the necessary differing elements elaborated on. If one desires a recap of the overall process, the assumptions, or the transposition processes they may refer to Section 4.2.5.3 on page 161.

The major reason why the two do differ lies in the nature of the design variables of the stages. In Stage One, the position variables were that of an order-based sequence-pair. In Stage Two though, these are replaced by the continuous non-order-based coordinate centroid positions of the objects. As a result, the letters presented in the diagrams before now represent coordinate centroid positions and orientation pairs that define the layout design of that given evolution (cell/gene). This is important to understand as it is for this reason that the genome repair processes of Stage Two differ from those in Stage One. Additionally, unlike in Stage One where the complete genome was operated on, only the portions of the genomes that are representative of the movable objects in the space are operated on in the implemented JGO of Stage Two. As a reminder, the genome repair processes are unique to this research and a required measure as a result of the problem formulation considered in this research.

4.3.4.3.1 Genome Repair Process

As a by-product of the unique formulation of this dissertation, the genomes (i.e. periods) making up a layout design can have differing lengths to account for the addition or potential subtraction of objects from the environment. Managerially speaking, this is representative of situations where a new or old out-dated asset is added or removed from the environment respectively. With the transposition processes altering the compositions of the parent designs by shuffling the genomes around, situations of inconsistent period genome length can arise. In this case, where the transposon genome length is shorter than the original genome, the scenario of insufficient genetic material arises. On the contrary, the opposite can too occur, where the transposon genome length exceeds that of the original genome. This case characterizes the scenario of additional genetic material.

Again, these scenarios arise as a unique by-product of this research which considers the assessment of evolving business models in parallel with an evolving layout design. The novel genome repair process developed consists of two sub-processes, one for each repair scenario mentioned before. It should be once more observed that only the pasted genomes of the copy and paste operations and all shifted and pasted genomes of the cut and paste operations need be inspected for repair. Amongst these, only those resulting in inconsistent genome length need be repaired by the appropriate repair process.

Additional Genetic Material Genome Repair Process:

The additional genetic material repair process is invoked in scenarios where excessive genetic material results as a by-product of a pasted transposon genome exceeding, in length, what is required to sufficiently define the objects present for that period evolution. Repairing this genome to be consistent in length with the original genome is a relatively straightforward process as all the necessary genetic material is already present, it merely needs to be systematically trimmed of unnecessary genes to match with what is required for that evolution. When repair of such a genome is required, the extra genetic material is simply trimmed from the end of the genome.

The rationale for trimming from the end rather than the start of the genome is as follows. The excess genes are likely to be the result of objects added to the environment. Provided that the genomes consist of only the movable objects and moreover that added objects are, in this implementation, appended to the end of the sequences, it is logical to trim these first from the sequences. By doing so the added objects are effectively removed from the space, leaving just the original movable objects, where original means those native to that period. Furthermore, removal of these and then the subsequent direct transfer without alteration of the remaining genes is advantageous as it has a then high probability of yielding a feasible design as the transposon period genome is inherently feasible.

An example of this genome repair process is demonstrated in Figure 34, where the transposon genome was discovered to be of length six, two more than that of the original genome (i.e. length required to sufficiently define the period layout design). For reference, the decimals in the cells are the normalized coordinate positions of the object centroid positions and are similarly that of the orientation pairs when repairing the orientation sequences. As demonstrated the last two genes, grayed, are simply trimmed to create the new period genome. As briefly mentioned, the orientation variable sequences are also trimmed in this same manner.

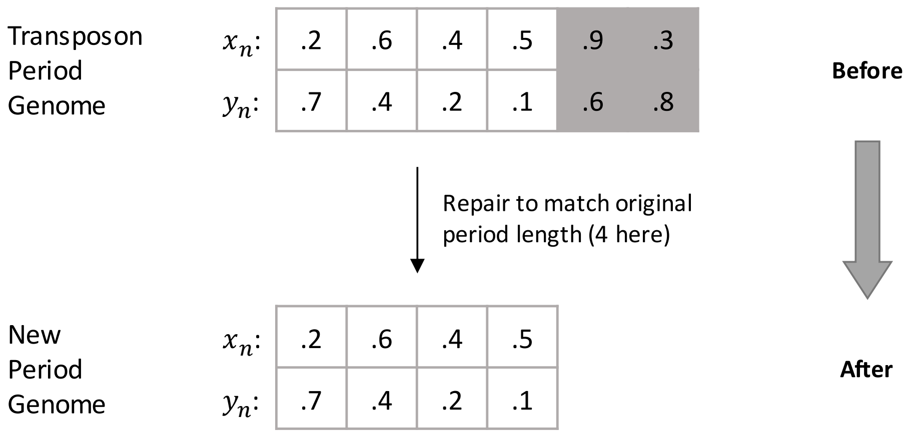


Figure 34 – Example of the additional genetic material genome repair process

Insufficient Genetic Material Genome Repair Process:

The insufficient genetic material repair process on the other hand, is invoked in scenarios where insufficient genetic material results as a by-product of a pasted transposon genome lacking, in length, what is required to sufficiently define the objects present for that period evolution. Since supplemental genetic material must first be transferred from another source before the genome can be completely populated, this repair process is a significantly more involved. In the developed method, the source of the additional information is that of the original genome just like before in Stage One. This was a sensible choice as it has both the necessary genetic material but also has the most up-to-date information for these additional objects. The repair process to rebuild the transposon genome of insufficient genetic material is thus as follows:

1) First, the additional genetic material, genes, from the end of the original genome are transferred into the same end positions of the new genome

2) Then, the remaining genes of the new genome are filled according to the following equation:

$$g' = g^{O} + c(g^{T} - g^{O}) \tag{6}$$

where g represents the gene variable (x, y), ‘ superscript represents the new period genome gene value, O superscript represents the original period genome value, T represents the transposon period genome value, and finally c represents a coefficient of adjustment whose range is from zero to one. This adjustment coefficient determines the degree to which the original genome value is adjusted in the direction of the transposon value. It is recommended that this value be relatively low as to not excessively alter the design from that of the original genome, which is inherently feasible. Overly adjusting in the direction of the transposon will result in a higher probability of the yielded design then becoming infeasible. Maintaining a relatively low c value can be thought of as providing a localized adjustment of the original genome in the direction of the transposon. A value of less than 0.25 is highly recommended especially for problems that are highly constrained. For such problems it is further recommended that this value be set much lower than that of 0.25. A value of 0.1 would be a good starting value. A visual example of this repair process, applied to the position variables and using a c value of 0.2, is provided in Figure 35. Note, step two and this equation only applies to the coordinate position variables which are continuous. The orientation variables, being binary, are treated differently, though step one remains the same. To repair the orientation variables, the remaining unfilled genes are filled by directly transferring the genes, i.e. orientation-pair binary bits, from the transposon to the new period genome.

The resulting genome, once again, will have now become consistent with the original genome length and thus with what is necessary to adequately define the layout design of the current period evolution.

An example of this genome repair process in practice is demonstrated in Figure 35, where the transposon genome was discovered to be of length four, two fewer than that of the original genome (i.e. length required to sufficiently define the period layout design). Starting with a new genome that is empty and of length six, the five and six genes, those that are absent in the transposon genome, are transferred from the original genome into the new empty genome as demonstrated by the grayed genes. Next, the remaining genes are transferred from the transposon genome into the new genome. The result of this is then the new genomes of both coordinate centroid positions and orientation-pairs that are consistent in length with that which is required to sufficiently describe the layout design of the period evolution.

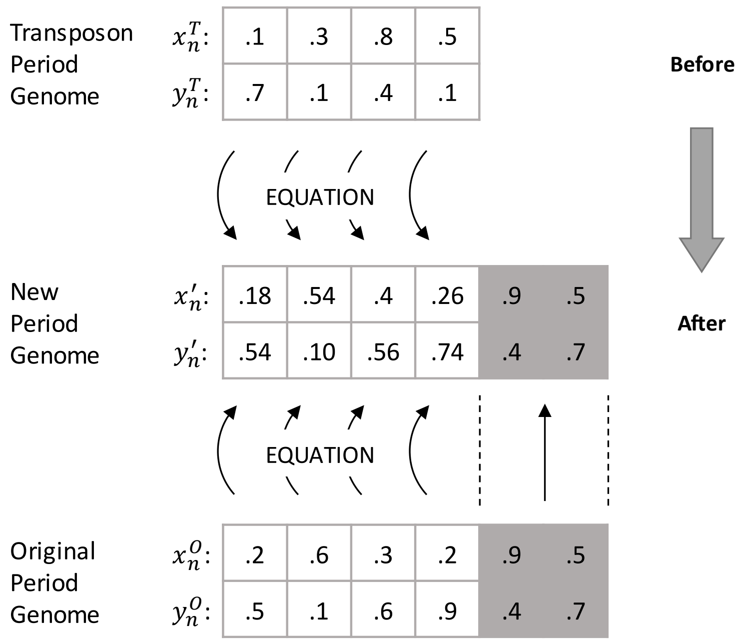


Figure 35 – Example of the insufficient genetic material genome repair process

4.3.4.4 Crossover Operator

After the probabilistic execution of the jumping gene operator, a crossover operator is leveraged to vary the genetic composition of the modified parents produced by the JGO or the selection operator if the JGO was by chance not executed. The deployed cross-over operator is a direct adoption of Mazinani et. al.'s continuous uniform method already applied to the DLP. Note, like before, only the portions of the chromosome representative of the movable objects in the space, are operated on in this implementation. This is to ensure that the constrained objects in the space remain unaltered and therefore in their required positions. This helps to ensure feasibility, relative to such objects, is maintained by the algorithm. This nuance is unique to this dissertation's implementation and is a result of the unique problem formulation considered. The deployed process for generating two offspring from that of the parent chromosomes is as follows:

1) First, a random uniformly distributed bit string equal in length to the period genome is generated as notionally demonstrated below for a genome consisting of four genes each

2) Then, the coordinate centroid positions for the offspring are generated according to the following equations:

$$\begin{aligned} g_i^{O1} &= \lambda_i g_i^{P1} + (1 - \lambda_i) g_i^{P2} \\ g_i^{O2} &= \lambda_i g_i^{P2} + (1 - \lambda_i) g_i^{P1} \end{aligned} \tag{7}$$

where *i* represents the gene of the bit string or genome, g_i^{O1} and g_i^{O2} represent the gene value (x or y coordinate position) of object *i*'s coordinate centroid position for offspring one and two respectively, λ_i the bit string of step one, and g_i^{P1} and g_i^{P2} the gene values of parent one and two respectively. The process continues with the orientations as follows:

3) First, a random binary bit string equal in length to the genome is generated

4) Next the orientation gene values from parent one are transferred to the first offspring from the first parent for genes aligning with bits in the binary string having a value of one

5) Then similarly, the orientation gene values from parent two are transferred to the first offspring from the second parent for genes aligning with bits in the binary string having a value of zero

6) Steps 2 and 3 are repeated for the second offspring except that now genes from parent one are transferred for zero bits and from parent two for one bits (i.e. simply the inversion of before)

The above process is performed for each genome of the chromosome or put alternatively, each period of the layout design. The result of this is then two reproduced offspring genomes. This concludes the discussion of the uniform crossover method deployed to effectively evolve the populations of Stage Two.

4.3.4.5 Mutation Operator

The last genetic operator deployed is mutation. As mentioned several times before, the mutation operator facilitates genetic diversity in the populations. The mutation operator deployed in Stage Two to provide this diversity, adopts Mazinani et. al.'s tri-mutation scheme with alteration to method one of their scheme. This scheme employs three different mutation methods to alter the provided offspring design. Note, in this deployment, mutation is applied to each offspring with a very small probability. Also, these three methods operate only on the coordinate centroid position variables of the design variables and furthermore only for those objects that are movable, or free, in the space. A separate method is deployed to provide orientation mutation in the design.

The first mutation method is a continuous localized adjustment of a single gene, the second a pair/tri-wise exchange of selected genes in a period, while the third is a pair-wise exchange approach applied to all genes of a selected period. The process starts by first randomly selecting which of these three methods will be deployed to alter the design. The mutation method selected is then performed producing a slightly mutated offspring design to then be considered for assignment to the current generation population. The three methods leveraged to mutate the offspring design are now outlined, starting with the continuous localized adjustment method.

4.3.4.5.1 Method I – Continuous Localized Adjustment Method:

The process deployed for mutating the offspring design in method one is as follows:

1) First a gene (g_i) in the chromosome is randomly selected for mutation

2) Then a standard normal number, z, is randomly generated

3) Finally, the gene selected in Step 1 is redefined according to the following function:

$$g_i' = \begin{cases} g_i + \left(g_i^U - g_i\right)\tanh(kz), & \tanh(kz) \geq 0 \\ g_i + \left(g_i - g_i^L\right)\tanh(kz), & \tanh(kz) < 0 \end{cases} \tag{8}$$

where g_i' is the redefined gene value, g_i the original gene value, g_i^U and g_i^L the gene's upper and lower bounds, k a user-defined coefficient that controls the degree of closeness the redefined value is to the original, and z the randomly generated standard normal number from Step 2. This function is slightly different from Mazinani et. al.'s original formula. Mazinani et. al. applied a similar formula but applied to the flexible bay layout design problem. Therefore, to account for this difference in problem formulations, the bay ranges of the function were replaced with the continuous variable bounds. This substitution provided the same effect of promoting mutation in the neighbourhood of the original gene while preventing extreme changes from occurring. Steps two and three are effectively executed twice, once for the x variable and once for the y variable of the randomly selected gene g_i. The redefined gene values overwrite the original gene values to form the newly mutated offspring design.

4.3.4.5.2 Method II – Pair/Tri-wise Exchange Method:

The process deployed for the second implemented method leverages the exchange of genes in the chromosome to mutate the offspring design. The process deployed is as follows:

1) A genome, or period, of the chromosome is randomly selected

2) Next, three genes in the selected genome are selected at random

3) Then the pair-wise and tri-wise exchange of the three genes selected in Step 2 is performed

The exchange in Step 3 yielding the greatest improvement in the design's performance is then accepted as the newly mutated offspring design.

4.3.4.5.3 Method III – Pair-wise Exchange Method:

The process deployed for the third implemented method is very similar to that of the second. It differs in that only pair-wise exchanges are considered. Additionally, instead of selecting only three genes of a randomly selected genome, or period, all genes are considered for exchange. The exact process is as follows:

1) A genome, or period, of the chromosome is randomly selected

2) Pair-wise exchange of this periods genes is then performed

Just like in method two, the exchange in Step 2 yielding the greatest improvement in the design's performance is then accepted as the newly mutated offspring design.

4.3.4.5.4 Orientation Mutation Method:

The mutation methods outlined prior are applied only to that of the coordinate centroid position variables of the design. To provide diversity on an orientation-basis the following procedure was developed and subsequently deployed:

1) A genome, or period, of the chromosome is randomly selected

2) Next a gene in this selected genome is random selected

3) Then, one of the two binary bit variables defining the gene is selected at random to be switched from its current value to its alternative (e.g. if its value is 0 it would be switched to 1)

Once the offspring has been mutated by one of the randomly selected methods from earlier and then this orientation method, the resulting mutated offspring is passed on for potential assignment, by the algorithm, to the current population baring it is found to be a feasible design (i.e. one that abides by all constraints of the problem formulation). This reproductive cycle off selecting parents, evolving them to produce offspring, and finally mutating these offspring continues until the current generation's population has been completely populated with feasible individuals, or designs. Once achieved the evolutionary process begins again for the next generation and continues this cycle until convergence is achieved. How the implemented algorithm of Stage Two registers convergence is now discussed.

4.3.5 Convergence Criteria

The convergence criteria deployed to establish the convergence of the Stage Two populations, the three of phase two and the merged one of phase five, leverages the same criterion as was deployed in Stage One to establish convergence. As a review, these include a time constraint, maximum generation limit, and a continual solution improvement check in the form of a generation stall measure. To avoid redundancy in this document, one may refer to Section 4.2.6 on page 190 for a complete discussion of these criterion. A discussion on how these constraints, limits, and stall measures are leveraged in Stage Two, will however be provided.

In Stage Two, and more specifically that of phase two where the three initial populations are evolved independently, the time constraint and maximum generation limit are leveraged to enforce the isolation period discussed earlier. If it is desired that the isolation period last for a specified duration of time, the time constraint criteria can be leveraged to ensure that once this time spent in isolation is met, the populations would then be merged. It is recommended that if this be the case, then one should define the maximum generations appropriately to allow for the time constraint to be met without first reaching the generational limit resulting in the populations being merged prematurely. Similarly, if it is desired that the isolation period last for a prescribed number of generations before the merger, then the generation limit can be prescribed accordingly, taking care, like before, to prescribe the time constraint appropriately. This logic only applied for the populations of phase two. As for the merged population it is recommended that the convergence criteria be defined appropriately to ensure the algorithm can completely converge on the best design for the provided layout problem.

4.3.6 Summary of Stage Two

As discussed earlier, Stage Two leverages a tri-population genetic algorithm to solve the MINLP formulation of the problem (i.e. designs defined by continuous centroid positions and binary orientation-pairs). Solution to such a formulation has since been established as being imperative to being able to assess a continuous layout and further adequately characterize real-life viable designs. The tri-population structure was implemented to provide improved and more robust solution performance. These initial populations are initialized by leveraging the collection of feasible designs generated throughout the solution procedures of Stage One. Novel methods were developed for several of the genetic operators deployed in the evolutionary process of the algorithm to handle the unique nature of the problem formulation of this research.

Before proceeding, it is important to revisit the overarching goal of this second stage in the LIVE methodology. As was established in the preceding chapter on the formulation of the methodology, the goal of Stage Two is to solve the detailed formulation of the continuous dynamic layout problem. Due to the complexity of such a formulation, the outlined algorithm and methods of Stage Two were developed to achieve solution to the problem most efficiently. With the developed solution procedures of Stage One and Two now thoroughly detailed, the next section elaborates on how the designs generated throughout these stages are evaluated for their performance and feasibility.

4.4 Step 3: Evaluating the Performance of a Layout Design

Preceding sections have focused on how the layout problems are initialized and subsequently solved in the LIVE methodology. A major component of this dissertation, and one that is critical to how the best solution is defined for the problem, has yet to be discussed though. This component relates to how the designs are evaluated for performance and feasibility. The focus of this section is thus on how the feasibility and performance of each of the designs generated and considered throughout Stages One and Two are established in this dissertation. This section is decomposed into two sub-sections. The first reviews the performance model developed to evaluate a design's quality, or fitness, while the second reviews the constraint model developed to determine the feasibility of a design.

The performance model developed applies across both stages and the constraint model largely does as well, though there are some key differences. These differences relate to the inherent nature of the Stage One formulation and more specifically that of the sequence-pair model deployed in it to represent the layout. Further, despite the design variables defining the layouts of Stage One differing from that of Stage Two, the performance and constraint models developed are mathematically identical. This is because though Stage One operates on sequence-pairs to establish the position of the objects in the space, the placement algorithm, outlined in Section 4.2.1.1 on page 128, can be leveraged to map these sequence-pairs to object *x, y* coordinate positions in the space. Once mapped, the designs of Stage One and Two are identically represented by a combination of *x, y* coordinate positions and orientation-pairs, which are then used in the models to establish design performance and feasibility. How the performance of a layout design is established in this dissertation is now presented.

4.4.1 Performance Model

A major emphasis of this dissertation was to provide a medium in which more informed, collaborative, and effective design decisions could be made. The performance model developed in this dissertation, to evaluate designs, was instrumental in realizing this goal. In an era were "data is power" or put alternatively, "data is knowledge and knowledge is power," access to information is a necessity [49]. The developed performance model provides this access in the layout design process thereby enabling more informed and effective data-based decisions to be made. This access is ever more important when tackling the design of layouts subject to evolving and uncertainty conditions.

In the developed model, key performance metrics are made transparent to the designer to help aid in the decision-making process. Additionally, rather than aggregating these performance metrics into a utility function, with no physical meaning, to define a design's overall performance (i.e. objective function in the optimization algorithms of before), a different approach was taken in this dissertation. Instead a cash-based model was deployed to define the performance of the layout design and more generally the system. Leveraging a cash-based model also has an added benefit of promoting collaboration, another pillar of this dissertation's overarching goal.

Defining layout performance on a cash-basis provides a metric that designers, engineers, and management alike can comprehend. Too often can valuable insight and information be lost during interactions involving these stakeholders. While a utility function is a common concept for engineers, it is less likely a concept known and comprehensible to management. This creates a language chasm between the stakeholders, which not only makes it difficult for information to effectively pass between them, but also inhibits effective collaboration. Leveraging a cash-based model bridges this gap, providing a common language in which all stakeholders can comprehend. This in turn helps to ensure that more effective communication and collaboration can occur between all stakeholders. Furthermore, it is management that holds the decision-making power. Managerial decisions almost always consider costs. Often management relies on metrics such as profit, net income, and retained earnings, to name a few, to inform their decisions. As such, a cash-based model was an ideal choice to deploy in this dissertation to evaluate the performance of a layout design. Moreover, these managerial accounting metrics of profit, net income, and retaining earnings are leveraged to form the foundation of the developed cash-based performance model.

The developed model summarizes the overall performance of a layout design with the accounting equation for retained earnings in the absence of dividends, provided below:

$$RE_t = RE_{t-1} + Net\ Income_t \tag{9}$$

where RE_t defines the returned earnings at the end of the current period, RE_{t-1} the returned earnings at the beginning of the period or end of the previous one, and $Net\ Income_t$ the net income for the current period. Retained earnings is an account that records the accumulated profits of a business [64]. Therefore, the returned earnings account value, at the end of the planning horizon, is representative of the system's cumulative performance over the entire horizon. As such, this ending balance becomes the objective function in the optimization algorithms of Stage One and Two. The goal of the algorithms is to then maximize this returned earnings metric. The design yielding the greatest retained earnings is then considered the best design of the provided scenario layout problem.

The net income component in the above equation represents the performance of the system, or design, in each of the periods of the planning horizon. Net income is synonymous to that of the systems profit. Elaborated, it is defined as the total revenue less all operating costs, business expenses, interest, and taxes paid out in a given period [65]. The developed model deploys the following equation to define the net income each period:

$$Net\ Income_t = Revenue_t - Costs_t - \phi_t \tag{10}$$

where $Revenue_t$ is the revenue for the current period, $Costs_t$ the costs for the current period, and ϕ_t is a cost penalty function, subtracted from the net income, to account for constraint violations in the current period. This penalty function will be elaborated on later when the constraint model of this dissertation is presented. For now, understand that this penalty function can be a form of an interest expense, business expense, or both. It is assumed though that there are no other prior or current debts other than those encapsulated in the penalty function, which is why there is no dedicated term for interest expenses in the above equation. Furthermore, it is assumed that taxes do not apply.

Inclusion of either of these would be relatively easy provided the modular nature in which the performance model was implemented. The above assumptions are a few of those implemented in this dissertation to simplify the analysis. More will be presented in subsequent sections and as needed. This, for now, is enough to gain an appreciation for the granularity of the implemented performance model.

Now, if there is no penalty for the current period, the net income equation simplifies to the equation for earnings before interest and taxes, also known as EBIT. In other words, EBIT for each period is just the revenue generated by the system ($Revenue_t$) less the costs incurred ($Costs_t$) to generate such revenue. The revenue generated each period is defined in this model as the value generated from the system producing products, where the specific products produced by the system and their associated market values are defined during the problem initialization step of the LIVE methodology outlined before. The costs component of the equation then is a compilation of several different costs associated with producing said products. Most of the costs considered in this dissertation are direct, in other words, direct costs of production. Included in this category are the material handling costs (MHCs); the metric often used in the literature to define the performance of a layout design. MHCs are only one of many costs that establish the cost of production though. As will be observed later when the costs accounted for in the model are presented, many other costs contribute to the cost of production. As mentioned before, data is knowledge and knowledge is power. Inclusion of such costs not only provides a more detailed evaluation of a layout design's performance, but also provides more data and therefore more power to make the correct design decisions down the road. In addition to the direct costs, indirect costs such as rearrangement costs and capital expenditures are also accounted for by the model.

Now before either of these revenue or cost functions can be outlined, two relevant discussions must precede to provide full closure to their formulations. The first of these addresses how the material handling distances, which are later leveraged to define the MHCs mentioned before, are established in this dissertation. This discussion also then consequently addresses the first major research gap identified in the beginning of this dissertation. The second is a presentation of the implemented process flow analysis. This analysis requires presentation before the revenue and cost functions as it establishes the system's actual production rates and therefore the overall revenue and costs. Before the process flow analysis or method of determining the material handling distances can be presented, a brief overview of how the objects physically interact with one another in the space is required to provide context to subsequent discussions.

4.4.1.1 Object Interactions

A multiple spacing interaction, unique to this dissertation, is deployed to simulate the difference between the space required to move about the objects in the environment and the space required to perform any necessary maintenance procedures on each object. The first, labeled as the walking spacing from here on out, identifies the closest distance to each object at which one can safely pass by. As a result, this walking spacing becomes integral in determining the flow distances between two objects. This is because the advanced method of determining the flow distances, which will be presented next, is built on the premise of providing the shortest path that does not violate such safety boundaries about the objects. The walking spacing is in turn used to simulate said boundaries of each object in the flow distance method. Additionally, it is these simulated walking boundary distances that are leveraged in Stage One to define the widths and heights of the objects in the placement algorithm. Defining the boundaries of the objects in Stage One according to these walking boundaries presents a major benefit when considering the constraints of the problem. This benefit will be observed when the constraint model is presented later.

The maintenance spacing on the other hand, represents the hard boundary constraint between two objects. In other words, two objects cannot be any closer than the summation of their individual maintenance spacing's. The maintenance spacing is object specific whereas the walking spacing is uniform across all objects. The maintenance spacing for infeasible regions, such as those representing interior walls, pillars, and those used to define arbitrary shaped facility layouts, are uniformly set to zero in this model. Further, an important restriction is placed on these spacing's relative to one another. This restriction is that the maintenance spacing be less than the walking spacing for every object. This, in conjunction with the walking boundaries being applied in Stage One, provides an added benefit when considering the constraints in the first stage. The only other restriction placed on both spacings is that they at least be greater than or equal to zero to ensure these boundaries fall outside the object's physical boundaries. The relationship between these different spacing boundaries is depicted in Figure 36. Note, in this and all subsequent figures the object's physical, maintenance, and walking boundaries are represented by black solid, dotted black, and red solid lines respectively and the I/O points by blue solid diamonds. The inclusion of this multiple spacing interaction concept required the inclusion of additional constraints; however, it also improves the detail of the formulation. It enables aisles to become a derived characteristic of the layout and a more accurate layout evaluation to be achieved, which was a core goal of this dissertation. Now that the object interactions have been outlined, the method developed to address the first research gap and define the material handling distances in this dissertation is presented.

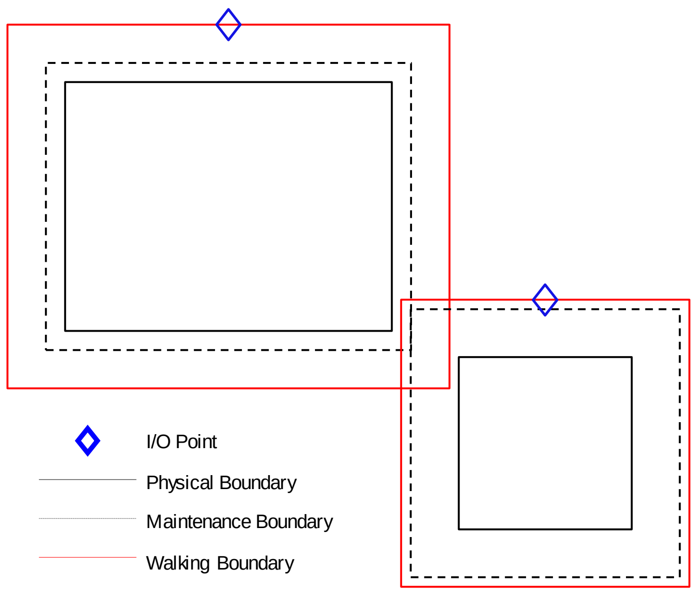


Figure 36 – Station spacing interactions

4.4.1.2 Advanced Material Handling Distance Method

As was mentioned before, MHCs are one of several direct costs accounted for by the developed model. The importance of the MHCs was also documented during the background and motivation chapter of this dissertation. As a reminder, it was identified that MHCs can contribute up to 50% of the operating costs and 70% of the total cost of producing a product [72]. Because of its importance, accurately modelling these MHCs is crucial. Moreover, it was identified in **Observation 3** that failing to account for flow feasibility when determining the material handling distances can result in suboptimal layout designs in practice. Then as was established later in **Assertion 9,** a material handling distance method that considers flow path feasibility was to be imperative to accurately evaluating a layout design such that suboptimal designs were avoided. With no such method implemented in the FLP literature, this presented the first major research gap. Therefore, an advanced flow distance method ensuring flow path feasibility was developed and deployed in this dissertation to provide closure of this gap. This advanced flow distance method is now presented.

The primary objective of the advanced flow distance method developed is to ensure flow feasibility in order to improve the accuracy of the layout evaluation. By using a branch and bound method, tailored to the problem, to determine the optimal feasible path distance (D_{ij}), this can be achieved. Such a method is sufficient since the problem is of relatively small size and the variables are discrete (e.g., the corners of the walking boundaries of each object). The developed tailored branch and bound method consists of three steps: the generation of initial candidate paths, a branching step which uses a splitting procedure to provide exploration, and a pruning step that systematically discards sub-optimal paths.

To generate the initial candidate solutions, or branches, a straight line is drawn between the two I/O points of a connection and all objects that are intersected by this line (i.e., violated) are identified. Understanding that to avoid violating the stations, the path must round the objects corner points that result in the largest deviation to each side of the line (i.e., largest normal distances to each side of the line), two branches can be formed per violated or bisected object. In other words, if two objects are bisected by this initial straight line, four candidate branches consisting of three points (the two connection points sandwiching the maximum violation point) are generated. Likewise, if three are bisected, six initial branches will be generated.

This process of taking the maximum normal distance points of the violated objects becomes the basis of the splitting procedure of the algorithm. This is visually demonstrated, in a limited capacity, in Figure 37. The next level of branches now has two segments that make up its potential path. Each one of these is evaluated independently and new branches are generated in a combinatorial manner where each branch contains the integration of a path deviation from each violated segment. This process continues for each subsequent branch until they are no longer in violation, thereby becoming a leaf, or until they are pruned. At every branch level the distances for each potential path are calculated by summing the direct distances of each of their segments as follows:

$$D_{ij} = \sum_{k=2}^{n} \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2} \tag{11}$$

where n defines the number of points that make up the path (which is one greater than the number of segments), k defines the leading point of the segment, and k-1 the trailing point. For example, the feasible paths generated in Figure 37 consist of two segments (i.e., $n = 3$). Determining the flow distance, D_{ij}, becomes the summation of the two segment lengths in this example.

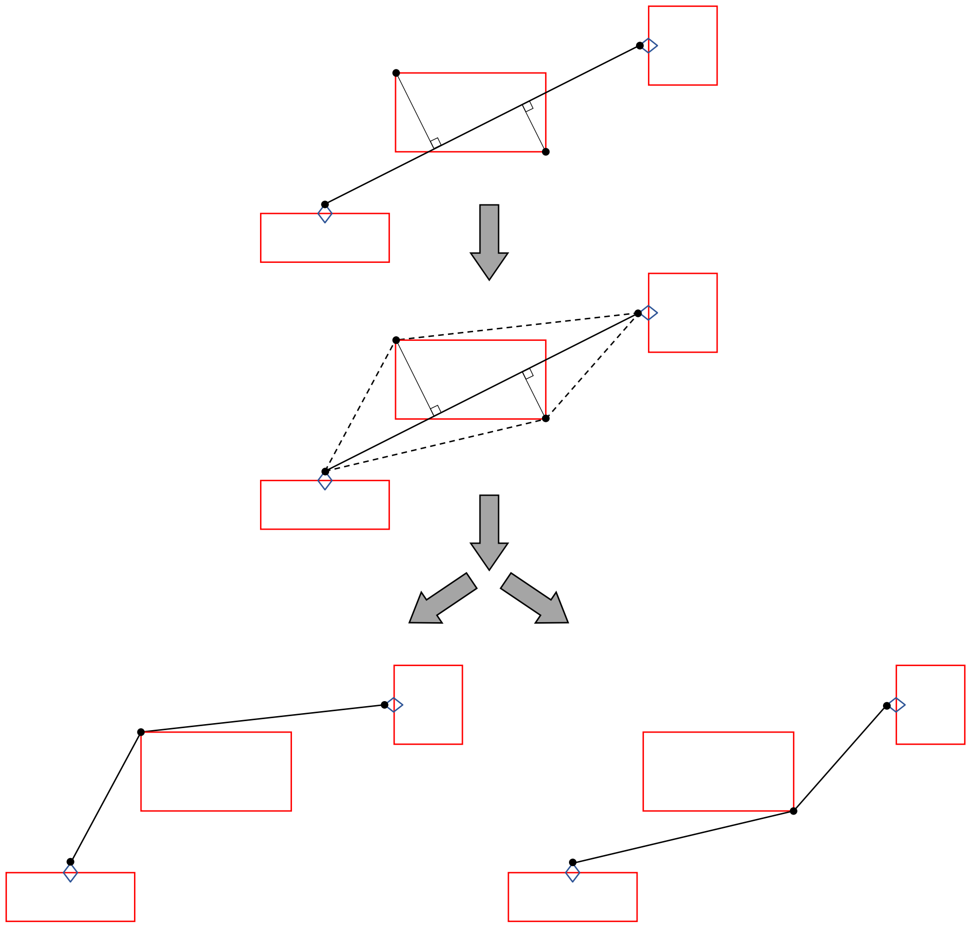


Figure 37 – Splitting procedure visualization for the optimal feasible path generation algorithm

The splitting procedure, detailed above, is based on the premise of deviating from a straight line; therefore, each subsequent split can only further increase the path distance. As a result, this enables the inclusion of elitism to be implemented in the form of a pruning step, which is only active in this formulation once the first feasible path is discovered. Once a feasible path is found, which is analogous to a leaf in this formulation, all other branches can be compared to this discovered path by comparing their distance as found using Equation (11). Any of these potential paths that fail to have a more optimal path distance are pruned from the search. This is possible since the basis of the formulation is that each subsequent branch can be no better than its parent branch as it incorporates a path deviation. This greatly improves the algorithm by systematically avoiding unnecessary computations.

The algorithm is also improved by upfront eliminating all corner points that are infeasible, which includes those falling outside the outer boundary, within infeasible regions, or within the walking boundaries of other objects. Additionally, to avoid unnecessary computations related to the duplication of paths, which are unavoidable occasionally, a check for paths that are the subset of others already established is included.

Inclusion of this flow distance method required additional constraints to be accounted for which act in avoiding unnecessary executions of this formulation when a feasible path is unachievable. For example, if an I/O point falls in an infeasible region then it would not be accessible. This inaccessibility would result in the algorithm persisting until it terminates at the implemented maximum branch generation limit that is included in the formulation. Although an infinite loop is avoided by this limit, excessive time would be spent performing computations that could be avoided upfront. These additional constraints help to accomplish this avoidance upfront. These constraints, along with others will be detailed later when the constraint model is presented.

The algorithm outlined above is executed for each unique process flow segment (object to object transfer of the products) present within each period of the layout design. Once all the material handling distances have been determined the process flow analysis of the system can then be performed.

4.4.1.3 Process Flow Analysis

The way revenue is generated is by transforming inputs (raw materials, labor, capital, etc.) into outputs (i.e. products in this case) which can then be sold in the market for a profit [101]. To provide this transformation, activities add value as the product proceeds through its process in the system. As it pertains to the problem of this dissertation, these activities and their capacities are known following the problem initialization step of the LIVE methodology. For reference, these activities occur at the station objects, which can be workstations, machines, staging areas, etc., but also between the stations when the products are being handled. It was also assumed in the initialization step that the products to be produced by the system each period of the planning horizon are known and thus defined. Furthermore, the process flows of each of these products, analogous to the sequence of station objects visited by the products, are also established in this step. Moreover, the distances between the station objects have since been determined with the deployment of the developed advanced distance method previously outlined. With the process flows and the capacities of the activities composing these processes known, a process flow analysis of the system is nearly possible.

The last remaining property that needs definition is the production rates of each of these products throughout the planning horizon. Fortunately, these production rates are another property defined during the initialization of the problem. Recall though, these production rates were defined before as desired production rates. This distinction was strategic. These production rates are those the designer chooses/hopes to achieve, not necessarily the rates in which the system can sustain. To know what the system can sustain and whether these rates are achievable is where the integration of the process flow analysis in this dissertation became necessary. Its inclusion also has the benefit of providing added insight into the system and the operations. Operation-based data such as the bottlenecks of the system and the utilizations throughout become available data, which can then be leveraged by the designer and management to make more informed decisions regarding the design of the layout.

The function of the process flow analysis in the performance model is thus twofold. It first functions in identifying if the provided production rates are feasible according to the system's capacity. If the current design of the system (i.e. layout design, assets present, labor capacity, etc.) cannot meet the prescribed production rates the process analysis then performs its second function of then determining what the actual production rates should be such that the system is operating at its maximum capacity. Determining these actual production rates is instrumental in ensuring that the layout designs considered in Stage One and Two are accurately evaluated.

In the developed model, the system's current operating capacities are determined by establishing the utilization levels of all the stations as well as the handlers of the system. The latter is important as it directly considers the layout design. Layout's that are better configured, and assuming all else constant, will produce lower handler utilization levels and vice versa. The general equation deployed to determine the utilization of each station and the handlers is as follows:

$$Utilization\ (U) = \frac{Production\ Hours}{Available\ Hours} \tag{12}$$

where the available hours are defined as the number of work hours per day (*WH*) and the production hours is the required production time needed, in a given day, to sustain the production rates provided.

4.4.1.3.1 Station Utilizations

For a specific station, this production time constitutes the time it takes to produce, at the provided rate, all products for which it is involved in producing (i.e. all product-process for which it is a part of the process-flow) and for which is present (different depending on the scenario and period of the layout design). The equation implemented to establish the production time for each relevant station is as follows:

$$Production\ Hours = \sum_{j} \frac{PR_j}{CAS_j} \tag{13}$$

where *j* are the processes for which the station of interest is a part of, *PR* are the current production rates, and *CAS* are the station's capacities for each of the *j* product-processes. The ratio of *PR* to *CAS* can be understood as being each product-processes contribution to the station's production hours. The summation then establishes the number of hours then needed in each day to sustain that provided *PR*. Dividing this by the work hours per day, as established in Equation (12), provides the utilization level of the station. This computation is performed for each relevant station in a given period of the layout design and moreover, at each forecasting point of the scenario structure defined for the problem. *WH* is defined at each of these forecasting points and so too is the *PR*. Any stations having a utilization level exceeding a value of one, or when converted to a percent, a value of a hundred (100%), is indicative that the station cannot sustain the provided production rates (*PR*) and action must then be taken to remedy this violation of the system's maximum capacity.

4.4.1.3.2 Handler Utilizations

Likewise, the production time for the handlers constitutes the time it takes to move each product throughout the space (i.e. from station to station) per the provided production rates. The equation implemented to establish this production time is as follows:

$$Production\ Hours = \sum_j \frac{PR_j}{CBS_j} \tag{14}$$

where *PR* are the production rates like before and where *CBS* is the capacity between-station for product-process *j* (i.e. process flow segment), which is defined as follows:

$$CBS_j = \frac{1}{\sum_k \frac{1}{CBSS_{jk}}} \tag{15}$$

where *CBSS* is the capacity of a between-station segment *k* of the product-process *j* and is a function of the segment handling distance as follows:

$$CBSS_{jk} = \frac{CH_{jk}}{D_{jk}} \tag{16}$$

where CH_{jk} is the handler flow-rate capacity for segment *k* of product-process *j* and D_{jk} is the segment flow distance as established by the material handling method outlined before. It is relevant to note that in this developed model, the between processes were assumed to be independent of each other (i.e. two products with the same between process do not share transport between, i.e. two separate carts are required). Though this assumption is perhaps far from ideal, especially in the case of job shop environments, it was implemented to reduce the complexity of the model.

Now the inverse of the inverse summation in Equation (15) enables the capacities of each individual segment of the j product-process to be joined to establish the overall capacity of the product-process j. Dividing this then from the production rates enables the handling production hours for each product-process to be determined and subsequently the utilization levels of the handlers. Like before with the stations, any product-process yielding a handler utilization exceeding a value of one or a hundred percent is indicative that the prescribed production rates cannot be sustained by the handlers between the stations. Just like before, this requires addressment.

4.4.1.3.3 Dynamic Adjustment of the Production Rates

Now for any situation where the station or handler utilizations exceed that in which is possible per the system's capacities, the production rate requires adjustment in order to bring the system back in line with what is possible. Not doing so would result in a design appearing far better than it actual would be in practice; therefore, it was paramount that such situation were addressed and this problem remedied. In the developed model, a method of dynamically adjusting the production rates such that the system's maximum capacities were then met was implemented. To perform this adjustment, only those product-processes (j) associated with station or handler violations (i.e. utilizations exceeding 100%) are adjusted. Those not associated; do not need adjustment as their respective stations or handlers are not operating beyond maximum capacity as it is.

Adjustment to them would be counterproductive as it would only reduce the profitability of the system. For the product-processes that are associated with violated handlers or stations the production rates of these can be adjusted in one of two ways depending on how the designer has established how to handle such situations while defining the scenario in step one of the LIVE methodology.

The first option is to adjust the production rates of these relevant product-processes while maintaining the originally prescribed product-process production rate ratios. In other words, if three processes are to be adjusted and their relative rates are eight, four, and two respectively, then the ratios would be 4:2:1. The three processes would then be adjusted according to these ratios until every one of these processes meet the system's capacities. In other words, the utilizations of the handlers and/or stations coupled to these product-processes all are then less than or equal to hundred percent. The second option is to adjust the production rates while maintaining the most profitable product-processes. In this case, the least profitable product-processes are decreased or eliminated first before the more profitable ones are. This is done until the system capacities are met by all stations and/or handlers associated with these processes. The estimated manufacturing cost and market value inputs defined earlier in step one of the LIVE methodology, are leveraged in this option to identify the order of these relevant processes from least profitable to most profitable. Regardless of which option is deployed by the designer, MATLAB's fmincon function is leveraged to efficiently adjust these production rates of the product-processes until the system's maximum capacities are met.

Now because the computations are dependent on scenario inputs conditions that change across the horizon, this utilization check is performed at each of the distinct scenario horizon forecasting points. Furthermore, because the handler utilizations are a function of the handling distances, which are a function of the placements of the stations in the space (i.e. layout configuration), these computations need be performed for each unique layout design. The utilizations for each station and the handlers on a product-process-basis are recorded and therefore accessible to the designer. Moreover, the original and adjusted production rates are retained for the designer to review posterior. The availability of this data can help better inform the designer on the performance of the layout design(s) and strategic business decisions considered in the scenario. With the production rates adjusted to meet the systems capabilities, the revenues and costs can then be computed for the system.

4.4.1.4 Revenue and Costs Functions

The revenue and cost components of Equation (10) and the functions implemented in the model to define them are now presented. The discussion that follows provides an understanding of how the revenue and costs for each period were defined in this model, while the mathematical integrations of the lower level functions are omitted provided their simplistic forms.

The implemented functions are product-process based. In other words, each product-process uniquely contributes to the bottom-line of the system. Such an approach is a key enabler to allowing process changes to be analysed by the designer. The capability to analyse process introductions, eliminations, changes, and fluctuating production demands enables a wide range of strategic business decisions regarding the design of the system and layout to be considered. Before proceeding, two overarching assumptions of the implemented functions are as follows:

1) A year consists of 52 weeks, each week consisting of 7 days

2) All conditions either behave discretely or linearly across the forecasting segments of the horizon

4.4.1.4.1 <u>Revenue Function</u>

The implemented revenue function is product-process-based as mentioned before. As such, determining the revenue for a given period *t* requires the contribution of each product-process to be established. Moreover, each period *t* can span one or multiple segments of the forecasting horizon. This dependency is demonstrated in Equation (17), which defines the revenue for a given period *t*:

$$Revenue_t = \sum_i \sum_j R_{ij} \tag{17}$$

where *i* are the horizon segments in which period *t* spans, j are the relevant product-processes of the period, and R_{ij} is the contribution of a product-process *j* in segment *i* to the revenue of the period. Recall, that while within a period the layout configuration remains unchanged, the conditions are likely not to. Given that the conditions are likely to change, the revenue, being a function of these conditions, across each segment *i* will too vary. Now in the case where the period spans exactly one of the forecasting horizon segments, $Revenue_t$ becomes equivalent to the summation of each product-processes contribution across the segment ($\sum_j R_j$). Figure 38, below, is a recreation of the example provided back in the problem initialization section when defining the format of the input conditions. This figure has been supplemented here whereby the horizon segments that the above summation would encompass for period two, *t = 2* are labelled. In this specific example, two segments would be spanned.

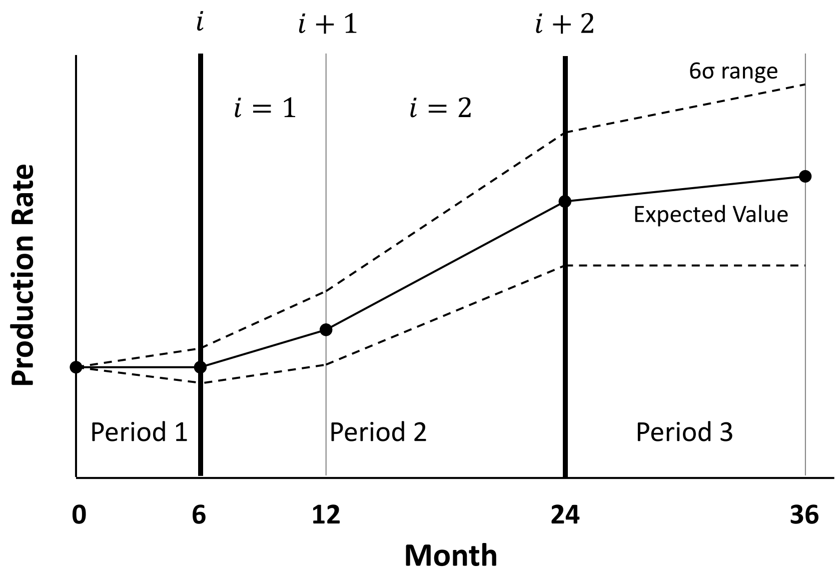


Figure 38 – Segment indexing example for production rate condition

The equation for each product-process *j's* contribution to the revenue in segment *i* of the period *t* is as follows:

$$R_{ij} = \int_{t_i}^{t_{i+1}} MV_j(t) \cdot PR_j(t) \cdot dt \tag{18}$$

where the revenue is just a function of two conditions: the market value, *MV*, of the product-process *j*, which was defined in step one of the methodology, and its production rate, *PR*, which was either confirmed as being possible or altered accordingly in the process analysis outlined before. Note, t_i is the time at the start of the segment, t_{i+1} the time at the end of the segment. The integration in Equation (20) would be carried out twice for the prior example where period two was being evaluated. The functional forms of these two conditions across the segments are as follows:

$$MV_j(t) = MV_{ij} + \frac{MV_{i+1,j} - MV_{ij}}{t_{i+1} - t_i}(t - t_i) \tag{19}$$

$$PR_j(t) = \frac{52WD_i}{12}\left[PR_{ij} + \frac{PR_{i+1,j} - PR_{ij}}{t_{i+1} - t_i}(t - t_i)\right] \tag{20}$$

where *WD* is the work days per week (defined in the problem initialization) and defined discretely across the segment and according to the value at the start of the segment. The leading term in the production rate function ensures that it is converted appropriately from units/day to units/month to align with the monthly scale of the planning horizon and scenario structure. Integrating these functions, multiplied together, across the segments and for each product-process yields the revenue for the period.

As can be observed, the functional form of these conditions is linear. As was established before, the behaviour of the conditions across the forecasting segments is assumed to be linear in this dissertation. With that said, the implementation is modular enough that if one were to choose a different functional form, the integration could be performed without major changes to the underlying model. In future discussions of the cost function, this depth of discussion will not be achieved. Note though that the forms of the low-level functions are all linear in the model.

4.4.1.4.2 Cost Function

With the implemented revenue functions defined, the costs associated with generating the revenue and moreover those costs captured in the developed model are presented. The implemented cost function encapsulates a variety of costs, including direct and indirect costs of production as well as capital expenditures. The summarizing cost function of the model is presented below:

$$Costs_t = DCP_t + ICP_t + CAPEX_t \tag{21}$$

where DCP_t are direct costs of production, ICP_t the indirect costs of production and $CAPEX_t$ the capital expenditures for the period. In the developed performance model, the ICP_t are a function of the DCP_t and as such will be presented first.

4.4.1.4.2.1 Direct Costs of Production

The direct cost of production component, DCP_t, encompasses several different costs associated with producing the products. Each of these costs vary as a function of the forecasting segment and thus, like before, must be summed across any segments encompassed by the given period as follows:

$$DCP_t = \sum_i C_i \tag{22}$$

where i is the forecasting segments encompassed by the period and C_i the cumulative cost of each segment. This cumulative cost of each segment can be further decomposed as follows:

$$C_i = MCAS_i + MCBS_i \tag{23}$$

where $MCAS_i$ are the manufacturing costs at stations and $MCBS_i$ the manufacturing costs between-stations. The former is related to the activities performed at the stations to transform the product while the latter relates to the handling activities performed to move products from one station to the next.

The costs at stations can even further be decomposed into the core costs of the cost model as follows:

$$MCAS_i = DLPC_i + DLSC_i + DCC_i \tag{24}$$

where $DLPC_i$ is the cumulative direct production labor costs, $DLSC_i$ the cumulative direct setup labor costs, and DCC_i the cumulative direct consumable costs for segment i. Each of these are summed across the relevant product-processes for the current period t just as was done before with the revenue. Moreover, the $DLPC_i$ and $DLSC_i$ are summed over each of the stations associated with each of these product-processes. This mathematically is as follows:

$$DLPC_i = \sum_j \sum_s DLPC_{ijs} \tag{25}$$

$$DLSC_i = \sum_j \sum_s DLSC_{ijs} \tag{26}$$

$$DCC_i = \sum_j DCC_{ij}\, DCC_i = \sum_j DCC_{ij} \tag{27}$$

where *j* is the relevant product-process of the segment, *s* the stations of the product-process *j* and as such $DLPC_{ijs}$ and $DLSC_{ijs}$ are the direct production and setup labor costs associated with station *s* producing product *j*. DCC_{ij} is then the direct consumable cost associated with producing product *j*. These costs are defined by the following integration equations:

$$DCC_{ij} = \int_{t_i}^{t_{i+1}} DCC_j(t) \cdot PR_j(t) \cdot dt \tag{28}$$

$$DLPC_{ijs} = \int_{t_i}^{t_{i+1}} \frac{NW_s \cdot LCAS_s}{CAS_{js}} \cdot LCA(t) \cdot PR_j(t) \cdot dt \tag{29}$$

$$DLSC_{ijs} = \int_{t_i}^{t_{i+1}} \frac{NW_s \cdot LCAS_s}{SAS_{js}} \cdot \frac{LCA(t) \cdot PR_j(t)}{SR_j(t)} \cdot dt \tag{30}$$

where DCC_j is the direct consumable cost to produce a single product *j*, NW_s are the number of workers at station *s*, $LCAS_s$ is the average labor rate of a workers at station *s*, CAS_{js} the capacity of station *s* in producing product *j*, SAS_{js} in a similar fashion the setup capacity of station *s* in setting up for product *j* production, LCA the labor rate adjustment factor, and $SR_j(t)$ the setup frequency of product j across each segment. Recall, all these properties were defined during the problem initialization step of the LIVE methodology. Their elaborated definitions, units, and uses are outlined in Appendix C. Integrating these functions over the segment for each process and further station, when necessary, establishes the direct costs of production at the stations for the segment.

Before continuing, it is relevant to note that as demonstrated in Equation (30), the setup times are allocated on a daily-basis. Though in some cases it may not be necessary or practical to setup production for a product on a daily-basis, a simplifying assumption was made. It was assumed that setups be distributed across the time segments. It was believed that this would more accurately account for the setup costs within the segment and furthermore enable the impact that setup frequency has on the costs and personnel utilization (time associated with setup) to be observed.

Continuing with the direct costs of production component of the summarizing cost function, the direct costs of production between the stations, $MCBS_i$ accounts for the costs associated with handling the products. It is decomposed further into two components as follows:

$$MCBS_i = DLHC_i + OHC_i \tag{31}$$

where $DLHC_i$ is the cumulative direct handling labor costs and OHC_i the cumulative other handling costs, more commonly referred to in the literature as the material handling costs (MHCs). In this model, the traditional MHC function is broken into two different components for more cost granularity. It should be noted that reducing the labor costs to zero will render the first component of Equation (31) irrelevant leaving just the second component which can then be leveraged to emulate the literature standard MHC function as both these are a function of the material handling distances as will be observed. Both these cumulative costs are a function of the product-process and handling segment of the process flow. These are summed across these dimensions as follows:

$$DLHC_i = \sum_j \sum_k DLHC_{ijk} \tag{32}$$

$$OHC_i = \sum_j \sum_k OHC_{ijk} \tag{33}$$

where *j* as always is the relevant product-processes of the segment and *k* is the handling segment of the product-process *j*. As such, $DLHC_{ijk}$ and OHC_{ijk} are then the direct handling labor cost and other handling cost for segment *k* of product-process *j* in segment *i*. To obtain these costs, the following integration equations are deployed:

$$DLHC_{ijk} = \int_{t_i}^{t_{i+1}} \frac{NH_{jk} \cdot LCBS_{jk}}{CBSS_{jk}} \cdot LCA(t) \cdot PR_j(t) \cdot dt \tag{34}$$

$$OHC_{ijk} = \int_{t_i}^{t_{i+1}} OHC_{jk} \cdot D_{jk} \cdot PR_j(t) \cdot dt \tag{35}$$

where NH_{jk} is the number of handlers for segment k of product-process j, $LCBS_{jk}$ the average handler labor cost for segment k of product-process j, OHC_{jk} the other handling cost per unit product for each handling segment k of product-process j, D_{jk} the handling distance for segment k of product-process j, and $CBSS_{jk}$ the capacity of a between station segment k, which is the same one used before when computing the utilizations in the process analysis. This property is a function of the handling distance making then the direct handling labor costs also a function of the handling distances. As such, a layout better configured will yield lower handling labor costs and furthermore lower other handling costs, if relevant. Integrating these functions over the segment for each process establishes the direct costs of production between the stations, $MCBS_i$, for the forecasting segment. Moreover, with its establishment, all direct costs of production accounted for in the model are then defined, thereby making DCP_t of the summarizing cost function provided in Equation (21) known.

4.4.1.4.2.2 Indirect Costs of Production

The indirect cost of production component, ICP_t of Equation (21), encompasses two subcategories of costs in the model. These include the costs associated with rearrangement (unique to the DLP) as well as those associated with the direct production of the products. This is mathematically depicted as follows:

$$ICP_t = ILC_t + PRIC_t + RC_t \tag{36}$$

where $PRIC_t$ is established in the model by leveraging the direct costs of production outlined before. The production related indirect costs (*PRIC*) are established as a percentage of the direct costs of production on a product-processes-basis as follows:

$$PRIC_t = \sum_i \sum_j p_j C_{ij} \tag{37}$$

where p_j is the percentage of the product-process j's direct costs of production C_{ij} for segment i of the horizon. C_{ij} is the same as C_i before just not aggregated across the product-process in the lower level functions provided earlier (Equations (28) - (30), (34), and (35)). These indirect costs account for expenses related to utilities usages, rent (based on the space of the floor that the product-process encompasses), a portion of the insurance costs and other indirect costs on a product-process-basis. It should be noted that though administrative and selling expenses are not directly captured in the model they can be accounted for in this cost category. Adjusting the percentage to include such selling and administration expenses associated with each product *j* enables these expenses to be accounted for and furthermore enables the designer to consider such expenses on a product-process-basis. This allocated approach provides an accurate account for such costs which can vary based on the product (e.g. some products are harder to sell than others).

The second component in Equation (36), ILC_t, represents the indirect labor cost. In other words, the costs associated with workers sitting idle and not contributing to value adding activities. This indirect labor cost is established as follows:

$$ILC_t = \sum_i TLC_i - DLC_i \tag{38}$$

where DLC_i and TLC_i are defined as follows and represent the direct and total labor costs across a forecasting segment:

$$DLC_i = \sum_j DLPC_{ij} + DLSC_{ij} + DLHC_{ij} \tag{39}$$

$$TLC_i = \sum_{HR} TLC_{i,HR} \tag{40}$$

where direct labor costs, $DLPC_{ij}$, $DLSC_{ij}$, and $DLHC_{ij}$ from earlier are leveraged to establish the direct labor costs of the segment and $TLC_{i,HR}$ is the total labor cost across the segment and is defined as:

$$TLC_{i,HR} = \int_{t_i}^{t_{i+1}} \frac{52 \cdot WD_i \cdot WH_i}{12} \cdot LC_{HR} \cdot LCA(t) \cdot dt \tag{41}$$

where LC_{HR} is the total labor cost for all personnel of the system and where it was assumed that all personnel on average work the standard number of work days per week and work hours per day as defined by the designer during the problem initialization step.

The other indirect costs accounted for in the model are those associated with the rearrangement of the layout from one period to the next. This cost is decomposed into two components. The first is the cost associated with physically moving the stations in the space and the second is the loss of production that comes as a by-product of having to cease the production of any product-processes that are affected by this rearrangement. In other words, any product-processes involving stations which are to be moved. The summarizing function for these rearrangement costs are as follows:

$$RC_t = MVC_t + LPC_t \tag{42}$$

where MVC_t is the movement cost and LPC_t the loss of production cost for the period. Unlike the costs presented before, the movement costs are solely a function of how the layout configuration has changed and therefore does not require integration across any forecasting segments. The equation implemented to define the movement cost is as follows:

$$MVC_t = \sum_{s} \left((CM_s + CRSC_s) \cdot d_s + CI_s \cdot r_s \right) \tag{43}$$

where *s* represents all common stations between the previous and current period layouts, d_s the rectilinear distance change of station s, r_s the station's rearrangement state, CM_s the cost of moving the station on a unit distance-basis, $CRSC_s$ the cost of rerouting any supporting conduit of the station on a unit distance-basis, and CI_s the cost of uninstalling and reinstalling station *s*. The magnitudes of CM_s and $CRSC_s$ both depend on the distance moved, which is why it is multiplied by the change distance d_s, while CI_s is only dependent on the rearrangement state r_s. This state has a value of zero when the station remains unchanged both from a position and orientation standpoint while if either its position or orientation changes it will have a value of one. In practice the movement of a station requires that the station be uninstalled and reinstalled (e.g. a CNC machine unbolted and bolted back down to the floor after being moved). This component accounts for this occurrence. CM_s and $CRSC_s$ account for the labor and equipment costs associated with picking up and moving the station to another location (e.g. forklift and operator cost) along with the need to reroute supporting conduit (e.g. HVAC, electrical wiring, network cables, etc.) to the station's new location in the layout.

The other component of the rearrangement costs is the loss of production, LPC_t, that results as a by-product of the rearrangement. The loss of production cost accounts for the potential profit lost from operations that have to be shut down temporarily while the layout is rearranged. It was assumed that only those product-processes associated with a station that is displaced in any way (moved or rotated), need be halted during the rearrangement phase. Several additional assumptions regarding this rearrangement phase were also made. First, it was assumed that rearrangement is to occur at the onset of the period. Second, the duration of this rearrangement phase was assumed to be equivalent to the longest rearrangement time amongst the stations being rearranged. In other words, all influenced product-processes are halted for the same duration of time, that time being equal to the maximum time amongst the impacted stations. This assumption was made to simplify the process of computing the loss of production cost. Another assumption made was that this rearrangement time be based on the number of working days per week (*WD*) and the working hours per day (*WH*) at the start of the rearrangement. Provided that the rearrangement occurs at the onset of the period, these conditions coincide with the start of the period and thus a known forecast segment point (*i*). It was also assumed that rearrangements are sufficiently spread apart such that operations can restart after rearrangement and before the next one commences.

Now to determine this rearrangement duration, the rearrangement times for each station first require definition. These times are determined as follows:

$$T_s = t_{install,s} + t_{uninstall,s} + \frac{d_s}{MVR_s} \tag{44}$$

where $t_{install,s}$ is the time to reinstall the station in the new location, $t_{uninstall,s}$ the time to uninstall the station in the old location, d_s the rectilinear distance moved (same as in Equation (43)), and MVR_s the nominal rate in which the station can be safely moved. The last term allows the rearrangement time to then become distance-based. Like before, these are known from the problem initialization step of the LIVE methodology. To then determine the rearrangement duration for all impacted product-processes, Equation (45) is deployed:

$$t_r = \max_s(T_s) \tag{45}$$

where the maximum station rearrangement time is found, and the rearrangement duration set to this time. With the rearrangement duration for all impacted product-processes the same, per the earlier noted assumption, the loss of production for these impacted processes can then be determined. The summarizing equation for computing this loss of production cost is as follows:

$$LPC_t = Revenue_{t_r} - \left(DCP_{t_r} + PRIC_{t_r}\right) - TLC_{t_r} \tag{46}$$

where t is the current period and the $Revenue_{t_r}$, $,DCP_{t_r}$, $PRIC_{t_r}$, and TLC_{t_r} are the revenues, direct costs of production, indirect costs of direct production, and total idle labor costs for the impacted product-processes and stations over the rearrangement duration t_r. The above components are fundamentally the same as those presented before with subscripts *t*. The only difference is, instead of integrating across all segments of the period, the integrations proceed only over those segments encompassed by the rearrangement duration. If the rearrangement occurs within the first segment and this rearrangement corresponds to the $i=1$ forecasting point, then the integrations would proceed from $t_{i=1}$ to $t_{i=1} + t_r$. Additionally, these equations are only summed over those product-processes, *j*, that are impacted by the rearranged stations. In light of this understanding, the $Revenue_{t_r}$ can be observed as only the revenue generated over the duration for just the impacted product-processes. The term in the parenthesis can be understood as being the direct and indirect costs of production that are coupled to these product-processes. The joining of these two components can be viewed as the profit lost from these halted product-processes. The last term is a rather important addition. This term accounts for the idle labor costs associated with these product-processes. As workers attached to these stations sit idle as the rearrangement occurs, they are not contributing any added value to the system, yet they are remaining paid. This last term accounts for such idle labor. With the loss of production for the period (LPC_t)_defined, the indirect cost component, ICP_t,of the summarizing cost function is then also defined.

4.4.1.4.2.3 Other Valuable Cost Metrics

In addition to the above outlined costs, other valuable metrics, based on these are also computed in the developed performance model for added insight into the system and its operational performance. The first is the utilization level of human resources, i.e. workers. This utilization is established as follows:

$$U_{HR,i} = \frac{DLC_i}{TLC_i} \tag{47}$$

With the revenue and costs, with the exception of *CAPEX*, established, the profit margin on a period, forecasting segment, and product-process-basis can be defined as follows:

$$PM_{t|i|j} = Revenue_{t|i|j} - Costs_{t|i|j} \tag{48}$$

Where when evaluating Equation (48) on a forecasting segment *i* and product-process-basis *j* the cost function above excludes those costs associated with rearrangement and moreover any *CAPEX*. On the other hand, when evaluating on a period *t* basis, the rearrangement costs are included. Furthermore, it also includes the *CAPEX* component of the cost function, which is the focus of the next discussion.

4.4.1.4.2.4 Capital Expenditures (CAPEX)

The last component yet to be defined of the summarizing cost function is *CAPEX*, or the capital expenditures of the period. This cost is unique to the dynamic layout problem considered in this dissertation and accounts for expenditures related to the acquisition or sale of a station (machine, workstation, equipment, etc.). It is assumed in the model, that this is the only form of capital expenditures present. Moreover, the acquisition cost includes all costs related to the acquisition and initial installation of the station while the salvage value is that received for a station less the costs associated with removing the station from the environment. The capital expenditure for each period is thus as follows:

$$CAPEX_t = \sum_{sa} Aquistion\ Cost_{sa} + \sum_{ss} Salvage\ Value_{ss} \tag{49}$$

where *sa* are the stations acquired at the onset of the period, *ss* the stations sold following the previous period, and where the salvaged values are subtracted from the acquisition costs to indicate a gain in capital.

Another assumption made here is that the total cost of acquiring and installing a station is applied all at once in the period in which it is originally acquired. Conventionally, this acquisition cost would be realized only as the station depreciates over its life. In other words, the cost would be realized over a span of time, not all at once. This assumption was made as a result of the planning horizon being finite in length and as such the total acquisition cost, if applied in this manner, could potentially not be realized completely within the analysis of the horizon. This assumption however ensures that regardless of the stations life span, the full cost of the station would be accounted for when considering the performance of the layout across the provided horizon, thereby providing an accurate evaluation of the system and layout design. With the *CAPEX* component of the cost function now defined for the model, the summarizing cost function implemented is completely established and therefore so too are two of the three components of the system performance summarizing net income equation presented earlier, Equation (10). The third will be presented later when the developed constraint model is discussed.

4.4.1.5 Local Robustness to Production Uncertainty

The above discussion highlights the developed performance model's fundamental structure, assumptions, and equations as it pertains to defining the nominal revenues and costs of the net income equation presented before. Until now, the implementation of the localized robustness method has yet to be addressed, and for good reason, as its implementation leverages all the prior equations. In the model, Norman and Smith's statistical method was implemented to account for production uncertainty. Though the fundamentals of their method were adopted, much the rest is different as a result of the much more detailed performance model developed here. In Norman and Smith's implementation, the objective function was defined using a statistical percentile of the rectilinear-based material handling cost metric as follows:

$$L(\Pi) = \sum_j PR_j \cdot MHC_j + z_p \sqrt{\sum_j \sigma_j^2 \cdot MHC_j^2} \tag{50}$$

where $L(\Pi)$ is the objective function for layout Π, PR_j is the production rate of product j, MHC_j is the material handling cost per unit of product j, σ_j^2 the production variance of the production rate, and z_p the standard normal *z* value for percentile p. The first term in this equation can be understood as being the expected value of the material handling costs while the square root portion of the second term the standard deviation of the material handling costs for layout Π. Considering this, Equation (50) can be presented as follows:

$$L(\Pi) = E(\Pi) + z_p s(\Pi) \tag{51}$$

where $E(\Pi)$ is the expected term and $s(\Pi)$ the standard deviation term mentioned before. Extending to the dynamic form of the layout problem produces the following variant of this equation:

$$L_t(\Pi_t) = E_t(\Pi_t) + z_p s_t(\Pi_t) \tag{52}$$

where t is representative of the equation applying on a period-basis. Further extension of this statistical percentile approach requires the terms of this equation to be redefined to account for the significantly more comprehensive performance model of this dissertation. Instead of $E(\Pi)$ being the expected value of just the material handling costs, it now becomes the expected value of the layout design's net income and similarly, $s(\Pi)$ its standard deviation for the provided period. In other words, $E_t(\Pi_t)$ is synonymous to $Net\ Income_t$, which is calculated by leveraging Equation (10) from before. As such, in the absence of uncertainty, Equation (52) reduces to that of Equation (10) and the performance model of before remains as is. In the presence of uncertainty however, an additional term is, by extension, effectively appended to the summarizing retained earnings objective function presented at the beginning of this section, Equation (9) for reference. This additional term can be understood as accounting for the uncertainty associated with the production rate and is the second term on the right-hand side of Equation (52).

Now as this applies across a period and not just on a per unit time-basis, this $s_t(\Pi_t)$ term is expanded to encapsulate the time spanned for the provided period as follows:

$$s_t(\Pi_t) = \sum_i s_i(\Pi_t) \tag{53}$$

where, like before, *i* is the forecasting segments spanned by period *t* and $s_i(\Pi_t)$ is the standard deviation of the net income for segment *i*.

$$s_i(\Pi_t) = \int_{t_i}^{t_{i+1}} s(\Pi_t, t) \cdot dt \tag{54}$$

where $s(\Pi_t, t)$ is the standard deviation of the net income as a function of time and thus integrated across the time range of segment *i* (t_i to t_{i+1}). As was assumed before to simplify the performance model, it is again assumed here that $s(\Pi_t, t)$ behaves linearly across the segment and can thus be defined as follows:

$$s(\Pi_t, t) = \frac{52WD_i}{12}\left[s(\Pi_t, t_i) + \frac{s(\Pi_t, t_{i+1}) - s(\Pi_t, t_i)}{t_{i+1} - t_i}(t - t_i)\right] \tag{55}$$

where $s(\Pi_t, t_i)$ and $s(\Pi_t, t_{i+1})$ are the standard deviation of the net income at forecasting segment *i* and *i+1* respectively. To understand how these are then defined in the developed model, Equation (50) is revisited where it is observed that this standard deviation term is a function of both the production rate variances and material handling costs per unit of product *j* squared. Extending this then to the performance model of this dissertation requires that the latter be redefined as the profit margin per unit of product *j* squared as follows:

$$s(\Pi_t, t) = \sqrt{\sum_j \sigma_j(t)^2 \cdot PM_j(t)^2} \tag{56}$$

where *t* would be either t_i or t_{i+1}, $\sigma_j(t)^2$ the production rate variance at time *t* (not to be confused with period *t*), and $PM_j(t)^2$ the profit margin per unit of product *j* squared at time *t*, which can be determined by leveraging a modified version of the product-process *j* variant of Equation (48) from before. Note that as was mentioned before, this equation when in this form excludes the rearrangement and *CAPEX* cost terms in the cost function thereby allowing it to accurately define the profit margin of product-process *j* related activities. A modified version of this equation is required as in its original form, Equation (48) yields the profit margin of product-process *j* on a cumulative-basis (multiplied by the production rate, PR_j) not a per unit-basis as is needed here. Moreover, the lower level integrations of the cost function across the segments are not to be performed. This is because the desire is to define the profit margin on a per unit basis at a specific point in time, where these points in time correlate to the forecasting points in the horizon as is depicted for an example problem in Figure 39. With this understanding, $PM_j(t)$ is then defined as follows:

$$PM_j(t) = MV_j(t) - MC_j(t) \tag{57}$$

where $MV_j(t)$ is nothing more than the designer supplied input market value condition and $MC_j(t)$ the calculated manufacturing cost for a single unit of product-process j at time t in the horizon. The former is then like $Revenue_j$ in Equation (48) while the latter is like that of $Costs_j$. If both these were to be multiplied by the production rate, PR_j, and then integrated over a segment they would then be equivalent to these terms in Equation (48). An easier way of achieving this without the need to alter the equations outlined before is to assign the production rate to unity for all product-processes and then not perform the integration, rather instead evaluating only at the forecasting points in order to establish $MC_j(t)$ at time t (i.e. t_i, t_{i+1}, …).

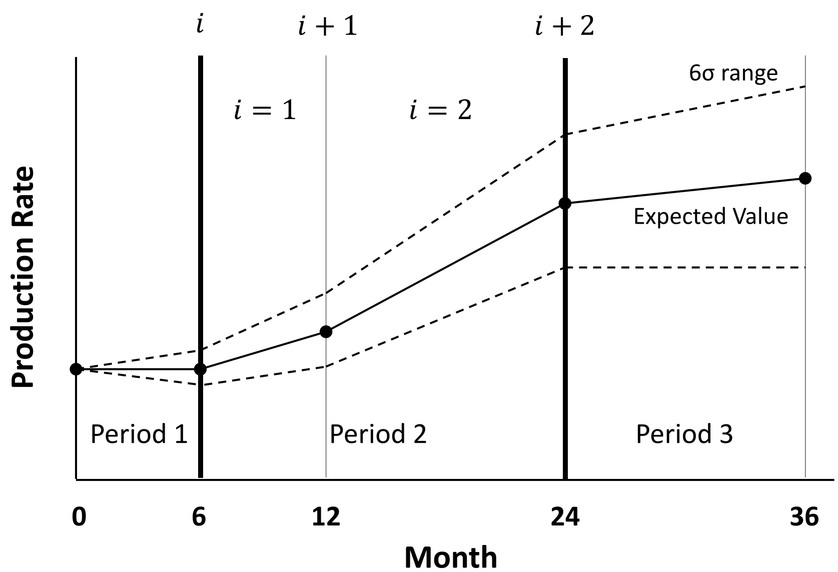


Figure 39 – Example problem

Now as for the production rate variances, $\sigma_j(t)^2$, these are defined by leveraging the designer supplied coefficient of variances for the production rates, noted at the beginning of this chapter when the problem initialization step of the LIVE methodology was outlined. The definition of the production rate variances are then as follows:

$$\sigma_j(t)^2 = \left(c_{v_j}(t) \cdot PR_j(t)\right)^2 \tag{58}$$

where $c_{v_j}(t)$ are the coefficients of variance across the planning horizon for the production rates, as established by the designer. The advantage of leveraging coefficients of variance is that they are a standardized measure of dispersion that provides the variability in relation to the expected production rate [5]. Mathematically a coefficient of variance is characterized as $c_v = \sigma/\mu$ where in this example the expected production rate replaces μ. In other words, the coefficients characterize the volatility of the production rates on a percentage-basis. For example, a coefficient of 10% (or 0.1) indicates that the standard deviation for a production rate of 10 units per day would then be 1 unit per day. Defining the uncertainty in this manner is advantageous as it allows for the standard deviation to then scale as the expected production rate scales.

Now that Equation (52) has been completely defined, it can be observed that this function enables production uncertainty to be explicitly considered on a localized continuous-basis. Additionally, said variability is provided on a product-basis (*j* subscript) enabling products to contribute differently. It is important to understand that for different values of z_p different designs will perform better by this equation. This outcome, as Norman and Smith concluded in their work, enables a robustness metric to be established by examining the performance of the design over a designer specified range of percentiles, *p*-values. As Norman and Smith identified, integrating Equation (52) over a range of p values (p_L to p_U) results in the following robustness metric (slightly modified to be on a period-basis):

$$RM_t(\Pi_t) = E_t(\Pi_t)(p_U - p_L) + \frac{s_t(\Pi_t)}{\sqrt{2\pi}}\left[\frac{1}{e^{\frac{\left(\Phi^{-1}(p_L)\right)^2}{2}}} - \frac{1}{e^{\frac{\left(\Phi^{-1}(p_U)\right)^2}{2}}}\right] \tag{59}$$

where p_U is the upper percentile, p_L the lower percentile, $\Phi^{-1}(p_{U/L})$ the inverse cumulative normal function evaluated at p, and $RM_t(\Pi_t)$ the robustness performance metric. Like that of Equation (52), the first term in Equation (59) is representative of the expected net income for the period t as computed before in Equation (10) while the second term is the contribution due to production uncertainty. To encapsulate this robustness metric, the summarizing objective function presented earlier, Equation (9) is revised as follows:

$$RR_t = RR_{t-1} + RM_t(\Pi_t) \tag{60}$$

where the design with the largest RR_t, or retained robustness, come the end of the last period t is then identified as being the design that performs best over the horizon and for the range of p values. In other words, it is deemed the most robust design for the provided conditions.

Now for closure, let's consider some unique cases. In the absence of production uncertainty, the designer has several options in which he can deploy to emulate this provided this outlined implementation. One option is to set the coefficient of variances to zero, this effectively eliminates the uncertainty contribution term, i.e. standard deviation term, which then leads to Equation (60) reducing to the original nominal retained earnings function from before, Equation (9). Another option would be to set $p_L = 0$ and $p_U = 1$. In this scenario, the bracket portion of the second term in Equation (59) reduces to zero thereby effectively cancelling out that term and moreover reducing Equation (59) to just that of the nominal net income computed leveraging Equation (10). Another available option is to set $p_L = p_U = 0.5$. When doing this the developed model automatically leverages the statistical percentile equation, Equation (52), to define the performance metric, albeit no longer a robustness performance metric. In this case the 50% percentile coincides with a z value of $z_{0.5} = 0$, thereby reducing the formulation once more to just that of the original definition of the expected or nominal net income value defined before in Equation (10). Either of these options reduces Equation (60) above to that of Equation (9), which can then be leveraged to sufficiently define the performance of the layout under no production uncertainty. Under this same logic, the designer, by setting the two percentiles equal, can also evaluate the problem at different percentile values if they so choose. Additionally, if a six-sigma evaluation is desired the designer can set $p_L = 0.001$ and $p_U = 0.999$ which emulates the 6σ ranges shown in Figure 39. This concludes the presentation of the statistical robustness method implemented in the developed model to provide robustness to localized production uncertainty.

4.4.2 Constraint Model

Having since established the model developed in this dissertation to determine the performance of layout designs, and the systems they are a part of, attention turns towards the model developed to establish the feasibility of these designs. Every design generated by the algorithms of Stage One and Two may not be feasible when applied in the real-world. The developed constraint model of this dissertation is thus tasked with distinguishing between a feasible and non-feasible design. To achieve this, the model needed to consider a variety of different constraints to handle the unique problem formulation of this dissertation. In general, the developed model includes five distinct constraint groups. These include object overlap avoidance, closed loop avoidance, I/O point accessibility, boundary, and finally budget constraints. These five constraint groups are further decomposed into two constraint types: hard and soft. The five constraint groups, their constraint type assignment, and their applications in the two solution stages, outlined earlier, are provided in Table 9.

Before addressing the individual constraint groups and their applications in each stage, an understanding of the two constraint types must first be established. While the avoidance (overlap and closed loop) and accessibility constraints are characterized as hard constraints, the boundary and budget constraints are not. Instead these constraints were strategically defined as soft constraints, whereby decision logic and penalty functions were implemented to account for the cost a layout design would incur from violating said constraints.

The differentiation between these two types has to do with how absolute the constraint is. In the case of the hard constraints, violation of any degree warrants rejection of the layout design in the algorithms of Stage One and Two. In other words, any design that does not abide by each of first three hard constraint groups is labeled infeasible and discarded by the solution algorithms as a result. For the first three constraint groups this hard distinction is quite logical. For example, in the case of the overlap avoidance constraints, two objects can’t occupy the same space, period, as otherwise this would be a violation of physics. Likewise, in the case of I/O point accessibility and closed loop avoidance constraints, which are unique to the constraint model developed for this dissertation and required as a result of the advanced flow distance method and multiple spacing interaction considered; if the input/output points of the stations cannot be accessed (i.e. reached) by the handlers, then this constitutes a break in the process flow and would thereby inhibit operations. As a result, these three constraint groups had to be established as hard constraints whereby when any one was violated, the layout design would be considered infeasible.

In the case of the soft constraints, the boundary and budget constraints, violation could potentially be permitted. With that said, violation, when allowed, to any degree would require a cost of violation to be incurred by the design. This cost then represents the penalty incurred, or the penalty function, ϕ_t, established earlier in this section when the net income equation of the summarizing objective function of the developed performance model was presented (Equation (10)). As noted, the layout design, despite violating one or both of these constraint groups could still yield a design that is considered feasible. There are specific circumstances that allow these constraints to be soft in nature and this to be true. When such circumstances are not so, these constraints default to being hard like that of the avoidance and accessibility constraints.

The circumstances that lead to these two constraint groups being soft in nature depends on the choices made by the designer/management regarding these constraints when establishing the scenarios in the problem initialization step of the LIVE methodology. Decision logic was implemented to enable designers/management to have control over whether these two constraints should be rendered hard or soft. This approach has a couple noteworthy advantages. The first is it enables strategy level decisions regarding these two soft constraints to be made by the designer/management, which could have a significant impact on the design deemed best by the performance model outlined before.

The second advantage of this approach is it enables all designs of Stage One to be potentially viable, depending on the executive strategy decisions made a priori by the designer/management. In the case where both the boundary and budget constraints, remain soft and the penalty functions thus active, every design yielded by the solution algorithm in Stage One will be viable by the complete set of constraints. This is because the hard constraints (overlap avoidance, closed loop, and I/O point accessibility) are inherently captured by how Stage One was formulated and the sequence-pair mathematical model deployed to represent the layout designs in this stage. As a brief review, this model is a stacking rule-based algorithm, comparable to the game Tetris, whereby the objects are stacked upon each other. This approach, when coupled with this dissertation's application of the walking spacing boundaries as the object's stacking boundaries, inherently ensures that objects cannot be overlapping and moreover that the I/O points will always remain accessible. In other words, all generated designs by the procedures of Stage One will be viable by these three hard constraints. This leaves only the soft constraints that need be satisfied for the design to be deemed feasible. If these soft constraints are in fact soft by design, the problem in Stage One effectively becomes unconstrained, thereby greatly improving the tractability of the problem in Stage One.

Regardless, even in the case where the soft constraints are rendered hard per the designer's choice, the constraint dimensionality in Stage One is greatly reduced to only the latter two constraint groups. Moreover, in Stage One only the top and right boundaries need be considered due once more to the stacking nature of the sequence-pair representation of Stage One. This in lies the major advantage of formulating the constraints in this fashion. The application of the constraint groups for both stages is again provided in Table 9 for reference.

At this point, the question becomes, how were the penalty functions of these soft constraints defined such that the objective function remained cost-based and additionally how were the hard forms of the five constraint groups defined in the developed constraint model? As for the latter question, a presentation of the mathematical implementation and further elaboration on these hard constraints is provided, for reference, in Appendix E. The former question of how the soft constraint penalty functions of the constraint model were defined is the focus of the discussion that follows. Note that all constraints and penalty functions are applied on a period-basis as they are dependent of only the layout configuration (i.e. placement of the objects) which is constant across the span of the period.

Table 9 – Constraint summary table

	Stage I	Stage II
Hard Constraints	✓	Overlap Avoidance
	✓	Closed Loop Avoidance
	✓	I/O Point Accessibility
Soft Constraints	Boundary (R/T)	Boundary (L/R/B/T)
	Budget	Budget

✓ = inherent under modeling formulation

Stage I = QAP/U-SP || Stage II = MIP

4.4.2.1 Penalty Function

The penalty functions implemented for the soft boundary and budget constraints needed to resemble realistic costs in order to align with the cash-based objective function of the performance model. The summarizing cost penalty function is composed of the two constraint group contributions as follows:

$$\phi_t = \phi_{boundary,t} + \phi_{budget,t} \tag{61}$$

where t is the period, $\phi_{boundary,t}$ is the total cost incurred for violating the boundary constraints and $\phi_{budget,t}$ the total cost incurred for violating the mandated budget constraints in period t.

4.4.2.1.1 Boundary Violation

To model the boundary violation penalty function, a layout expansion cost model was implemented. This model characterizes the cost incurred by an object placed outside the OML of the layout as a function of how much the boundaries of the space would need to be expanded to accommodate this placement. This penalty function is as follows:

$$\phi_{boundary,t} = SQAC \cdot \sum_{s} A_s \tag{62}$$

where s are the violating stations, *SQAC* is the square area cost of construction (a property provided by the designer in the first step of the LIVE methodology), and A_s is the square area expansion of the OML required to encapsulate the station s. *SQAC* is a relatively easy property to define based on a survey of local contractors and the market rate. A default value of 25 dollars per square foot is a good average starting cost. This penalty approach enables designers/management to consider the cost of expanding their existing facility to encapsulate what could be a more advantageous configuration of the objects. Sometimes though, such expansion is not an option (e.g. buildings adjacent to the facility) and when this is the case the boundary constraints become hard and resemble those provided in Appendix E**.**

In implementing this approach, the boundary constraints for the constrained objects had to remain hard regardless of the designer's choice on how to prescribe these constraints. This was done to restrict designs such that the constrained objects remained in their assigned locations. By not doing so, it was observed that the algorithm ran the risk of identifying optimal and viable designs whereby the fixed objects were not placed in their required positions. This was a result of the above outlined soft constraint penalty function, although penalizing the misplacement of the constrained objects, not penalizing severely enough to render the design suboptimal. To combat this, the constrained object's boundary constraints are mandated as always being hard (i.e. absolute).

Because of this implemented penalty function, modification to the advanced material handling distance algorithm was required to account for the potential relaxed nature of the boundary constraints. This was to enable feasible paths to still be searched for and found by the algorithm even when objects and their I/O points fell outside the original OML boundaries. The modification encompassed a design switch leveraging the OML boundaries to be observed by the algorithm as being pass-through in nature and further retaining consideration by the advanced algorithm even while not viable by the hard boundaries.

4.4.2.1.2 Budget Violation

Now to model the budget constraint penalty function, a financial business strategy model is leveraged. Implementing a penalty function to capture violations of the budget constraints is beneficial as such constraints do not necessarily have to be absolute in practice. These are more so strategic guidelines. In other words, operational changes such as a layout rearrangement are not purely contingent on the budgetary restrictions set forth initially. In theory, management could decide to make up any difference through financing. In the business accounting space, this is what is called debt financing, or in other words, financing through loans or the issuance of bonds to gain the necessary capital to enact such a strategic plan (e.g. layout restructuring). This financing comes at a cost however and it is this cost or cost of debt that becomes part of the model deployed in this dissertation to define the budget constraint penalty function.

In the developed model there is also an added option to leverage retained earnings when the budget, based on a percentage of the net income, is not enough alone. Since net income contributes to retained earnings, this option can be viewed as an extension of the budget. This is because whether it is retained earnings used directly or a larger percentage of the net income used, in the end the latter will effectively reduce the retained earnings by the same amount if the former was deployed. Now these two strategic options available to the designer/management to finance through debt or leverage retained earnings (i.e. extend the budget) can be jointly deployed or deployed individually. Moreover, neither can be deployed, which results in the budget constraints becoming hard and resembling those provided in Appendix E**.** When this is not the case and one of the options is deployed by the designer/management when initializing the problem, the remaining debt to be financed after applying the budget is computed as follows:

$$DEBT_t = \begin{cases} RCC_t + \phi_{budget,t} - p_b \cdot Net\ Income_{t-1}, & Net\ Income_{t-1} > 0 \\ RCC_t + \phi_{budget,t}, & Net\ Income_{t-1} \leq 0 \end{cases} \tag{63}$$

where RCC_t are only the capital costs associated with rearrangement (i.e. neglecting loss of production discussed in the last section when the cost function was presented), $\phi_{budget,t}$ is the boundary violation penalty cost defined before, p_b is the budget as a percentage of the previous period's $Net\ Income_{t-1}$ as established by the performance model outlined earlier. Note, net income can be negative (e.g. when the firm loses money), which is why when such is the case the budget is not subtracted from the capital costs. Additionally, the budget constraint penalty must also consider the cost of violating the boundary constraint ($\phi_{budget,t}$) so the two are order dependent and the boundary constraint cost penalty function outlined before must first be calculated. With the debt associated with the rearrangement known, the penalty function associated with each unique combination of the two strategic decision options available can be defined.

When debt financing is an option (as defined by the designer/management during the problem initialization), but leveraging retained earnings is not, the budget constraint penalty function becomes the following:

$$\phi_{budget,t} = \begin{cases} IR \cdot DEBT_t, & DEBT_t > 0 \\ 0, & DEBT_t \leq 0 \end{cases} \tag{64}$$

where IR represents the firms borrowing interest rate, a property easily established by a firm and thus defined during the problem initialization step of the LIVE methodology. If the $DEBT_t$ is less than or equal to zero, there is no budget constraint penalty as the budget alone can sustain the costs of the rearrangement. Now when such is not the case, the penalty becomes the cost of debt associated with financing the difference between what the budget can sustain and what is required. Moreover, the capital costs of rearrangement and the boundary penalty themselves are not included in this penalty as they are already accounted for in the costs each period.

Now when leveraging retained earnings and debt financing, the rearrangement is first financed through the retained earnings only to then be debt financed if required. The budget constraint penalty function in this case becomes:

$$\phi_{budget,t} = \begin{cases} 0, & RE_{t-1} \geq DEBT_t \\ IR \cdot (DEBT_t - RE_{t-1}), & RE_{t-1} < DEBT_t \end{cases} \tag{65}$$

where RE_{t-1} is the retained earnings after any budget has been applied. This is because the budget, as demonstrated before, is based on the net income which contributes to the retained earnings. Not subtracting said budget from the retained earnings first would make it appear that more funds are available when there are not. Now when the retained earnings are enough to fund the remaining debt after the budget is applied, then the cost penalty function is zero as the cost function outlined before accounts for these costs already. When retained earnings are not enough, the remaining debt after the available retained earnings are applied is then debt financed as demonstrated in the second condition in Equation (65). In the case where $RE_{t-1} < 0$, it is set to zero in the above equation and all debt must be financed.

Now in the scenario where retained earnings is an option, but debt financing is not; if the debt is greater than zero and retained earnings are not enough to cover the debt, then the rearrangement cannot be supported. In this case the budget constraint effectively becomes hard whereby the design is then deemed infeasible. In effect, the costs associated with rearrangement and the new layout cannot be absorbed by the business.

The decision by the designer/management regarding which of these options to deploy can be fundamental to the business' success going forth. For example, the decision to leverage debt financing impacts the firm's debt to equity ratio. For a publicly traded company, this can influence investor's perception of the firm and thus indirectly can impact the firm's access to equity. In most cases, the firm's business strategy will dictate the approach deployed (i.e. how these options are established for each scenario during the problem initialization). With that said the ability to assess how different business strategies can impact which layout design is best to implement is valuable insight. Such a capability, enabled by the developed constraint model, creates substantial value for designers and management. At this point, the developed performance and constraint models of this dissertation have been thoroughly presented. Before a summary of the implementation of this dissertation is provided, a brief recap of the performance and constraint models is provided.

4.4.3 Summary of the Developed Performance and Constraint Models

The developed performance and constraint models of this dissertation, outlined above, establish the performance and feasibility of each layout design, and the system it is a part of. The performance model incorporates a cash-based objective function, which aids in bridging the language chasm that often exists between stakeholders involved in the layout design process. By bridging this gap, improved collaboration can occur. The performance model also consists of a comprehensive analytical model that considers several costs associated with the system's operations and the restructuring of the layout. An advanced flow distance method that guarantees flow path feasibility was implemented to provide closure to a major research gap observed in this dissertation. The inclusion of this method provides improved material handling cost estimations that are better representative of the costs likely to be experienced in practice. A process flow analysis, leveraging the concept of utilization, was implemented to ensure production rates remain achievable relative to the system's capacity. A localized robustness method was also implemented to ensure robustness relative to production rate uncertainty.

The developed constraint model considers several constraints that ensure the layout design's real-life feasibility. Strategically defined penalty functions were incorporated to account for boundary and budget constraint violations when applicable. The developed method of deploying these penalty functions enables varying business-strategies to be evaluated by the LIVE methodology. The combination of the performance and constraint models provide improved insight into the performance and feasibility of designs. Supplementary data on the operations is accessible to the designer and management, which it is believed will facilitate collaboration and enable more informed decisions to be made regarding the design of layouts subject to unpredictable and evolving conditions.

4.5 Implementation Summary

The LIVE methodology is composed of three steps: problem initialization, whereby the scenario layout problems are established, problem solution, whereby each of the scenario's layout problems are solved and the best design identified, and finally analysis, whereby the designs are evaluated for performance and feasibility which ensures that the best performing, and feasible design is identified during the solution step. In step one, the designer and management establish a series of scenarios that range the design space they desire to investigate. This design space encompasses changing conditions as well as differing business strategies. For each of the scenarios the conditions and business decisions the layout design will be subject to are established by the designer and management.

Once established, the LIVE methodology proceeds into solving each of these scenario layout problems using a novel bi-model multi-stage hybrid solution approach. This approach consists of two stages, both of which deploy genetic algorithm-based optimization techniques. In Stage One a QAP/SP-U model is deployed to represent the layout. A novel feasible sequence-pair promoting method (FSPPM) was developed to promote the discovery of feasible sequence-pair designs during the optimization process. Additionally, novel methods were developed for many of the various genetic operators (e.g. a novel genome repair process) as well as for the perturbation method deployed by the FSA technique. In Stage Two a MINLP model is deployed to represent the detailed layout. The results of Stage One are leveraged to initialize a tri-population scheme and like that of Stage One, novel methods were developed for many of the genetic operators, specifically that of the jumping gene operator. Collectively, these efforts contribute greatly to advancing both that of the optimization portion of the layout problem literature, but also that of the mathematical programming literature. The mathematical programming techniques developed here apply beyond the scope of this problem application and universally advance solution to similarly structured combinatorial optimization problems.

In the last step of the LIVE methodology, the performance and feasibility of a design is evaluated using a comprehensive cash-based performance model. This model deploys an analytical cost model to characterize the system costs. In computing these costs a novel advanced flow distance method was developed, which provides closure to another major research gap of this dissertation. A localized robustness method was also implemented to provide design performance robustness relative to production uncertainty. Rounding out the implementation, a constraint model which considers several constraints was developed to ensure real-life design feasibility. A flexible penalty function was implemented to consider boundary and budget violations thereby enabling designers and management to consider different business strategies when establishing the scenarios in step one of the methodology. With that, a very brief summary of the LIVE methodology has been provided.

Before examining how the outlined implementation fairs in providing closure to the key research gaps identified and moreover substantiation of the hypotheses proposed before, the next chapter provides a reminder of salient observations from the literature review, the motivating research objective, the overarching hypothesis, and other notable hypotheses formed in this dissertation. Furthermore, the experimental approach deployed to test each hypothesis will be outlined before then presenting the results of the experiments.

CHAPTER 5

RECAPITULATION AND PROPOSED EXPERIMENTS

The goal of this chapter is to restate the research objective, the overarching hypothesis of this dissertation, and the secondary hypotheses formed as a result of major research gaps identified and insightful observations made up until this point. In addition to this recapitulation, the proposed experimental approach to substantiating these hypotheses is presented whereby a brief overview of each experiment in this approach is provided. Now as a reminder, the motivating research objective of this work was as follows:

> **Research Objective:** To establish an improved and robust methodology for exploring the design space of a detailed evolving manufacturing layout, enabling more informed and collaborative design decisions to be made under evolving and uncertain market and business model conditions.

This research objective was established after a comprehensive survey of the literature was performed whereby several key research gaps were identified. The first of these major research gaps pertained to the absence of a method to account for flow path feasibility when considering the material handling costs; the standard layout evaluation metric in the literature. Conventionally, rectilinear or Euclidean distance methods, which do not account for flow path feasibility, are deployed in the literature to establish these costs. Preliminary results had however demonstrated the importance of this flow path feasibility consideration when establishing the material handling costs. **Observation 3** identified that *failure to account for such flow path feasibility can result in the* *identification of suboptimal layout designs*. Given the significant contribution of these costs to a system's total production costs and therefore performance, closure of this gap became a requirement of this dissertation. **Hypothesis 1**, the first secondary hypothesis of this dissertation, was formed as a by-product of this observation and research gap:

> **Hypothesis 1:** **<u>If</u>** an advanced flow distance method that ensures flow feasibility is implemented to define the MHCs, **<u>then</u>** improved layout designs that are better representative of reality can be established for variable production environments where several interrelated processes are occurring concurrently.

In addition to the aforementioned research gap, another growing gap was observed as the problem formulation required to accurately define a layout was formed through numerous assertions and observations cited during the literature review. Through these observations and assertions, specifically the synthesis of **Observation 2** and **Assertion 11,** it was identified that t*o effectively design a layout subject to evolving and uncertain market conditions and business models, the problem must be structured as a budget constrained stochastic robust dynamic layout problem (RDLP) modeled under a continuous detailed mixed integer programming (MIP) approach*. As a reminder, robust refers to a layout being robust to fluctuations in the conditions and dynamic refers to a layout that evolves over time (i.e. is rearranged periodically over the planning horizon). Now in reaching this conclusion, it was further identified that such a unique formulation has never been pursued in the literature, which forms the second major research gap. As postulated before, much of this gap can be attributed to the difficult nature of such a problem formulation.

In relation to **Hypothesis 1** it was further observed that such an advanced flow distance method would require solution to a NP-hard difficult problem (a variant of the traveling salesman problem) for each unique flow connection in the system. As such, the inclusion of such a method only further contributes to the difficulty of the problem formulation of this dissertation. Observation of this led then to **Assertion 12** stating that *to handle the difficult and time intensive nature of such a problem formulation, identifying an efficient solution method was imperative*.

After exploring the literature pertaining to the solution of such a problem, it was then identified that to most effectively solve such a difficult and complex problem formulation that a bi-model multi-stage hybrid solution approach should be leveraged to accomplish this. **Assertion 17** was established, in response to **Question 1.1.1,** stating that *a simplified QAP/U-SP model of the problem formulation could be solved initially to provide partial solution to the overarching problem and better initialize the populations of a multi-population hybrid GA algorithm that ought to then be leveraged to provide solution to the MIP formulated version of the RDLP.* This assertion and the observations that led to it provided the basis for the structure of the then proposed bi-model multi-stage solution approach developed in this dissertation. Later while formulating the LIVE methodology proposed in this dissertation, it was further established that a synthesis of identified literature best techniques and most applicable models, modified to encompass the unique nature of the problem formulation of this dissertation, be implemented to form the foundations of the models and solution techniques leveraged in this proposed bi-model multi-stage solution approach to solving the MIP formulated RDLPs. As was outlined, Stage One of this approach leverages the more tractable QAP/U-SP model to generate solutions to then initialize Stage Two through the partial solution of the RDLP using a hybrid GA algorithm deploying FSA to enhance solution quality. With the results of Stage One forming the initial tri-populations of the GA algorithm of Stage Two, the more complex and difficult to solve MIP model is then solved to completion. Considering that observed in the literature and the proposition of this approach, **Hypothesis 2**, the second secondary hypothesis of this dissertation, was formed:

> **Hypothesis 2: <u>If</u>** the proposed bi-model multi-stage hybrid solution approach is implemented to solve the MIP formulated RDLP, **<u>then</u>** the problem will be solved most effectively, in terms of solution quality.

Now as noted earlier, the core objective of this research was *to establish an improved methodology that could enable more informed and collaborative layout design decisions to be made in the presence of evolving and uncertain market and business model conditions*. In light of observations made while reviewing the literature, such a methodology, which enables more informed layout design decisions to be made by providing adequate transparency into how the layout design performs in relation to the evolving and uncertain nature of the conditions, was not available. This in turn motivated the development of the proposed LIVE methodology of this dissertation.

Consisting of three steps, the LIVE methodology attempts to enable more informed layout design decisions to be made by the stakeholders involved in the process (designer, management, etc.). To achieve this the LIVE methodology uniquely handles the evolution of the market and business model conditions external to the solution procedure thereby enabling the designer to effectively observe how different condition forecasts impact the design of the layout and system. A series of RDLP scenarios, encompassing the various condition forecast combinations of interest, are formed by the designer. The bi-model multi-stage solution approach proposed before is leveraged to provide effective solution to each of these RDLP scenarios. A localized robustness method is infused to provide designs that remain robust to localized production uncertainty. To enable more collaboration amongst the stakeholders involved in layout design process, an analytical performance model was proposed to bridge the language chasm often present between the stakeholders involved in such a process. The proposition of this methodology then led to the overarching hypothesis of this dissertation:

> **Overarching Hypothesis: <u>If</u>** the problem of designing an environment subject to evolving and uncertain market and business model conditions is solved with the proposed LIVE methodology, **<u>then</u>** designers will be capable of making more informed and collaborative decisions on its design.

The Overarching Hypothesis is dependent on both Hypothesis 1 and 2. As such, if Hypothesis 1 and 2 are substantiated through experimentation, then the overarching hypothesis has the potential to be proven true. Substantiation of Hypothesis 1 proves that more optimal designs can be discovered when considering flow path feasibility while substantiation of Hypothesis 2 proves that the unique and difficult RDLP formulation of this dissertation can be effectively solved. Effective solution is a key component of the LIVE methodology. As such, proof of this acknowledges that the methodology can provide a designer with a medium to effectively design a layout subject to evolving and uncertain conditions. Complete substantiation of the Overarching Hypothesis will require further experimentation though. Substantiation of this hypothesis can only be achieved by application to a real-world problem whereby the methodology can be assessed for how capable it is at enabling designers to make more informed and collaborative decision regarding the design of a layout subject to evolving and uncertain conditions.

5.1 Experimental Approach

The experimental approach to substantiating the above restated hypotheses is composed of three distinct experiment sets. The experiments of these sets build upon each culminating in the final experiment, which applies the LIVE methodology to a real-world case study to observe the effectiveness of the methodology.

A representative *52 Problem Test Set* is constructed and leveraged in Experiment Set A and B to substantiate Hypothesis 1 and 2. In Experiment Set A, the developed FSPPM, advanced flow distance method, and FSA integration in Stage One of the bi-model multi-stage solution approach are analysed. Comparison of the first two to the literature baseline methods is examined while in the latter the necessity of integrating FSA in Stage One is considered. The outcomes of Experiment Set A are then leveraged in Experiment Set B. In Experiment Set B, the effectiveness of the proposed bi-model multi-stage solution approach is evaluated. Optimization parameter studies are performed to identify the best configuration of optimization parameters to deploy. The results of these studies are then leveraged by the last experiment. In Experiment Set C, the overarching hypothesis of this dissertation is substantiated by applying the LIVE methodology to a real-world layout design problem.

5.1.1 Experiment Set A: Validation of Methods

Purpose: Test the effectiveness of the FSPPM in promoting the discovery of feasible designs. Test the advanced flow distance methods ability to ensure flow path feasibility and discover optimal designs that are better representative of reality. Test the need to infuse FSA into the first stage of the bi-model multi-stage solution approach.

To perform these tests, Experiment Set A is decomposed into three distinct experiments; Experiments 1, 2, and 3.

In Experiment 1, a modified version of the *52 Problem Test Set* is leveraged to examine the FSPPM's ability to more frequently establish feasible designs. Before examining the performance of the method across the modified *52 Problem Test Set*, validation of the assumptions implemented to construct the FSPPM is first established. This is to be achieved by examining the number of designs that would need to be generated before 100 feasible designs are discovered for a standard problem. For the modified *52 Problem Test Set*, it will be shown that deployment of the FSPPM provides significant CPU time savings by identifying feasible designs more frequently and moreover enables problem sizes that would otherwise be unsolvable to become then solvable.

In Experiment 2, the advanced flow distance methods ability to ensure flow path feasibility and discover optimal designs that are better representative of reality is tested across the *52 Problem Test Set*. It will be shown that designs generated in Stage One leveraging the advanced flow distance method to determine the material handling costs are notably different from those generated when the literature standard rectilinear method is deployed. It will also be shown that it is of the upmost importance to consider flow path feasibility when optimizing the final layout design. In demonstrating this importance Hypothesis 1 will be substantiated by proving that the advanced flow distance method enables designs that are better representative of reality and therefore more effective in practice to be identified. Additionally, a comparison of the Stage One CPU times using both methods versus the populated unique design set to pass to Stage Two will also be examined. From this examination it will be shown that although the two methods differ from an optimality perspective, the CPU savings and similar diversity characteristics of the unique design set associated with using the traditional rectilinear method makes its use in Stage One a viable strategy.

In Experiment 3, the necessity of infusing the developed FSA technique in Stage One of the proposed bi-model multi-stage solution approach is tested, once more leveraging the *52 Problem Test Set* problems to do so. In this experiment it will be shown that though the FSA technique better ensures optimality is achieved, the additional time required to provide solution does not warrant its inclusion in Stage One when acting in initializing the Stage Two algorithm. When acting in providing final solution to the problem however, its implementation is recommended.

5.1.2 Experiment Set B: Optimization Performance Study

Purpose: Test the effectiveness of the Stage One and Two solution procedures of the proposed bi-model multi-stage solution approach to solving the complex layout formulation of this dissertation. Test different optimization parameter combinations to *identify the appropriate parameter sets to deploy to most effectively solve said layout problems*

To perform these tests, Experiment Set B is decomposed into two experiments; Experiment 4 and 5. The two experiments are split between the two stages of the proposed solution approach.

In Experiment 4, the developed Stage One solution procedures are tested, and the best optimization parameter sets identified. The solution procedures of Stage One are tested for the *52 Problem Test Set*. As will be shown, depending on the end goal of the Stage One algorithm (optimality vs. Stage Two initialization), a different parameter set ought to be deployed. Moreover, it will also be shown that these best parameter sets differ depending on the problem type being solved. That is whether solving a static or dynamic problem. Much like Experiment 4, in Experiment 5, the proposed Stage Two solution procedures are tested, and the optimization parameter set that best provides optimality identified. As will be shown, once more the best parameter set to deploy to provide optimality will depend on the problem type being solved. The synthesis of these two experiments provides substantiation to Hypothesis 2.

5.1.3 Experiment Set C: Real World Case Study

Purpose: To test the LIVE methodology by applying it to a real-world layout design problem and to test its ability to enable designers and stakeholders to make more informed and collaborative decisions.

To perform these tests, the final experiment, Experiment 6, examines the operations of an aerospace parts warehouse, the effectiveness of the current layout configuration, and the redesign of it. The study performed, examines the performance over a forecasted three-year period and for a multitude of different changes in the market conditions and business model. Robustness relative to production uncertainty is examined during the process of identifying the best redesign which will maximize profit over the three-year planning horizon. As will ultimately be shown in performing this study, the LIVE methodology enables more informed decisions to be made regarding the design of the layout and the operations of the system. Moreover, the integration of the detailed performance model will demonstrate its usefulness in enabling such decisions to be made and additionally providing expanded insight into the layout design process.

CHAPTER 6

EXPERIMENT SET A: VALIDATION OF METHODS

The goal of this chapter is to present the results of the Experiment Set A. This set consists of three distinct experiments as outlined before. As a reminder, the purpose of this experiment set is to test the following:

Purpose: Test the effectiveness of the FSPPM in promoting the discovery of feasible designs. Test the advanced flow distance methods ability to ensure flow path feasibility and discover optimal designs that are better representative of reality. Test the need to infuse FSA into the first stage of the bi-model multi-stage solution approach.

6.1 Experiment 1: FSPPM Validation and Testing

In Experiment 1, the novel Feasible Sequence-Pair Promoting Method (FSPPM) developed to improve the discovery of feasible sequence-pairs is tested. The methods ability to more frequently establish feasible designs is examined along with the assumptions implemented to construct it. This experiment has three parts. In the first, the FSPPMs construction and assumptions are validated. In the second, the FSPPMs ability to more frequently establish feasible designs relative to the literature standard purely random assignment method is examined. In the third, a *52 Problem Test Set* is solved provided several different options including one in which the FSPPM method is not deployed, replaced instead by a purely random method that is most often leveraged in the literature to establish sequence-pairs. Before examining the results of this *52 Problem Test Set* solution, the FSPPM fundamental construction is first justified.

6.1.1 Experiment 1.A: FSPPM Validation

The goal of Experiment 1.A was to validate the assumptions and construction of the developed FSPPM.

6.1.1.1 Apparatus Setup

To test the fundamental construction of the novel FSPPM, a problem consisting of two constrained objects in the physical space was leveraged. These two constrained objects are two of a hundred objects present in the space. The physical boundaries of the space were defined as encompassing a square space of 105 x 105 unit-distance in size. The dimensions of the objects were randomly generated to range between 2 to 7 in length and width. The constrained objects were specified as having a size of 5 x 4 and 4 x 4 respectively, and both non-rotated. The 5 x 4 object, *Object A*, was placed in the absolute bottom-left corner of the space (xy = 2.5, 2) while the 4 x 4 object, *Object B*, was placed near the top-right corner of the space (xy = 93, 103).

With this physical setup in place, a hundred feasible sequence-pairs were generated leveraging a purely random assignment method identical to the standard method deployed in the literature. In other words, each object is placed randomly in the sequence-pair apart from the last one to be placed, which by default must be placed in the only remaining unassigned sequence-pair position. After generating these hundred feasible sequence-pairs, the resulting negative and positive sequences of them were processed and the placement distributions (mean and standard deviation) of the two constrained objects in both these sequences were established. Additionally, the FSPPM was deployed and the bisecting diagonal distance (BDD) property of each of these constrained objects was determined. This BDD property directly establishes the most likely position in the sequence-pair the FSPPM would place the object (i.e. the mean position). The BDD property is equivalent to the right-hand side of Equation (5) noted before when the construction of the FSPPM was first outlined. The results of this test apparatus are now presented.

6.1.1.2 Testing Results and Analysis

Figure 40 provides the resulting sequence placement distributions of Experiment 1.A in a histogram format for the constrained objects, *Objects A* and *B*, in both sequences of the sequence-pair. The blue bars indicate the frequency of placement in the sequence position at the left edge of the bar and as defined on the x-axis. The red curves represent fitted normal distributions to the data. As observed, in the positive sequence, both *Object A* and *Object B* fall relatively close to the middle of the sequence. This mirrors expectations given their positions in the physical pace. This result is expected because in the physical space they fall close to the positive bisecting diagonal line. Object A falls exactly on the diagonal while Object B, is just to the left of it. Provided observations made regarding the placement of objects in the physical space relative to their placements in the sequences and this bisecting line (Appendix B), this result aligns well with such observations thereby confirming expectations. In the negative sequence, Object A, as demonstrated, absolutely falls in the first position of the sequence with zero deviation from this position. Object B on the other hand falls near the end of the sequence and centred about the 96th position in the sequence. Like before, this mirrors that in which is expected. Apart from Object A's placement in the negative sequence, which is a special circumstance, the distributions have standard deviation in the range of roughly 0.7 to 0.9.

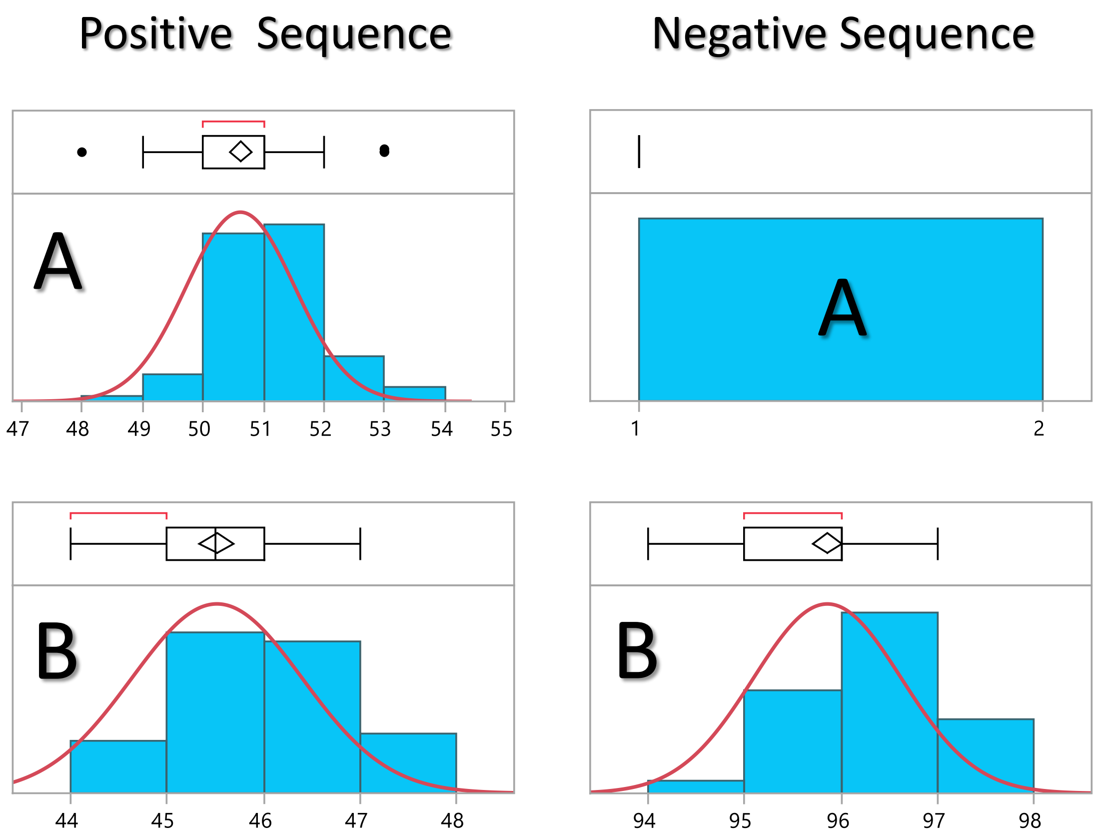


Figure 40 – Sequence placement distributions of Experiment 1.A

With the BDDs of each sequence defining the mean placement positions of the objects in the FSPPM, by directly comparing it to the mean of the distribution created by the hundred randomly generated sequence results, the method deployed by the FSPPM to define the mean of the placement distributions can be validated. In theory if the BDD of the FSPPM matches or is reasonably close to that of the mean of the randomly generated distributions, then the developed method to defining the mean of the constrained object distributions according to this BDD property is confirmed. As is demonstrated in Table 10, the computed BDD, or mean, by the FSPPM nearly identically matches the mean of the randomly generated distribution result. In fact, if one computes the difference between these, they would find that the BDD method of defining the mean, which is deployed by the novel FSPPM, is on average within 0.3% of the true expected value in this example.

Table 10 – Placement distribution statistics

Statistic	Positive Sequence		Negative Sequence	
	Object A	Object B	Object A	Object B
FSPPM's BDD	*50.75*	*45.60*	*1*	*95.10*
Mean (μ)	*50.62*	*45.52*	*1*	*95.86*
Sigma (σ)	*0.90*	*0.88*	*0*	*0.77*

6.1.1.3 Key Insights and Conclusions

Considering these results several key insights can be established:

1) The bisecting diagonal distance (BDD) property, or mean, derived by the FSPPM mirrors that of the true expected value of placement

2) The distributions about the expected value are normal

3) The standard deviation falls in the proximity of 0.7-0.9

4) Objects at the extreme corners always fall at the end of the respective sequence

These takeaways in turn enable several conclusions to be made regarding the implemented assumptions and approaches of the FSPPM. First, it can be concluded that the FSPPM derived BDD property for defining the mean placement position aligns extremely well with the true distribution mean. Second, the assumption implemented in the FSPPM to define the placement distributions normally was confirmed by observation of the random results forming normal distributions about the expected value. Moreover, recall that in the developed FSPPM, a secondary rule was also implemented in response to a placement observation relating to takeaway four above. In the developed FSPPM, if a constrained object were to fall at an extreme corner, the object without uncertainty would fall within the appropriate sequence at the end of the sequence. In the deployed method, the standard deviation was overwritten to be zero and the object's expected value assigned to be this corresponding end position in the sequence. This was proven not only to be a justifiable rule, but also a perfect mandate after observation of Object A's placement in the negative sequence. Having been placed at the absolute bottom-left corner of the space, it was always to then fall at the beginning of the negative sequence. As demonstrated, thanks to this implemented rule, the BDD perfectly matches that of the true mean and moreover, the overwritten sigma value of zero.

The results and conclusions presented before enable it to then be concluded that the assumptions and general construction of the FSPPM are reasonable and provide accurate depictions of the true placement distributions of constrained objects in the physical space.

6.1.2 Experiment 1.B: FSPPM vs. Random Sampling Method Comparison

The goal of Experiment 1.B was to examine and moreover confirm the FSPPMs ability to more frequently establish feasible designs relative to a purely random assignment method often implemented in the literature.

6.1.2.1 Apparatus Setup

To test the novel FSPPMs ability to more frequently discover feasible designs in the presence of constrained, i.e. fixed, objects in the space, four distinct problem setups of varying characteristics and dimensionality were leveraged. These four problems can be decomposed into two setups relating to the total number of objects in the space. In the first problem setup, a problem size of six total objects is the focus. Two distinct problems compose this setup; one where there are four unconstrained and two constrained objects and the other where there are three unconstrained and three constrained objects. In the second setup, a larger problem size of 12 total objects is the focus. This setup consists of two problems; one consisting of ten unconstrained and two constrained objects, and the other of eight unconstrained and four constrained. In addition to these four problems, the 4/2 and 10/2 (unconstrained/constrained) problems are considered under two different boundary conditions. These setups were strategically defined such that the FSPPMs ability to more effectively generate feasible designs across different problem sizes and number of constrained objects could be assessed. Moreover, consideration of different boundary conditions enables the FSPPMs relative capability to be examined for different layout white spaces. Table 11 provides a summary of these problems.

Table 11 – Problem characteristics of Experiment 1.B.

Setup	Problem	Unconstrained Objects	Constrained Objects	White space
Setup I (6 objects)	*A*	*4*	*2*	*46%*
	B	*3*	*3*	*46%*
	C	*4*	*2*	*19%*

Table 11 (continued)

Setup II (12 objects)	*D*	*10*	*2*	*46%*
	E	*8*	*4*	*46%*
	F	*10*	*2*	*26%*

For each of the six unique problems, a hundred feasible designs are generated using both the FSPPM and the purely random assignment method of the literature. In identifying these hundred feasible designs, the total number of generated designs required to do so is recorded. For each problem, five replications are performed leveraging each sampling technique (FSPPM vs. random) in order to establish an average number of total designs required before a hundred feasible ones are identified. The results of this testing are outlined next.

6.1.2.2 Testing Results and Analysis

The results for the testing apparatus and procedure are provided in Table 12 below. The values in the table are for the average number of designs generated by the sampling technique before identifying a hundred feasible designs (orientation and sequence pairs) across five replications of the sampling techniques. In addition to these values, a reduction factor and the percentage of samples required by the FSPPM relative to the random method are provided. Asterisks in the table represent results that needed to be extrapolated due to the inability of the random method to discover feasible designs in any sort of a reasonable duration of time. For these, the number of generated designs required to find a single feasible one was extrapolated for each of the five replications, then the average taken. In these instances, it took on the order of half a day to find just a single feasible design, which is what necessitated the need for extrapolation to remain capable of comparing the two sampling methods effectiveness relative to one another.

Table 12 – Sampling results of Experiment 1.B.

Layout Properties	**A**	**B**	**C**
Random Sampling	3,485,839	25,732	143,641
FSPPM Sampling	54,530	978	635
Reduction Factor	63.9	26.3	226.2
(Samples Required)	1.6%	3.8%	0.44%
Layout Properties	**D**	**E**	**F**
Random Sampling	*215,020,300	1,084,027	*314,258,200
FSPPM Sampling	1,820,519	10,444	8,147
Reduction Factor	118.1	103.8	38573
(Samples Required)	0.85%	0.96%	0.00%

The overarching observation from these sampling results is that the FSPPM outperforms the conventional random assignment method across the board, by identifying a hundred feasible designs in far fewer required samples (i.e. generated designs). This is easily identifiable by observation of the reduction factors and percentage of samples required by the FSPPM in Table 12. In each of these, the reduction factor is far greater than one and the percentage well below 5%. The provided results also demonstrate improvement relative to several different problem characteristics, whereby improvement is defined as the FSPPMs ability to discover feasible designs at a faster rate than that of the random method.

From the results, it can be identified that the FSPPMs improvement over the random method is proportional to the total number of objects in the space. As the total number increases, the FSPPMs advantage over a random method grows. Observation of either columns two, three, or four independently in Table 12 demonstrates this relative proportional improvement. Looking specifically at column three (results for Problems B and E) one can observe that while the number of fixed objects remains the same, and so too does the white space, the reduction factor, and thus improvement increases from a factor of 26.3 to 103.8, a nearly four times improvement over the random sampling method. The reason for why this is as follows. With more objects present, the sequences in the sequence-pair become longer resulting in there being more potential design permutations. At the same time, only a select number of these would result in the constrained objects being placed appropriately in the sequences such that a feasible design is yielded. This is like the needle in a haystack analogy whereby the haystack has grown making it only that much more difficult to find the very few needles. As such, the likelihood of by chance placing the constrained objects in the correct position in the sequence diminishes. This is why, as the total number of objects increases, it takes considerably more samples for the random method to identify a hundred feasible designs when compared to the FSPPM, which is far less impacted by this occurrence. In the six-object case, there are far fewer permutations and thus the gap between the two methods is reduced, yet still noticeable.

It can also be observed that the relative improvement is inversely proportional to the white space. In other words, as the white space available to place objects decreases, the improvement of the FSPPM over the random method increases. This relationship is demonstrated by observation of columns two and three in setup one or setup two. While both noticeably have more difficulty finding feasible designs, due to the limited white space constricting the combinations of sequence placement and orientation that would allow all objects to then fall within the bounds of the space, the random method struggles noticeably more. In fact, as observed by the results in setup two, the random method results had to be extrapolated due to the method taking excessively long to produce even a single feasible design. This in lies an instrumental advantage of the FSPPM over the standard random method. While the random method could barely identify a single feasible design, the FSPPM remained capable of identifying a hundred such designs in a very respectable duration of time and number of required samples (1.8 million in the most constraining case, Problem D). The consequence of this observed outcome will be further explored as the next part of the experiment, Experiment 1.C, is presented. The implication of this will be explicitly established in this part of the experiment. For now though, it is important to merely understand the relationship between the white space and the ability of the two methods to a) identify feasible designs and b) the FSPPMs ability to noticeable outperform the random method.

Another notable observation of these results is as follows. The relative improvement of the FSPPM over that of the random method is proportional to the number of constrained objects. In effect, as the number of constrained objects increases, and all else remaining constant, the gap in performance between the FSPPM and random method increases whereby the FSPPM is far superior in identifying feasible designs. This can be observed by viewing columns three and four for either the setup one or two problems. As demonstrated, as the number of constrained objects increases from two to three and two to four, the reduction factor increases by an order of magnitude or significantly more in the case of the setup two problems. Note, like before, the random method required extrapolation to provide comparison. The method again struggled to identify feasible designs. This can be understood as being a by-product of having to appropriately place a greater number of constrained objects in the space. If even one is to be placed errantly, the design is deemed infeasible. As such, there is a greater likelihood for the random method to fail in establishing a feasible solution as the number of constrained objects increases. What is more interesting is while the random method becomes significantly less effective as the number of constrained objects increases, the FSPPM does the complete opposite, in effect identifying the hundred feasible designs in fewer required samples. The reason for this is as follows.

As the number of constrained objects increases, the dimensionality of the problem effectively decreases. Recall, the FSPPM handles the placement of these constrained objects in the sequence-pair before then randomly inserting the remaining unconstrained objects. As such, it is these constrained objects that must be strategically placed in the space such that when the stacking-based QAP/U-SP model maps the sequence-pair to the physical placements in the space, these objects fall at their required placement positions. With the FSPPM, by its construction, having a high likelihood of appropriately placing them such that a feasible design is generated, the more constrained objects there are, the fewer unconstrained objects there are that then need to be placed accordingly by the method to then ensure that a feasible design is completely generated. Though not an entirely shocking observation, it remains a noteworthy one provided its implication, which will be discussed later.

Now among these three observations, the impact that each have relative to one another on the improvement of the FSPPM over the random method of the literature is as follows. The increase in the number of constrained objects effectively produces the most significant improvement. Much of this is likely the result of the two methods behaving inversely. While the FSPPM becomes more effective, reducing the number of required designs, the random method does the contrary, requiring not only more, but a significantly greater number of required designs. After the number of constrained objects, the total number of objects has the next greatest impact on the improvement gap between the two methods. This is likely the result of the exponential increase in possible permutations of the sequences. Finally, although having the least impact of the three, the white space available produces a noticeably growing gap in performance between the two methods as the white space decreases.

6.1.2.3 Key Insights and Conclusions

Considering these results and observations, several summarizing insights can then be established:

1) The FSPPM consistently outperforms the traditional random assignment method of the literature

2) The performance gap between the FSPPM and random method grows as the number of objects increase, the number of constrained objects increase, and as the white space decreases

3) The FSPPM enables larger sized and thus more difficult problems to remain tractable, problems that are intractable for the random method

The latter of these insights is a theme that will continue into the last part of Experiment 1 which follows. In this part of Experiment 1 however, it was observed that despite the increasing number of constrained objects that were present in the space, the FSPPM became more effective in generating feasible designs. This outcome implies that despite the problem increasing in constraint dimensionality (in the form of constrained objects) and thus difficulty, the FSPPM remains more than effective in generating feasible sequence-pair designs. Moreover, it seemingly thrives under such conditions. This combined with the other derived insights and result observations made before; the following conclusion can then be made. The developed FSPPM, effectively promotes the more frequent generation of feasible sequence-pairs designs when presented with a space consisting of constrained objects.

6.1.3 Experiment 1.C: 52 Problem Test Set Comparison

Having since validated the assumptions implemented to construct the FSPPM and moreover confirmed the FSPPMs ability to effectively promote the more frequent generation of feasible sequence-pairs in the preceding sub-experiments; the goal of Experiment 1.C was to then examine the FSPPMs ability to work in tandem with the Stage One optimization techniques to promote the discovery of feasible designs and in turn, provide more effective solution to problems. Furthermore, identification of the appropriate sigma value to deploy going forward was another secondary goal of Experiment 1.C.

6.1.3.1 Apparatus Setup

To test the FSPPMs ability to promote the discovery of feasible designs and therefore aid the optimization technique of Stage One, a modified variant of the *52 Problem Test Set* outlined in the Appendix F is leveraged. This test set is modified by altering the test problems with a 75% white space to have that of a 46% white space by means of altering the boundary dimensions. This was done to better observe the differences between the two methods. Other than this, the problem setups are identical to those provided in Appendix F. Only the problems consisting of constrained objects are examined provided that the FSPSPMs purpose is to facilitate the placement of such objects. This refines the *52 Problem Test Set* to a set of 32 problems (6-13, 19-26, 32-39, 45-52). These problems are solved using the baseline random assignment method along with the FSPPM for four different sigma values (0.6, 0.7, 0.8, and 0.9). For each of these five sampling method approaches, five replications of the 32 problems are solved. Each problem was solved for a fixed number of 100 generations such that a direct comparison on both a time and solution quality perspective could be made. The input parameters used in this experiment for the Stage One solver are provided for reference in Table 13. The raw results are also provided in Appendix G, Section G.2 for further reference.

Table 13 – Input optimization parameters of Experiment 1.C.

Optimization Parameter	Value
Population Size	200
Percent Elite	0.05
Percent Jumping Gene	0.4
Percent Crossover	0.7
Percent Mutation	0.05
Percent Feasible	0.8
Max Pop Initial Time	360
Max Generations	100
Max Stall Generations	100

6.1.3.2 Testing Results and Analysis

To directly compare the influence the FSPPM has on the solution performance relative to the more traditional random method, the results for the four FSPPM sigma value approaches are aggregated to form a singular result representative of the FSPPM for comparison purposes. The results that follow are a further refinement of the 32 problems noted before and are subject to no budgetary constraints or robustness considerations. Only the single period problems (6-13 and 19-26) are encapsulated in these numbers. The reason for this is that the random method was unable to solve the large sized dynamic problems of the set. Without solution, a direct comparison was thus impossible. This is again a direct observation of the random methods inability to frequently identify feasible solutions. For such sized problems this sampling method struggled to find a single feasible solution despite an exhaustive duration of time spent initializing the problem. With an initial population void of feasible solutions, the GA was virtually helpless in providing solution to the problem. It effectively entered an infinite loop in the first generation as it attempted to discover feasible solutions from infeasible parents. As alluded to before, the major implication of this is that without the FSPPM, larger sized problems subject to constrained spatial properties and constrained objects become unsolvable. With that established, the results of these single period problems are provided. Note that since the techniques used to generate a sample by both methods translate across periods, by extrapolation the differences between the two methods demonstrated below would only become greater as the number of periods is increased (if the random method could solve such problems to begin with).

The major performance metrics of time spent in the GA, time spent initializing the population (where the sampling method is directly leveraged), optimal objective value, and the unique solutions generated by the methods are examined for both methods in Table 14 and Table 15. Recall, that with Stage One initializing the populations of the Stage Two algorithm, these metrics will directly influence the Stage Two's performance and it is for this reason that the two methods are compared across these dimensions. The average result of these solution performance metrics for the two approaches is provided in Table 14. Along with it, the standard deviations of these metrics for each unique problem then averaged across the 16 single period problems examined, are provided in Table 15.

Table 14 – Random vs. FSPPM mean performance comparison

Method	Obs.	Time in GA (sec)	Time Initializing Population (sec)	Optimal Objective	Unique Solutions
Random	80	160.19	177.27	468527.78	1730.75
FSPPM	320	131.50	33.00	508718.75	1760.47

Table 15 – Random vs. FSPPM standard deviation performance comparison

Method	Std. Dev. Time in GA	Std. Dev. Time Initializing	Std. Dev. Opt. Objective	Std. Dev. Unique Solutions
Random	24.28	4.54	22357.98	303.44
FSPPM	14.09	2.65	21901.65	411.99

As observed, the FSPPM provides significant CPU time savings while initializing the population in Stage One. A difference of over 5 times less than that of the random method was observed on average. Additionally, the FSPPM provided improved CPU speed within the GA evolutionary process, reducing the time to solve the problems by about 30 seconds on average. The improved time within the GA is likely a by-product of the FSPPMs more refined placement of the objects. This makes reproduced offspring more likely to be feasible thereby speeding the process of populating the next generation and thus problem solution.

The FSPPMs faster solution did come at a cost though. Its greediness in placing the constrained objects resulted in it, on average, producing suboptimal solutions across the 16 problems examined. Interestingly though, the FSPPM on average outperformed the random method in discovering unique solutions, albeit fractionally (1.6% more designs). This difference could be attributed to the low sample size or it may be the result of the FSPPMs combined random and guided technique facilitating the more effective discovery of derivative, yet unique, solutions.

Now not surprisingly, the FSPPM is more consistent, i.e. robust, in terms of its GA time, initialization time, and objective function value. Its greater inconsistency in terms of the unique solution can be attributed to the greediness of the FSPPM. In some instances, it may become too localized in the design space, getting stuck in an overly refined area of the space thereby producing a more limited number of unique solutions. At the same time its localized approach enables it to find more derivative solutions. The interplay between these two competing behaviours is likely what leads to this greater spread in the unique solution results for the FSPPM. At the same time, the random method remains capable of avoiding such interplay and thus has a smaller spread relative to this metric.

Despite the observable inferior solution quality and greater spread in these performance metrics, the FSPPMs notable CPU time savings, more consistent solution speed, and optimal solution value makes it the preferred method in Stage One for initializing the population. Moreover, the random methods inability to solve problem sizes exceeding a single period makes it a rather limited and ineffective sampling method to deploy. As such, going forward, the FSPPM will be leveraged to provide solution in proceeding experiments. Having now established the FSPPM as the preferred method, identifying which sigma value to deploy to define the placement distribution of the constrained objects became the focus. To identify this value the results of the four different sigma value options tested were compared. The results of this comparison are provided in Table 16 and Table 17 below.

Table 16 – FSPPM sigma value mean performance comparison

FSPPM Sigma Value	Obs.	Time in GA (sec)	Time Initializing Population (sec)	Optimal Objective	Unique Solutions
0.6	80	129.65	32.70	511775.00	1730.09
0.7	**80**	**129.87**	**32.12**	**505412.50**	**1738.54**
0.8	80	130.93	33.08	509037.50	1780.53
0.9	80	135.55	34.09	508650.00	1792.73

Table 17 – FSPPM sigma value standard deviation performance comparison

FSPPM Sigma Value	Std. Dev. Time in GA	Std. Dev. Time Initializing	Std. Dev. Opt. Objective	Std. Dev. Unique Solutions
0.6	11.57	2.77	20272.10	376.78
0.7	**11.79**	**2.90**	**24018.97**	**428.15**
0.8	11.95	2.25	18851.30	414.15
0.9	21.05	2.67	24464.22	428.87

Overall, not a significant degree of differentiation between the different sigma values used in the FSPPM to define the placements of the constrained objects in the space is observed. With that said, there are some observable differences. While a sigma value of 0.6 produced the fastest GA solution times, it sacrificed a lot of optimality in doing so. Not surprisingly, it also generated fewer unique solutions. Such a low sigma value led to it being overly greedy, which is also what likely resulted in its inferior optimality discovery performance compared to the other options studied. It is also observed that as the sigma value increases the number of unique solutions increases, as can be expected. A higher sigma value means that the placement of the constrained object about its expected position is less known, or in this context, has a larger chance of being placed about it rather than right at it. This leads to a greater potential of discovering alternative solutions whereby the constrained object does not fall exactly at the expected position. This explains why the larger sigma options were able to find more unique solutions. Interestingly though, on average, this did not translate to finding a more optimal solution. In general, it can also be observed that as the sigma value decreases the time to initialize the population (i.e. find X number of feasible solutions) also decreases. This is intuitive. With the placement about the expected position becoming less uncertain, there is a smaller chance of not placing the constrained objects at their expected position and thus a smaller chance of finding an infeasible solution. A smaller chance of finding an infeasible solution translates to fewer executions of the generational loop and sampling method which then translates to CPU time savings.

Now, looking at the standard deviations of the performance results, which were taken across the five replications of each unique problem and then averaged across the problems for each sigma value option, there is once again not any major differences with regards to the solution times. The consistency of the options in generating unique and optimal solutions does however present more noteworthy differences. As observed, the 0.6 sigma option produced the lowest unique solution standard deviation, the second lowest relative to the optimal solution, and lowest with respect to time spent in the GA. Much of this can be attributed to its refined placement of the constrained objects, making it more consistent in these metrics. The sigma option of 0.8 produced the most consistent optimal solution discovery across the problem set tested and moreover was the most consistent in terms of its time spent initializing the solution. The sigma option of 0.7 was one of the least consistent options in terms of optimality discovery.

The goal of performing this study was to identify a suitable sigma option to deploy in subsequent experiments across the *52 Problem Test Set*. In light of these results and observations, a case could realistically be made for either a sigma value of 0.7 or 0.8. The other two options of 0.6 and 0.9 tend not to have any discernible advantages over that of the other two and thus are removed from the conversation. While the sigma option of 0.8 was more consistent in its optimal solution value, this optimal solution on average was less optimal than that produced by the 0.7 option tested. On average the 0.8 option also tended to produce a greater number of unique solutions. Despite this, the sigma option of 0.7 was ultimately chosen, as on average, it was faster in solving the problems and discovered optimality better than that of the 0.8 option all while only producing less than 2% fewer unique solutions. With the goal of Stage One primarily being to most rapidly generate unique solutions, the sigma option of 0.7 was chosen going forward as a result. Again though, a case could be made for either option as outlined before.

6.1.3.3 Experiment Summary and Conclusions

After an analysis of these results, the following conclusion can be made. For smaller problems the best strategy may be to deploy a purely random sampling method, while for larger less tractable problems the FSPPM would prove the better option. The results demonstrate that for smaller problems of dimensionality less than 6 objects, the random method was capable of more frequently discovering superior solutions while the FSPPM method did not perform as well, though still reasonably well. For such small problems, the random method remained capable of finding feasible solutions at a fast-enough rate that it remained able to fully populate the next generation in a reasonable duration. At the same time, the random nature of the method enabled it to discover alternative solutions, outside of what the FSPPM could discover, leading to the superior convergence performance. While this was true, it was noticeably slower in doing so which in lies its shortcoming while tackling larger sized problems. For larger problems, the random method simply could not generate feasible solutions at a reasonable rate for such a constrained and difficult problem formulation considered in this dissertation. In fact, for problem sizes greater than 6 and 12 objects and only a single period, it struggled to make it past the population initialization stage of the solution algorithm. Once entering the first generation it proved virtually impossible for the random method to discover enough feasible solutions to populate the next generation, essentially leading the algorithm to becoming stalled as it endlessly searched for feasible solutions. With few feasible solutions present to begin with, it made it only that much more unlikely that such a random method would find feasible solutions. This, however, was less a shortcoming of the FSPPM and truly where it began to shine.

While larger problems were unsolvable by the random method, the FSPPM enabled such problems to remain tractable. So, although the FSPPM may have been sacrificing some potential optimality, it had the major benefit of remaining capable of solving such larger and less tractable problems. Between the time savings and ability to solve larger problems, it can then be concluded that the *FSPPM method is more effective* *at generating feasible solutions and therefore should be deployed going forward*. Moreover, a *sigma value of 0.7 should be leveraged when deploying the FSPPM* to provide the most effective placement of the constrained objects.

6.2 Experiment 2: Advanced Flow Distance Method Testing

In Experiment 2, the novel advanced flow distance method developed in this research, which ensures flow path feasibility thereby providing closure to a major research gap, is tested. The advanced method is directly compared to the traditional rectilinear distance method to establish the importance of considering flow path feasibility when evaluating a layout design. This comparison is performed across the *52 Problem Test Set* and the results of this examination are provided next.

6.2.1 Apparatus Setup

To test the difference between a rectilinear distance method and the advanced distance method developed as part of this dissertation work to evaluate a layout design, the *52 Problem Test Set* provided in Appendix F was leveraged. Each unique problem of this set was solved using both the rectilinear distance method and the advanced method to determine the material handling distances. Within the developed performance model, all other costs (such as those at station) were rendered zero, apart from the other handling costs, in order to focus just on the impact that the different handling distance methods have on the cost function. Moreover, the capacities were set high enough in the problems of the *52 Problem Test Set* such that the dynamic production rate method implemented would not skew the results by reducing the rates and thus the costs. Five replications of each of these problem and method solution combinations were performed to enable an average result to be established. Furthermore, the rectilinear results were post-processed by applying the advanced method to both the best layout solution identified and the entire layout design set generated in Stage One (i.e. all feasible layout designs found). This was done to enable the difference between the two methods to be observed on a layout-basis. The Stage One optimization parameters provided in Table 13, and used in Experiment 1, were once more used here in Experiment 2 to produce the results presented. Additionally, the conclusions of Experiment 1 were leveraged here in Experiment 2. In other words, in solving the problems, the FSPPM was deployed along with a sigma value of 0.7. Less aggregated results, from that in which are presented next, are provided in Appendix G, Section G.2 for further reference. First though, a noteworthy observation made while producing the results of this experiment is presented.

6.2.2 A Notable Observation

While initially generating the results of this experiment, difficulties initializing the Stage One population were observed for the larger 12 object problems. While the algorithm was able to initialize the SLPs of just one period relatively easily, being capable of finding 160 solutions in a matter of seconds, initialization of the DLPs of three periods was a struggle, taking over 6 mins to find just 6 feasible solutions on average. This created a major roadblock. Because the DLPs were populated with so few feasible solutions, the GA then struggled to evolve the population. When the GA would select parents, there was a very high probability that one or both designs selected would themselves then be infeasible. In fact, there was just a 3% chance of selecting a feasible solution as opposed to an 80% chance (80% feasible designs were mandated for the initial population as an optimization parameter in Stage One) for the static variant of the same problem. As such, after the genetic operators passed the infeasible genetic material onto the offspring, there was a very marginal chance that the produced offspring would themselves be feasible. This is turn led to the GA spinning its wheels, so to speak, as it searched exhaustively and with little hope of creating offspring that would be feasible.

Interestingly though, it was observed that it took only 4.5 seconds or so to initialize the 12 object SLPs (i.e. generate 160 feasible solutions). This result spurred further investigation, which then identified that the reason for the difference has to do with the dimensionality of the problem being substantial less. For the DLPs, the algorithm had to simultaneously find sequence-pair and orientation-pairs that were feasible, not only in the first, but also second and third periods. So, while the first and third period may be feasible, if the second was not the whole design was considered infeasible. The SLPs on the other hand simply needed to establish a single layout, rather than three simultaneously, that were feasible. This spurred the question of how could this reduced dimensionality be leveraged going forward when initializing the DLPs?

With the literature and the concept behind a slot machine as inspiration, a revised approach to initializing the Stage One populations for DLP types was formed. Since it took only 4.5 seconds to generate the 160 required feasible solutions for one period, each period layout design could be independently initialized to retain a roughly linear time increase with respect to the number of periods forming the DLP, allowing feasible layout designs to then be formed at a faster rate.

The revised population initialization procedure then redeveloped for Stage One was as follows. Only for problems of the DLP type, a method resembling a slot machine was to be deployed. Once the first period layout design had been identified by the sampling method as feasible, just the second period design was then generated using the FSPPM method, just as before, until a feasible layout is discovered. Once this period was then found to be feasible, the algorithm continued to the next and so on and so forth until all periods contained feasible layout designs. This revised initialization approach to the DLP variants, yielded significant computational savings as can be observed in Table 18. The results presented are average CPU times for five replications of initializing the population of Stage One (200 population size, 160 feasible designs) for the 12 object problems of the *52 Problem Test Set*. Because for the original method, generating the required 160 feasible solutions was not achievable in the time frame allotted, the provided result is an extrapolation of the number of feasible solutions generated in the allotted six-minute time frame allowed to initialize the population in Stage One. In this time frame the average number of feasible designs was just six, so extrapolated to 160 designs yields the 9600 second figure provided below. As can be observed, the resulting revised initialization method created a reduction in CPU time of over 520 times, ultimately allowing the 160 feasible designs to be discovered in just under 20 seconds, which is far below the allotted 6 minutes mandated for initializing the Stage One population. In addition to these dramatic time savings in initializing the population, the fully populated initial population of 160 feasible designs subsequently enabled the GA evolutionary process to perform significantly better. The process was observed as no longer becoming stalled in the first generation, nor had difficulty in identifying feasible designs as it evolved the population.

Table 18 – Revised Stage One population initialization procedure CPU time comparison

Approach	SLPs	DLPs
Original Method	4.41	9600.00
Revised Method	4.41	18.40
Reduction Factor		521.80

Considering the substantial improvements gained in solution performance (time and optimality) by deploying this revised Stage One initialization procedure for the DLPs, this approach was then deployed from here forth in all subsequent experiments as well as in the results of this experiment. By leveraging this revised approach, it allowed problems that were before occasionally unsolvable, to now always be solvable.

6.2.3 Testing Results and Analysis

With the revised Stage One initialization procedure deployed, the results of Experiment 2 are now presented. Before presenting the results of this experiment, a few approach definitions and their symbolic representations must be summarized. Throughout this experiment, three distinct approaches are leveraged to establish the objective function values presented in the results to follow. The first approach leverages the simple rectilinear method in the performance model to establish the material handling costs and thus objective function value, provided the setup outlined before for this experiment. This approach in turn optimizes according to this rectilinear-based objective function. An adjacent approach to this, leverages these resulting layout designs of this previous approach and then applies posterior the advanced flow distance method to each design to enable a direct comparison between the two methods to be achieved. The final approach deployed in this experiment leverages the advanced flow distance method to establish the material handling costs and thus objective function value while solving the 52 problems of the test set. This approach allows the best layout design according to the advanced method to be directly compared to the best layout design according to the rectilinear method. This then enables the potential profit loss to be observed when a rectilinear method, which does not account for flow path feasibility, is used while optimizing. These outlined approaches and their symbol representations are summarized in Table 19.

Table 19 – Experiment two symbol definitions

Symbol	Approach
R	Optimized with rectilinear
A	Optimized with advanced
RA	Optimized with rectilinear, advanced post-applied

To test the novel advanced flow distance method, which ensures flow path feasibility, against the standard method in the literature, a rectilinear method, the two methods are first directly compared for the same layout designs. This is achieved by comparing the rectilinear optimization results (R) to these resulting layout designs posterior evaluated with the advanced flow distance method (RA). The results for the 12 problem SLPs are presented in Table 20. In these results, those for the population mean are the average objective function value across every design discovered to be feasible during the optimization process (in some cases this was over five thousand unique designs) whereas the optimal solution results are for just that of the best identified solution by the approach. Here the percent difference columns represent how much higher the design(s) costs are when the advanced method is posterior applied to the design(s).

Table 20 – Rectilinear vs. rectilinear post-processed with the advanced method for the twelve object SLPs

Problem	Characteristics	Optimal Solution			Population Mean		
		R	RA	% Difference	R	RA	% Difference
14	Non-constrained	162600	186800	14.9%	416400	1438000	245.3%
15		161200	235500	46.1%	434200	1404000	223.4%
16		168200	238750	41.9%	468400	804200	71.7%
17		156800	170400	8.7%	415600	399000	-4.0%
18		175600	217000	23.6%	459000	777400	69.4%
19	Constrained Higher White space	532200	455800	-14.4%	701600	1219000	73.7%
20		596000	532800	-10.6%	920600	1074400	16.7%
21		507200	445000	-12.3%	666600	587600	-11.9%
22		609400	533200	-12.5%	908200	797000	-12.2%
23	Constrained Lower White space	423200	392000	-7.4%	616400	822600	33.5%
24		475400	435500	-8.4%	926400	1382200	49.2%
25		410000	367250	-10.4%	584800	1319000	125.5%
26		469600	428500	-8.8%	791000	1278000	61.6%
Average				3.9%			72.5%

As can be observed, the major takeaway from these results is that across all the layouts discovered, i.e. the population mean, on average, the rectilinear method underestimates the material handling costs by over 70%. In other words, the costs for these designs are over 70% greater when flow path feasibility is considered. Such a difference in costs could dramatically impact a business's bottom-line, which is why accurately defining these costs using the advanced method is imperative.

For the optimal solution, on average the resulting costs are 4% higher. Most of this is a by-product of the unconstrained problems producing significantly higher costs, thereby outweighing the lower costs found for the other constrained problems. These lower costs for the constrained problems are a result of the advanced method creating direct paths that become shorter than the rectilinear distance. With the advanced method, a path can traverse from point to point in a direct line which, by Pythagorean Theorem, is shorter than the collective x and y changes, which is how the rectilinear method establishes the distances. It is for this reason, that for these solutions the direct shortest route results in a shorter distance and thus lower costs. Moreover, in these problems the fixed objects are placed so as to stretch the layout across the entire space. This stretching allows for more opportunity to shorten the path by traversing directly as opposed to in a rectilinear fashion. Regardless, this result still demonstrates that a notable difference exists between the two methods. On average for the optimal solutions, the two methods produce costs that are nearly 17% different (figure established by taking the average of the absolute of the percent difference column results). This difference only widens as you look across the entire population.

Now that the two methods have been compared directly and the differences established, how do the rectilinear optimized results then posterior evaluated for flow path feasibility with the advanced method (RA) compare to the advanced optimized results (A) for this same set of problems? This is answered by the comparison provided below in Table 21. The percent difference column here indicates how much lower of a cost the advanced method optimized layout design produces when compared to the rectilinear method layout design. Comparing these two results demonstrates that if the rectilinear method was to be deployed to establish the performance of the layout, how much inferior the resulting layout design would be if implemented in practice. In other words, how less efficient the design would be when compared to that design identified by the advanced method as being the best solution.

Table 21 – Rectilinear vs. advanced method for the twelve object SLPs

Problem	Characteristics	Optimal Solution			Population Mean		
		RA	A	% Difference	RA	A	% Difference
14	Non-Constrained	186800	163800	-12.3%	1438000	398000	-261.3%
15		235500	171200	-27.3%	1404000	406600	-245.3%
16		238750	171000	-28.4%	804200	471400	-70.6%
17		170400	171200	0.5%	399000	400200	0.3%
18		217000	159600	-26.5%	777400	418400	-85.8%
19	Constrained Higher White space	455800	430800	-5.5%	1219000	626400	-94.6%
20		532800	509400	-4.4%	1074400	802200	-33.9%
21		445000	419400	-5.8%	587600	600800	2.2%
22		533200	502400	-5.8%	797000	811200	1.8%
23	Constrained Lower White space	392000	359200	-8.4%	822600	558000	-47.4%
24		435500	398000	-8.6%	1382200	777400	-77.8%
25		367250	333600	-9.2%	1319000	523200	-152.1%
26		428500	387800	-9.5%	1278000	774800	-64.9%
Average				-11.6%			-86.9%

Except for a single outlier, Problems 17, across these 13 problems of the *52 Problem Test Set*, the best solution identified by Stage One is consistently inferior when flow path feasibility is not accounted for by the performance model. In other words, the rectilinear optimization results produce designs that are on average nearly 12% suboptimal in comparison to those identified by the Stage One algorithm when the advanced method was leveraged. Twelve percent may not seem significant until you consider that for a company with manufacturing costs on the order a million dollars. Twelve percent then becomes $120,000 in savings. Furthermore, this inferiority extends, in fact greatly widens, when viewed across the entire population set of Stage One (i.e. population mean results). As can be observed from the population mean results, on average across all replications and problems of this set examined, the designs identified are over 86.9% inferior when compared to the set produced during the optimization using the advanced method. Some of this can likely be attributed to the advanced optimization generating more unique solution in the local vicinity of the global optimum thereby reducing the population mean average. Regardless, this result once more establishes the distinct difference between the two methods and the clear inferior nature of the layout designs generated by the standard rectilinear method of the literature. In the case of the general population, the resulting design set is inferior by over 80%. That translates to an average cost of $800,000 more for a firm with costs of one million dollars. This only further justifies the need to consider flow path feasibility when designing a layout.

For comparison purposes, let’s observe the resulting difference in the best layout designs generated by the two methods for Problem 18. The resulting layout designs are demonstrated below in Figure 41, Figure 42, and Figure 43. Figure 41 is the design of the five rectilinear replications of Problem 18 that was the most optimal whereas Figure 43 provides the most optimal of the advanced replications of the same problem.

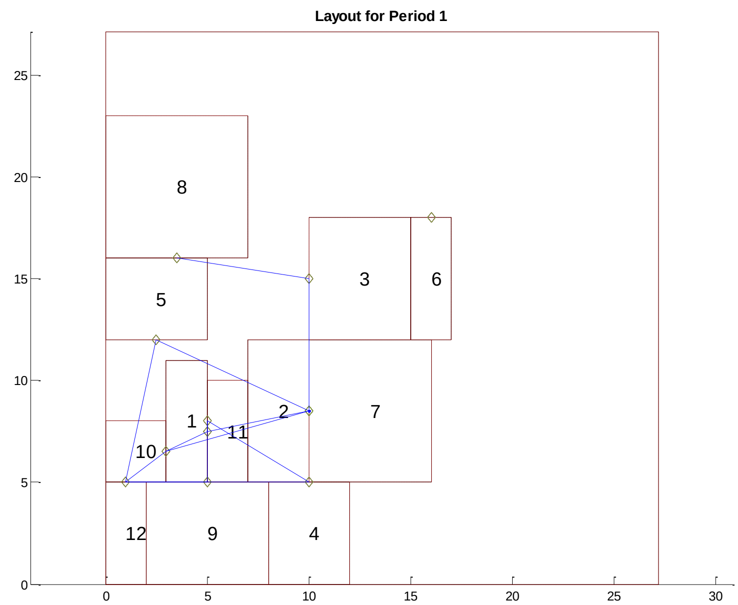


Figure 41 – Best layout design generated by the rectilinear method with direct paths

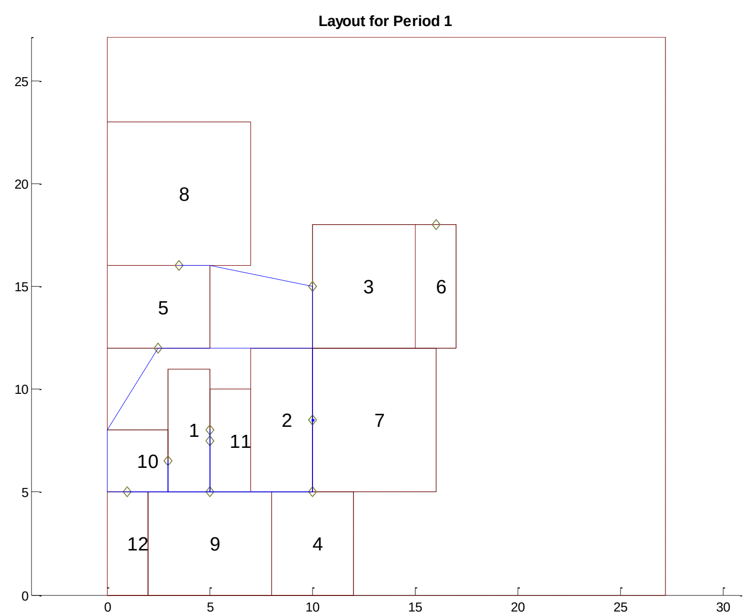


Figure 42 – Best layout design generated by the rectilinear method with flow feasible paths

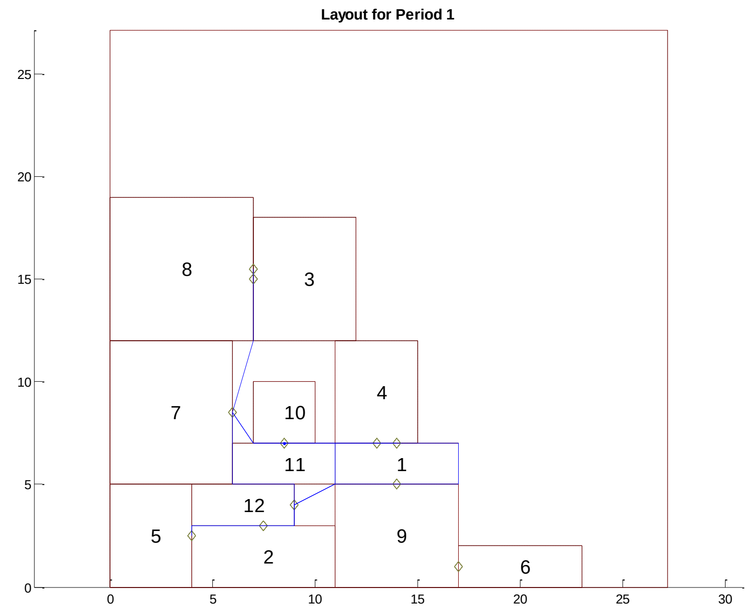


Figure 43 – Best layout design generated by the advanced method

As you can be observed by viewing Figure 41 or Figure 42 against Figure 43, the two layout designs identified as the best solution by the two methods are distinctly different. Also, the major flaw of the rectilinear method in evaluating the layout design is directly observable in this example. Observing Figure 41 and the direct paths shown (blue lines), many of these paths traverse directly through other objects. Objects 1, 11, and 2 are all vertically oriented to leverage its shorter dimension to effectively reduce the distance between I/O points. Doing so though forces the flow to then have to flow about their long dimension in order to get around to the opposing sides to then make connections between it and other objects when flow feasibility is considered (i.e. not traversing through another physical object in the space). This is directly observed in Figure 42 whereby the same layout design is posterior evaluated with the advanced method and as shown, the paths must travel about the ends of these objects. This is the major flaw of the rectilinear or Euclidean methods of the literature and is what originally motivated the development of the advanced flow distance method at the start of this research. By not accounting for such flow path feasibility, the resulting design that is considered best is ultimately significantly inferior to the design that, when implemented in practice, would be the most effective, i.e. that shown in Figure 43. In this example, the design identified as the best by the rectilinear method was over 30% less optimal than it could have been had the optimization procedure considered flow path feasibility. This means a layout design that is less effective and by extension, one that would yield lower production profit margins due to higher material handling costs.

Having since established the difference between the two methods and moreover the importance of considering flow path feasibility via the advanced method, a broader look at the comparison across the entire problem set is presented next. The difference between the population means of the rectilinear and rectilinear results post-processed with the advanced method are provided in Table 22, distinguished on a problem type and size-basis as well as overall. The percent difference results represent how much the objective function increases by applying the advanced method that ensures flow path feasibility or put alternatively, how much the rectilinear method underestimates the true material handling costs for the layout design. As can be observed, the rectilinear method on average greatly underestimates the material handling distances and thus costs, more so for the larger sized problems of 12 objects. This is likely because there are more objects the material must flow about, which effectively increases the material handling distance and therefore costs. On average across the two problem sizes the rectilinear method underestimates costs by nearly 40, that is for the SLPs.

The results of the DLPs can be misleading. The reason for why the 6 object problems produces a population mean with lower costs is not because the rectilinear method overestimates the costs on average. Instead, this result is likely the by-product of the DLPs not converging to the extent that the SLPs had due to the added dimensionality of the problem. By not converging sufficiently, the population mean would effectively be higher as fewer neighbouring designs about the global optimum would be found, which would effectively reduce the population mean. With so many unique layout designs generated for each replication of each problem and furthermore multiple layouts to be processed for each period of each layout design, this meant processing millions of designs for the 12 object DLPs. As such, the results of the 12 object DLPs post-processed with the advanced method were unfortunately too computationally burdensome to process and as such could not be provided here as a direct comparison. It is believed that had this post-processing been achievable, a similar result to that observed for the 6 object DLPs would too have been observed. Overall, across the two problem types, the rectilinear method still significantly underestimates the costs at about 25%.

Table 22 – Average difference between rectilinear and rectilinear post-processed advanced distance method population mean values

Problem Characteristics		Difference		
Type	Size	Size + Type	Type	Overall
SLP	6	6.17%	39.31%	24.09%
	12	72.45%		
DLP	6	-6.35%	-6.35%	
	12			

In addition to this examination, the difference between the post-processed rectilinear population mean results and the advanced population means results are provided and summarized in Table 23. A similar outcome is observed here whereby the percent differences now indicate how much lower, on average, the population layout designs discovered while using the advanced method are in comparison to those discovered while using the rectilinear method. As such, a negative number here symbolizes how much lower the advanced method results are. As demonstrated in the table, for the SLPs the advanced method results in a population with a cost objective of nearly 50% less than that of the rectilinear population with most of this difference being attributed to the 12 object SLPs. These, on average, have populations with an average cost nearly 90% less than that of its rectilinear counterpart. Once more, the 6 object DLPs demonstrate the contrary. Again, likely due to a lack of adequate convergence when compared to the lower dimensionality SLPs. Regardless, on average across both problem types, the difference still remains such that the advanced method produces designs that are on average about 30% more optimal or put alternatively, the rectilinear method produced designs that are on average 30% less ideal when put into practice whereby flow path feasibility must be considered.

Table 23 – Average difference between rectilinear and advanced distance method population mean values

Problem Characteristics		Difference		
Problem Type	Problem Size	Size + Type	Type	Overall
SLP	6	-8.26%	-47.57%	-29.84%
	12	-86.89%		

Table 23 (continued)

DLP	6	5.63%	5.63%
	12		

At this point, the importance of considering flow path feasibility and thus of the novel advanced flow distance method developed as part of this dissertation has since been well established, thereby providing substantiation of Hypothesis 1. Now though, attention is turned towards whether this necessitates the need to implement the advanced method during the first stage of the proposed bi-model multi-stage solution approach. To consider this, comparisons of the rectilinear results post-processed with the advanced flow distance method are compared to that of the advanced results across the problem set on a problem type and size-basis. A summarization of these results relative to the best identified layout designs solutions objective function value (Optimal Obj.) the number of unique designs identified, and the time required to solve the problems are provided in Table 24. Positive values in this table for the differences indicate how much better the metric result is for the rectilinear method relative to the advanced method. A lower objective value, a higher number of unique solutions generated, and a faster/lower solution time are all better. In that sense, anytime a positive value is observed in the table, it indicates the preference lying with that of the rectilinear method (i.e. the rectilinear performs better in that dimension).

Table 24 – Comparison of methods across the metrics of interest in Stage One

Approach	Problem Type	Problem Size	Optimal Obj.	Unique	Time (min)
RA	SLP	6	125307	2757	2.0
		12	372877	5790	2.3
	DLP	6	451969	8508	3.9
		12	1253754	11253	5.5
A	SLP	6	123868	2707	12.0
		12	321338	5222	90.1
	DLP	6	430277	8164	39.4
		12	1192103	6352	185.7
Difference	SLP	6	-12.9%	1.7%	10.0
		12	-14.5%	8.9%	87.8
	DLP	6	-15.9%	7.0%	35.5
		12	-12.3%	107.6%	180.2

From the provided results it became quite evident that in terms of true optimality, the rectilinear method consistently underperformed that of the advanced method approach. The advanced method, as expected provided its consideration of flow path feasibility, identified layout designs that on average were over 12%, across all problem types and sizes, better configured for efficiency. In other words, an over 12% reduction in costs can be realized by considering flow path feasibility when optimizing the design. While this is true, the rectilinear consistently outperformed the advanced method in terms of identifying unique designs and problem solution. Most notably the rectilinear method produced over 107.6% more feasible designs than did the advanced method for that of the larger 12 object DLPs (average value on an instance basis). The reason why this result was so much more significant compared to the others has to do with the added dimensionality of the problem. Due to the size of these problems, the algorithm did not converge to the level in which the others had, resulting in fewer solutions identified throughout the process. Much of this can be attributed to the stark difference in computation time required to solve the problems across all problem types and sizes.

As is evident, the rectilinear method consistently outperforms the advanced method in terms of computational speed. Being that the advanced method is solving a variant of the traveling salesman problem for each unique flow connection in the design, this is not all that surprising. The difference is solution time is quite stark, which further fuels the debate as to whether the advanced method needs to be implemented when only initializing Stage Two. As shown, while across all problem types and sizes the rectilinear method can solve the problems in five and half minutes or less, even the smallest sized problem could not be solved in less than 10 minutes when the advanced method was used. Moreover, the difference between the two methods solution times only further grew as the problem size grew, whereby growth was mostly linear relative to both the number of objects and number of periods (SLP, a single period; DLP, three periods). While the largest problems, the 12 object DLPs, could on average be solved in just five and half minutes with the rectilinear method, it took the advanced method over three hours to provide solution to the same problems. This created a difference in time of over 180 times. In other words, it took the advanced method over 180 times longer to provide solution to the problems.

6.2.4 Experiment Summary and Conclusions

As a reminder, the purpose of this experiment was to test the difference between using the rectilinear distance method of the literature and the advanced distance method developed in this dissertation. Additionally, the experiment sought to demonstrate the importance of considering flow path feasibility when evaluating a layout design in order to substantiate Hypothesis 1. As a reminder Hypothesis 1 was as follows:

> **Hypothesis 1:** **If** an advanced flow distance method that ensures flow feasibility is implemented to define the MHCs, **then** improved layout designs that are better representative of reality can be established for variable production environments where several interrelated processes are occurring concurrently.

As was well established throughout this experiment, leveraging the novel advanced flow distance method consistently provided improved layout designs that were better representative of reality. This was demonstrated by comparing the results of the rectilinear method to that of the advanced method across the *52 Problem Test Set* and noting the differences between the two. It became evident that in order to most accurately design a layout, the novel flow distance method was required. The rectilinear method consistently underestimated the true material handling costs by underestimating the handling distances.

Though the advanced method is necessary to identify the most optimal layout design in practice, its necessity in Stage One was further examined. When Stage One is functioning as a means of initializing the Stage Two algorithm, and more complex layout model representation, optimality is not the only metric that should be considered. In the last part of the experiment, the performances of both the rectilinear and advanced method were examined across the entire problem set and relative to all three-key metrics. While, as demonstrated before, the rectilinear method underperformed the advanced method in terms of optimality, it outperformed the advanced method in terms of both unique designs identified as well as solution time. The rectilinear method more than just outperformed the advanced method; it significantly outperformed it relative to solution time, yielding solution times, in some cases over 80 times faster. Instead of taking hours to run a problem, it took mere minutes to do so. This was always a well understood drawback of considering flow path feasibility, but as established before, a necessary one to ensure a layout design is accurately evaluated.

With that said, the stark CPU time savings yielded by leveraging the rectilinear method in Stage One to initialize Stage Two, and additionally the larger unique population sets generated, both collective outweigh giving up roughly 12% in optimality that occurs when not considering flow path feasibility via the advanced method. With a main goal of the Stage One being to rapidly generate feasible designs to enable Stage Two to then be initialized effectively, this sacrifice seems reasonable. As such, going forth it is recommended that when Stage One is functioning as a means of initializing the Stage Two algorithm, the rectilinear method ought to be leveraged to provided effective and fast solution to the problems. This conclusion will be leveraged going forth in later experiments. On the other hand, when Stage One is to function as the end of the layout design process, the advanced method must be deployed to ensure that the resulting design yielded via the optimization will in fact be the best one when implemented in practice.

6.3 Experiment 3: FSA Implementation in Stage One

In Experiment 3, the FSA technique of Stage One is tested, and its requirement in the bi-model multi-stage solution approach examined. Once more, this examination and testing is performed across the *52 Problem Test Set*.

6.3.1 Apparatus Setup

To test the FSA method and establish its need for inclusion in the bi-model multi-stage solution approach, the *52 Problem Test Set* provided in Appendix F was leveraged once more. Additionally, the conclusions of the preceding experiments, Experiment 1 and 2 are leveraged. In other words, the FSPPM method with a sigma value of 0.7 is deployed and the rectilinear method implemented to define the flow distances. The Stage One optimization parameters summarized in Table 13, and used in Experiment 1 and 2, were once more used here in Experiment 3 when solving the problems. This was done to enable a direct comparison to be performed between the results for when the FSA technique is included and for when it is not (i.e. those generated in Experiment 2). Five replications of each of the 52 problems of the set are solved with the FSA technique applied in Stage One to the fittest solution each generation. The purely rectilinear results (R) from the proceeding experiment are leveraged here to provide comparison. In implementing the FSA technique, the FSA parameters provided in Table 25 were deployed to generate the results that will be discussed next.

Table 25 – FSA optimization parameters

Optimization Parameter	Value
Maximum Number of Iterations	15
Sample Size	500
Probability of Uphill Move Acceptance	0.9
Probability of Reassigning Fixed Object	0.7
Probability of Swapping Adjacent Objects	0.8
Probability of Rotating Unconstrained Objects	0.9
c Coefficient (higher = more greedy search)	100
k Coefficient (higher for larger problems)	3
McKendall Method Option (ON/OFF)	ON

6.3.2 Testing Results and Analysis

Deploying the FSA optimization parameters noted above and applying the FSA technique to the further refinement of the best solution each generation in the Stage One GA yields the results presented in Table 26 when compared to those results found before in Experiment 2 for when FSA was not applied. The results presented in Table 26 are for the yielded solution objective values and run times after 100 generations or until the max time limit of 3 hours and 20 minutes (i.e. 200 minutes) was achieved. Additionally, the CPU time per generation is also provided to enable a better comparison to be made between the two approaches as it removes any dependency on the number of generations executed and thus the impact the time limit may have had. This limit became relevant when solving the DLPs of the problem set, as will be observed.

Table 26 – Comparison of Stage One results with and without FSA included

Problem Characteristics		Difference in Metrics		
Type	Size	Optimal Obj. (%)	Run Time (mins)	Time/Generations (sec)
SLP	6	-10.4%	36	11
	12	-3.3%	56	17
DLP	6	-2.5%	206	78
	12	-6.1%	208	108
Average		-5.57%	126	53

Observation of the results in Table 26 sheds great insight into the need to deploy the FSA technique while solving the problems in Stage One. On one hand, deploying the FSA each generation on the best solution enables improved solution optimality to be achieved by the Stage One GA. On average across the 52 problems tested, the improvement in optimality was over 5.5%. This improvement was more prominent in the 6 object SLPs. The improved optimality can certainly be attributed to the further refinement of the solution. It was often observed in the non-FSA results that while the object placements were well positioned for optimality, the rotations sometimes did not follow suit. It is believed that because the FSA technique was deployed with such a high probability of rotation (90%), this is likely the major reason for the improved optimality that was yielded by deploying the FSA technique in Stage One.

While optimality discovery was by far a major advantage of deploying the FSA technique, it came at a cost, and a major one at that. As can be observed, when FSA was deployed, the Stage One algorithm was noticeably slower regardless of the characteristics of the problem. In the case of the DLPs, the algorithm terminated often well before reaching 100 generations as the maximum time limit was reached. At the same time, when not deployed, the algorithm breezed through the 100 generations, on average, in under 6 minutes, even for that of the DLPs. The time per generation results really demonstrate the difference between the two approaches and the amount of time the FSA contributes to the run time. Since all else remained the same to when the FSA was not deployed, the difference provided in Table 26 demonstrates the added time per generation that can be attributed solely to the FSA techniques execution. Moreover, this result is for just that of 15 iterations and a sample size of 500 deployed, both of which arguably are on the low end of what could be considered sufficient for solution by the FSA technique. Increasing either would only further increase the difference between the two approaches.

In post-processing the results of this experiment, many of the FSA convergences were observed to be very abrupt, whereby the final optimal solution was achieved well in advance of the algorithm terminating, whether be it after 100 generations or once the time limit was reached. With all convergence data having been saved, and thus available, this occurrence was further studied. Given the improved optimality performance, yet abrupt convergence behavior and thus many stalled generations present (i.e. generations where no further improvement in solution was achieved), it was of interest to see how the FSAs solution time would compare to the time it took to achieve a solution at least as optimal as the result yielded when FSA was not deployed. With per generation times along with the optimal solution value each generation available, this comparison was possible.

Taking the number of generations to achieve optimality by each approach and multiplying it by the per generation time, yielded the results provided below in Table 27.

Table 27 – Time to optimality comparison

Problem Characteristics		Time to Optimality (mins)		Difference
Type	Size	FSA Off	FSA On	
SLP	6	23.9	1.2	22.7
	12	45.2	2.0	43.2
DLP	6	166.0	3.3	162.7
	12	197.7	5.3	192.4
Average		108.2	3.0	105.2

As can be observed in Table 27, though the FSA technique was able to, on average, reach optimality sooner than when the algorithm terminated, it still took on average, across the problems, close to 2 hours to do so. At the same time the FSA took on average a mere 3 minutes, creating a difference of over an hour and forty minutes between the two approaches. What about the time the FSA approach took to achieve the optimality of the FSA excluded approach, however? The results answering this question are provided in Table 28. As can be observed, that while the FSA approach is able to reach the optimal solution of the FSA excluded approach about 10 minutes (108.2 from previous table subtracted by 95.8 here) sooner than reaching its own optimal solution, it still takes considerably longer for this approach to reach the same level of optimality discovery; over an hour and half longer in fact. Even for the smallest problem size examined, it still took over 20 minutes longer to achieve the same level of optimality.

Table 28 – Time to FSA Off optimality when FSA On

Problem Characteristics		Time (mins)
Type	Size	
SLP	6	21.5
	12	44.1
DLP	6	160.7
	12	156.9
Average		95.8

6.3.3 Experiment Summary and Conclusions

The results of this Experiment 3 provide clear insight into whether the FSA technique needs inclusion in Stage One. While it was demonstrated that the FSA technique clearly improved solution optimality, achieving this optimality came at a cost. This cost was an algorithm in Stage One that provided solution at a much slower rate, taking noticeably longer per generation. If optimality is the only objective of Stage One, in other words, the designer is solving the problem to finality using just Stage One, then it would be imperative to deploy the FSA technique. The longer solution times in this case would be acceptable provided that a better solution would then likely be found by the algorithm.

Since Stage One acts in its primary function as a means of initializing the Stage Two algorithm, more than just the solution optimality metric must be considered. As was the case before in Experiment 2, here the substantially longer solution times that accompany the FSAs deployment, make it non-ideal for inclusion while acting in initializing Stage Two. With the algorithm not only taking longer, but also often not reaching completion of all 100 generations; the algorithm has fewer opportunities to identify other potential solutions, thereby reducing the set of solutions that can be provided to Stage Two and moreover reducing the diversity of this solution set.

Between the FSA techniques significantly longer solution times and reduced diversity of the design set passed to Stage Two, the combination of the two outweigh the FSAs ability to provide, on average, 5.5% more optimal solutions. As it was also recommended that the rectilinear method be deployed, and it itself also lacks optimality with respect to flow path feasibility and thus true optimality, this is not such a sacrifice. Overly optimizing for optimality based on the rectilinear result could do more harm than good. As such, when initializing Stage Two, the FSA technique should not be deployed. This result will be leveraged going forth in subsequent experiments.

6.4 Summarizing Statements of Experiment Set A

As a reminder, the purpose of Experiment Set A was to test the effectiveness of the FSPPM, the novel advanced flow distance method, and the need to infuse FSA into the first stage of the proposed bi-model multi-stage solution approach. In Experiment 1, it was demonstrated that the FSPPM performs far better than the traditional random assignment method of the literature. The FSPPM demonstrated its ability to discover feasible layout designs at a far faster rate and as such, made problems before unsolvable to then become solvable.

In Experiment 2, it was demonstrated that the novel advanced flow distance method does well in ensuring flow path feasibility. It was also demonstrated that there exists a distinct difference between the traditional rectilinear result and that generated when deploying the advanced method. Moreover, it was proven that optimizing relative to the rectilinear result yields noticeably inferior layout designs compared to when the advanced method is used. These results provided substantiation to Hypothesis 1:

> **Hypothesis 1: If** an advanced flow distance method that ensures flow feasibility is implemented to define the MHCs, **then** improved layout designs that are better representative of reality can be established for variable production environments where several interrelated processes are occurring concurrently.

In the final experiment, Experiment 3 of this set, it was demonstrated that while infusing the FSA technique provides improved optimality, the substantial time cost associated with its execution deters its application in Stage One when initialization of Stage Two is the goal.

Overall, it was concluded that the FSPPM ought to be deployed with a sigma value of 0.7 regardless of the goal of Stage One (initialization or final solution). In the case of initialization, it was concluded that the rectilinear method be deployed and FSA not. On the other hand, when final solution and thus optimality is the goal of Stage One, the advanced flow distance method and FSA ought to both be deployed to ensure the best layout design is identified by the Stage One algorithm.

CHAPTER 7

EXPERIMENT SET B: OPTIMIZATION STUDIES

The goal of this chapter is to present the results of the Experiment Set B. This set consists of two distinct experiments, Experiments 4 and 5, as outlined before. As a reminder, the purpose of this experiment set is to test the following:

Purpose: Test the effectiveness of the Stage One and Two solution procedures of the proposed bi-model multi-stage solution approach to solving the complex layout formulation of this dissertation. Test different optimization parameter combinations to identify the appropriate parameter sets to deploy to most effectively solve said layout problems.

Additionally, it should be acknowledged that the experiments build upon each other, whereby the resulting conclusions of the preceding experiments are leveraged in future experiments. From the preceding experimental results and the conclusions formed, the FSPPM with a sigma value of 0.7 is leveraged, a rectilinear material handling distance method deployed in Stage One, and the FSA technique turned off in the experiments of this Experiment Set B.

Note, that without solution to a comparable layout formulation to compare to in the literature, the solution procedures cannot be compared to other solution procedures to establish effectiveness relative to a baseline. Many of the procedures developed in this dissertation are tailored to the advanced and uniquely formulated, layout problem addressed in this dissertation research. As such, comparison across other formulations is not possible. The best alterative was then to establish a literature standard baseline for such a formulation, then establish a new problem set to test the procedures upon, and finally perform optimization parameter studies to then identify the most effective approach to solving said problems.

7.1 Experiment 4: Stage One Optimization Study

In Experiment 4, the effectiveness of the developed Stage One solution procedure is tested, and the best optimization parameter sets identified. To test the Stage One solver's effectiveness, its ability to solve the *52 Problem Test Set* is examined. The *52 Problem Test Set* problems are solved using different optimization parameter combinations to identify both the parameter set that provides the best optimality discovery (to deploy when only Stage One is to be executed) and a set that will provide the second stage with the best population set to initialize its populations with (to be referred to as the initialization set going forth). The latter directly address the posed question of to what extent the first stage needs to be solved to adequately initialize the Stage Two populations.

7.1.1 Apparatus Setup

To achieve this parameter set identification, an L18 orthogonal array screening design of experiment (DOE) was leveraged to identify the best combination of Stage One optimization parameters for optimality and initialization. The following Stage One optimization parameters were examined: population size, elite, jumping gene, crossover, mutation, and initial population feasible solution percentages, as well as the number of generations to be performed. Table 29 outlines these DOE factors considered (i.e.

optimization parameters) and their associated factor levels (i.e. settings) that were tested in the experiment.

Table 29 –Factor table for Stage One optimization parameter study

Factor		**Levels**		
Symbol	Description	Level 1	Level 2	Level 3
A	Population Size	50	100	200
B	Percent Elite	0.05	0.1	
C	Percent Jumping Gene	0.4	0.6	
D	Percent Crossover	0.7	0.9	
E	Percent Mutation	0.05	0.15	
F	Percent Feasible in Initial Population	0.8	1	
G	Number of Generations	50	75	100

Table 61, found in Appendix G, Section G.5, provides the L18 orthogonal array leveraged to perform the optimization parameter study for Stage One. For each of these experiments, five replications of the *52 Problem Test Set* problems were solved for a total of 5 replications x 18 experiments x 52 problems = 4680 solution instances. An L18 array was chosen provided that the number of degrees of freedom totalled 9 and additionally a complete full factorial design of these factors would have taken on the order of months to run across several machines. Therefore, a more selective and efficient approach was required.

Once each of these 4680 solutions were generated, the results were then post-processed. Post processing entailed transforming the performance metrics to a relative percentage deviation (RPD), which effectively normalized the data for cross comparison purposes and such that they could then be subsequently transformed into signal-to-noise

(S/N) ratios. The relative percentage deviation equation leveraged to establish the RDP value for each problem of each experiment was as follows:

$$RDP_p = \underset{r}{\text{mean}}\left(\frac{Metric_{p,r} - LB_p}{LB_p}\right) \tag{66}$$

where p is each problem of the *52 Problem Test Set*, r the replication, $Metric_{p,r}$ the metric value for replication r of problem p, and LB_p the minimum discovered metric value across all replications and experiments for problem p. With the RDP values for each of the 52 problems determined for each of the 18 experiments tested, the S/N ratios were then computed. The metrics of optimal objective value, solution time, and unique solutions generated were all examined as these three-solution metrics directly impact the quality of the solution set provided to Stage Two and moreover, to the Stage One's solution effectiveness.

Provided that these three-performance metrics of interest: optimal objective value, solution time, and number of unique solutions generated each have differing preferred values, the appropriate S/N ratio equation needed to be applied to each. For the optimal objective value and solution time, smaller is better, but for the unique solution metric, larger is better. The appropriate S/N ratio provided below was applied to each metric result:

$$S/N\ Ratio_k = \begin{cases} -10\log_{10}\left[\dfrac{1}{M}\displaystyle\sum_{p=1}^{M} RPD_p^2\right], & smaller\ is\ better \\ -10\log_{10}\left[\dfrac{1}{M}\displaystyle\sum_{p=1}^{M} \dfrac{1}{RPD_p^2}\right], & larger\ is\ better \end{cases} \tag{67}$$

where M are the number of problems and *k* is the trial experiment.

These S/N ratios were then leveraged to establish the S/N ratios for each level of each factor tested by taking the average S/N ratio across those experiments consisting of that factor level. Doing so enabled the main effect of each factor to be established and subsequently examined to identify the appropriate factor level to deploy for each optimization set sought. Based on how the developed jumping gene operator was designed, which works on a period-basis, the operator is rendered unexecuted for the static single period problems of the test set. As such, inclusion of the static problems would skew the results. Thus, to avoid this, the two distinct problem types were separated, and individual analysis performed for each of them. For the static layout problems, the results of the first 26 problems of the *52 Problem Test Set* results were examined (i.e. problems 1-26). For the dynamic problems, the last 26 problems of the test set results were examined (i.e. problems 27-52). The jumping gene factor result in the SLP results can be neglected given its non-application to such a problem type.

7.1.2 Testing Results and Analysis

The computed S/N ratios for each of the metrics of interest for the 26 SLPs and the 26 DLPs are provided in Table 68 in Appendix G, Section G.5.

Leveraging the S/N ratio results of Table 68, the S/N ratios are averaged for each level of the control factors and plotted against them. Below in Figure 44 and Figure 45 the results for the optimal objective mean S/N ratios values across the two problem types (SLP vs. DLP) are plotted. Those for the other two metrics of interest are provided in Figure 84, Figure 85, Figure 86, and Figure 87 in Appendix G, Section G.5. Since the aim is to maximize the S/N ratio, the level with the highest S/N ratio is considered the factor level that produces the best metric value, in this case the optimal objective value. In other words, the factor levels that best discover optimality. Observation of these main effects plots of the control factors indicates positive relationships relative to population size, percent mutation, and the number of generations in the SLP dataset while negative relationships are observed relative to the other factors, though many of them are marginally negative and thus could be considered neutral. As for the DLP dataset, population size and number of generations once more have distinct positive relationships, but interestingly, the mutation percentage is negative, while the percent elite is now positive. With respect to population size and number of generations across both, diminishing returns can be observed across each problem type, apart from population size in the SLPs where an increasing return in optimality is observed as the population size increases. Considering these results, the optimal parameter settings that enable the solution procedures of Stage One to best discover optimality across both problem types are identified and provided in Table 30.

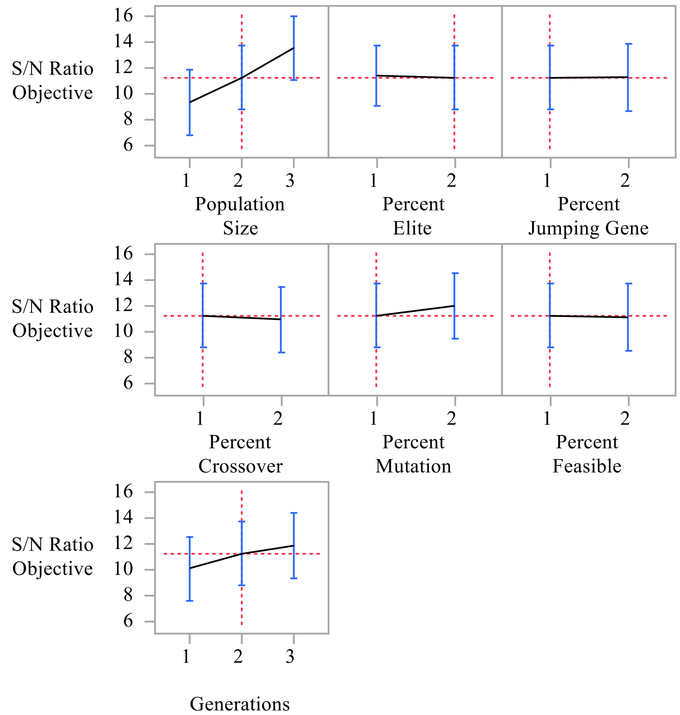


Figure 44 – Mean optimal objective S/N ratio for each level of the control factors for the SLPs in Stage One

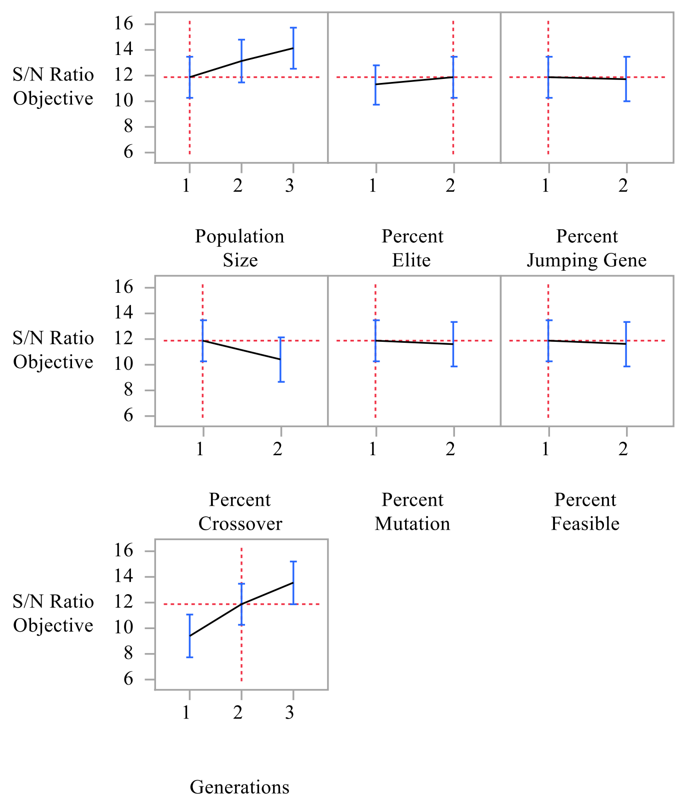


Figure 45 – Mean optimal objective S/N ratio for each level of the control factors for the DLPs in Stage One

Table 30 – Best control factor settings to achieve optimality in Stage One

Factor	Description	SLP Best Levels	DLP Best Levels
A	Population Size	200	200
B	Percent Elite	0.05	0.10
C	Percent Jumping Gene	0.60	0.40
D	Percent Crossover	0.70	0.70
E	Percent Mutation	0.15	0.05
F	Percent Feasible in Initial Population	0.80	0.80
G	Number of Generations	100	100

Not surprising, the optimal settings leverage the highest levels of both the number of generations and population size across both problem types. An 80% feasible initial population is preferred along with a lower level of crossover and jumping gene operation execution (neglect in the SLPs). While a large amount of mutation is desired in the SLPs, a lower amount is preferred in the DLPs; a rather interesting result. This could be a result of the added dimensionality resulting in non-feasible solution discovery occurring with this added variability. In the SLPs, the larger mutation likely enables the algorithm to avoid becoming stuck within local minimums and likewise a low elite percentage in these SLP is preferred which follows this same logic. It may prefer to be higher in the case of the DLPs to home in on the optimal solution faster and before reaching the maximum number of generations.

These results were further analysed by leveraging a multifactor analysis of variance technique (ANOVA). The ANOVA analysis enables the relative importance of each control factor's main effects on the response variable, S/N ratio of the optimal objective value in this instance, to be identified. The results of the ANOVA analysis for both the SLP and DLP datasets are provided in Table 69 and Table 70 respectively and can be found in Appendix G, Section G.5. Observation of the P-value statistics indicates that population size and number of generations are the two most significant factors in both problem types. While this is true, the relative importance of each of these is flipped between the two problem types. In the case of the SLPs, the population size is the most significant factor, attributing to over 64% of the variation, followed by the number of generations, which contributes just a bit over 11%. The contrary is the case in that of the DLPs. In the DLPs, the number of generations is now the most significant, attributing over 60% to the variation while the population size is less so at only about 18%. This could be the result of the added dimensionality in the DLPs, whereby it requires additional generations to converge to optimality. In the case of the SLPs and their lower dimensionality, the greater population size is more important in allowing the algorithm to cover the complete design space effectively. Both control factors being the most significant is also expected and therefore provides a good confirmation of expectations. In either of the problem types the combination of these two control factors attribute to over 75% of the variation in the datasets (84% in the SLPs, 77% in the DLPs). Additionally, and as expected, the jumping gene control factor was the least significant in the SLPs. Recalling that in this problem type the jumping gene factor is irrelevant, this result validates expectations. Interestingly though, even in the DLPs dataset, the jumping gene control factor is the least significant factor of the set attributing to only about 0.1% of the variation in the data. This result will be leveraged going forth into Experiment 5 when the optimization parameters chosen as control factors in the Stage Two optimization parameter study are to be established.

Having since identified the best parameter set to provide optimality discovery, the parameter set that generates the best population set for initializing the second stage's initial populations is now examined. The mean S/N ratios across all three metrics of interest, optimality, solution time, and unique solutions generated, for each level of the control factors are leveraged to do so. Once more, this analysis is performed individually for the SLP and DLP datasets. The results plotted in Figure 44, Figure 84, and Figure 85 for the SLPs along with those plotted in Figure 45, Figure 86, and Figure 87 are simultaneously leveraged to identify such a parameter set that would best initialize Stage Two. Instead of observing just the optimal objective metric, as was done before, a more balanced approach is required to establish this set. A multi-criterion weighted average approach was deployed for a series of weighting schemes applied to each of the three metrics of interest. The results of these weighting schemes are provided in Table 31 and Table 32 for the SLP and DLP problem datasets respectively.

Table 31 – Preferred factor levels for different weighting schemes for the SLPs in Stage One

Metric	Weights				
Objective	**0.45**	0.40	0.30	0.00	0.00
Time	**0.10**	0.30	0.50	1.00	0.00
Unique	**0.45**	0.30	0.20	0.00	1.00
Factor	**Preferred Levels**				
A	**200**	200	100	50	200
B	**0.10**	0.10	0.10	0.05	0.10

Table 31 (continued)

C	**0.60**	0.40	0.40	0.40	0.40
D	**0.90**	0.90	0.90	0.70	0.90
E	**0.15**	0.15	0.15	0.05	0.15
F	**0.80**	0.80	0.80	1.00	0.80
G	**75**	50	50	50	75

Table 32 – Preferred factor levels for different weighting schemes for the DLPs in Stage One

Metric	Weights				
Objective	**0.45**	0.40	0.30	0.00	0.00
Time	**0.10**	0.30	0.50	1.00	0.00
Unique	**0.45**	0.30	0.20	0.00	1.00
Factor	**Preferred Levels**				
A	**200**	200	100	50	200
B	**0.10**	0.10	0.10	0.10	0.10
C	**0.60**	0.60	0.60	0.60	0.60
D	**0.90**	0.90	0.90	0.70	0.90
E	**0.05**	0.05	0.05	0.05	0.15
F	**0.80**	0.80	0.80	0.80	0.80
G	**100**	100	50	50	100

The weighting schemes presented in Table 31 and Table 32 are only a few considered. Observation of these, as well as others, demonstrated that for the preferred population size to ever change from 200, at least a weight of 50% had to be applied to the time metric in either problem type to result in the swing to the second factor level. This is due to both the objective and unique solution metrics preferring the higher factor level to maximize its performance relative to these two metrics. Looking at the second to last column, the best time scheme, not surprisingly the number of generations and population size are at the lowest levels. In the last column, best unique solutions generated scheme, as expected the population size is preferred to be at the highest level tested. Interestingly though, the number of generations is preferred to be at the middle factor level of 75 generations in the case of the SLPs, which goes against intuition. Looking further at the data, the difference in the S/N ratios relative to this metric from factor setting two to three (75 to 100) was relatively small. As such, either factor would be acceptable here. In the case of the SLPs, this is likely a result of diminishing returns on unique design discovery as the optimizer surpasses 75 generations. At this point, the algorithm has sufficiently converged, likely leading to it not finding many new unique solutions in the additional 25 generations. As for the DLPs, the added generations provide the algorithm with more opportunities to discover unique solutions across the larger design variable dimensionality that is present in the DLPs. This explains why the higher factor setting is preferred for that problem type.

Ultimately, a balanced approach between the objective and unique solution metric was chosen as the best set to deploy when initializing Stage Two. Since the observed solution times for the 52 problems were all quite reasonable, more emphasis was placed on the other two metrics. The split was chosen to be 45/10/45 as demonstrated in the first weighting scheme provided in Table 31 and Table 32. This approach balances both optimality and diversity in the population set that is provided to the second stage yet does not completely neglect the time factor. Moreover, all these solutions will be feasible. As such, optimality, diversity, and feasibility are all well balanced in the population set provided to Stage Two to initialize the populations with this weighting scheme. As noted, this balance in the initial population is paramount to the success of not only the genetic algorithm of this dissertation, but that of any genetic algorithm. The best Stage One optimization parameter set values to deploy when initializing Stage Two are thus summarized in Table 33.

Table 33 – Best control factor settings when initializing Stage Two

Factor	Description	SLP Best Levels	DLP Best Levels
A	Population Size	200	200
B	Percent Elite	0.10	0.10
C	Percent Jumping Gene	0.60	0.60
D	Percent Crossover	0.90	0.90
E	Percent Mutation	0.15	0.05
F	Percent Feasible in Initial Population	0.80	0.80
G	Number of Generations	75	100

As is demonstrated in Table 33 by the setting for the number of generations, overly solving Stage One should be avoided when solving SLPs. This is a direct acknowledgement of the prior proposed question of to what extent Stage One should be solved when initializing Stage Two. As is evident, the preferred option falls in the middle of the range considered at 75 generations for the SLPs. At this level, a sufficient population set can be constructed by Stage One to then adequately populate the Stage Two solution procedure. At the same time, when solving DLPs, a greater number of generations should be leveraged to provide additional convergence and unique solution discovery. This outcome validates expectations.

7.1.3 Experiment Summary and Conclusions

As was observed above, it can be concluded that to most effectively solve problems in Stage One, the end goal of Stage One should be considered along with the problem type before defining the best parameter settings to deploy. If the end goal is to solve the problems leveraging only that of Stage One, then the optimality parameter sets provided in Table 30 should be deployed. This configuration of parameter settings yields the best optimality discovery on average across the *52 Problem Test Set* examined in this experiment. When the goal of Stage One is to initialize Stage Two rather than just solving the problem to optimality however, the parameter sets yielding the best balance of time, optimality, and unique solution generation are presented in Table 33. Though the two parameter sets are very similar they do have some notable differences such as the number of generations (in the case of the SLPs) and many of the reproductive factor settings to deploy. While the optimality set prefers lower levels of jumping gene and crossover, the initialization set prefers higher. Considering these results, the parameter sets best for initializing Stage Two, and provided in Table 33, are then leveraged going forward in subsequent experiments.

7.2 Experiment 5: Stage Two Optimization Study

In Experiment 5, the effectiveness of the developed Stage Two solution procedure is tested, and the optimization parameter sets that best result in optimality identified. To test the Stage Two solver’s effectiveness, its ability to solve the *52 Problem Test Set* is once more examined. Like before in Experiment 4, the *52 Problem Test Set* problems are solved using different optimization parameter combinations to identify the parameter set that provides the best optimality discovery. With Stage Two being the last solution procedure of the proposed bi-model multi-stage solution approach, optimality is the core and moreover only objective of this solution procedure.

7.2.1 Apparatus Setup

Before the Stage Two solution procedures could be tested, Stage One first needed to be executed to establish the generated population sets to then be leveraged to initialize the populations of Stage Two. With the identified Stage One optimization sets not being exactly one of the 18 experimental trials tested, five replications of the *52 Problem Test Set* problems were first solved with the optimization parameter set values identified before for initializing Stage Two and provided in Table 33. As was the case before in Experiment 4, the same FSPPM setup, material handling, and FSA configurations noted at the beginning of this chapter were deployed in generating these results. Once the solutions to these 260 instances (52 problems x 5 replications) were obtained, the Stage Two solution procedures were then able to be tested.

A notable difference in testing Stage Two compared to the testing of Stage One performed in the preceding Experiment 4, is that in solving the *52 Problem Test Set* in Stage Two and in performing the optimization study, the advanced flow distance method was deployed to calculate the material handling distances in place of the rectilinear method. Doing so allows Stage Two then to solve the overarching problem formulation of this dissertation, a MILP modelled DLP solved with a flow distance method ensuring flow path feasibility.

Now to achieve this parameter set identification in Stage Two, an L18 orthogonal array screening design of experiment (DOE) was once again leveraged to identify the best combination of Stage Two optimization parameters for optimality across both problem types (SLP and DLP). Given the dimensionality of the optimization parameters available in Stage Two with the tri-population GA scheme deployed, the parameters to consider needed to be selected strategically to avoid the computational time from becoming unmanageable.

Recall that in Stage Two, four distinct populations are evolved. For each of these, individual optimization parameter values could in theory be defined for each of the parameters. These four populations are outlined below in Table 34, which includes their symbol representation, title, general description, and period of the Stage Two GA in which their solution occurs within. As observed, the first three populations are directly initialized from that of the results generated in Stage One, while the fourth is a merged population occurring after the isolation period and consisting of a combination of the three isolation populations. The symbol representations in this table will be leveraged going forward to distinguish which population the control factors are associated with.

Table 34 – Stage Two control factor symbol definitions

Symbol	Title	Description	Period
1	Population 1	Stage I Best Designs	Isolation
2	Population 2	Stage I Anti-Best Designs	Isolation
3	Population 3	Stage I Random Designs	Isolation
M	Population Merged	Migrated of Isolation Populations	Post-Isolation

The following Stage Two optimization parameters were examined in the study: population size, jumping gene, crossover, mutation, isolation generations, merged generations, as well as the migration rate (i.e. how the merged population is formed from the isolation populations). Table 35 outlies these DOE factors considered (i.e. optimization parameters) and their associated factor levels (i.e. settings) tested.

Table 35 –Factor table for Stage Two optimization parameter study

Factor		**Levels**		
Symbol	Description	Level 1	Level 2	Level 3
A	Population Size (1 | 3)	100	200	300
B	Population Size (2 | M)	50	100	200
C	Percent Jumping Gene (1 | 2 | 3 | M)	0.2	0.4	
D	Percent Crossover (1 | 2 | 3)	0.7	0.9	
E	Percent Crossover (M)	0.7	0.9	
F	Percent Mutation (1 | 2 | 3)	0.1	0.15	
G	Percent Mutation (M)	0.1	0.15	
H	Isolation Generations (1 | 2 | 3)	15	35	
I	Merged Generations (M)	50	75	100
J	Migration Rate	(0.1, 0.3, 0.6)	(0.3, 0.3, 0.4)	(0.6, 0.1, 0.3)

As can be observed, the elite percentage was not one of the selected parameters. This decision was based on its relative contribution to the variation observed in Experiment 4, where it was not a major contributor. Inclusion of the jumping gene control factor was included provided its unique application in this dissertation. With that said, provided that the jumping gene was the least significant factor in Stage One across either problem type, the factor became an aggregated parameter that was applied uniformly across all populations in Stage Two. Two separate factors were applied to the four populations whereby the sizes of the one and three populations were combined into a single factor while the two and merged populations were combined into another. This grouping was done as in the developed solution approach the merged population size is dictated by the minimum population size of the isolation populations. In this case, that would often likely be population two, which consists of the anti-best solutions from Stage One and had previously been observed in prior literature research as being a less desirable population of the three in contributing to the performance of the algorithm. Both the crossover and mutation percentages were separated on an isolation and post-isolation population-basis to observe how the two may contribute differently whether being applied in each period of the algorithm. The number of isolation generations became another factor along with that of the merged number of generations. Being a significant factor in Experiment 4, it was appropriate to consider two distinct factors here. Moreover, an understanding of how long the isolation period should proceed for is useful knowledge to obtain and can be done so with this approach. The migration rate, or the composition of the merged population from that of the isolation populations, became the last control factor in the study. This was done to observe which distribution would result in the best performance of the algorithm. Note, the first decimal value relates to the fraction of the merged population that is taken from population one, the second from two, and the third from the third population, as defined before in Table 34.

Table 71, found in Appendix G, Section G.6, provides the L18 orthogonal array leveraged to perform the optimization parameter study for Stage Two relative to the established control factors. For each of these experiments, five replications of the *52 Problem Test Set* problems were solved for a total of 5 replications x 18 experiments x 52 problems = 4680 solution instances in total. An L18 array was chosen provided that the number of degrees of freedom totalled 14 and additionally a complete full factorial design of these factors would have taken on the order of months to run, which is why a more selective and efficient approach was chosen.

Like before in Experiment 4, once each of these 4680 solutions were generated, the results were then post-processed. Post processing entailed transforming the optimal objective metric (i.e. objective function) to a relative percentage deviation (RPD) and then to a S/N ratio using Equations (66) and (67) outlined before. Only the optimal objective value was analysed here as it is the only metric of importance in Stage Two.

7.2.2 Testing Results and Analysis

The computed S/N ratios for the objective function for the 26 SLPs and the 26 DLPs, which make up the *52 Problem Test Set,* are provided in Table 74. Table 74 can be found in Appendix G, Section G.6.

Leveraging the S/N ratio results of Table 74, the S/N ratios are averaged for each level of the control factors, and then plotted against these factor levels. Below in Figure 46 and Figure 47, the results for the optimal objective mean S/N ratios values across the two problem types (SLP vs. DLP) are plotted. Since the aim is to maximize the S/N ratio, the level with the highest S/N ratio is considered the factor level that produces the best optimal objective value. In other words, the factor levels that best enables the algorithm to discover optimality.

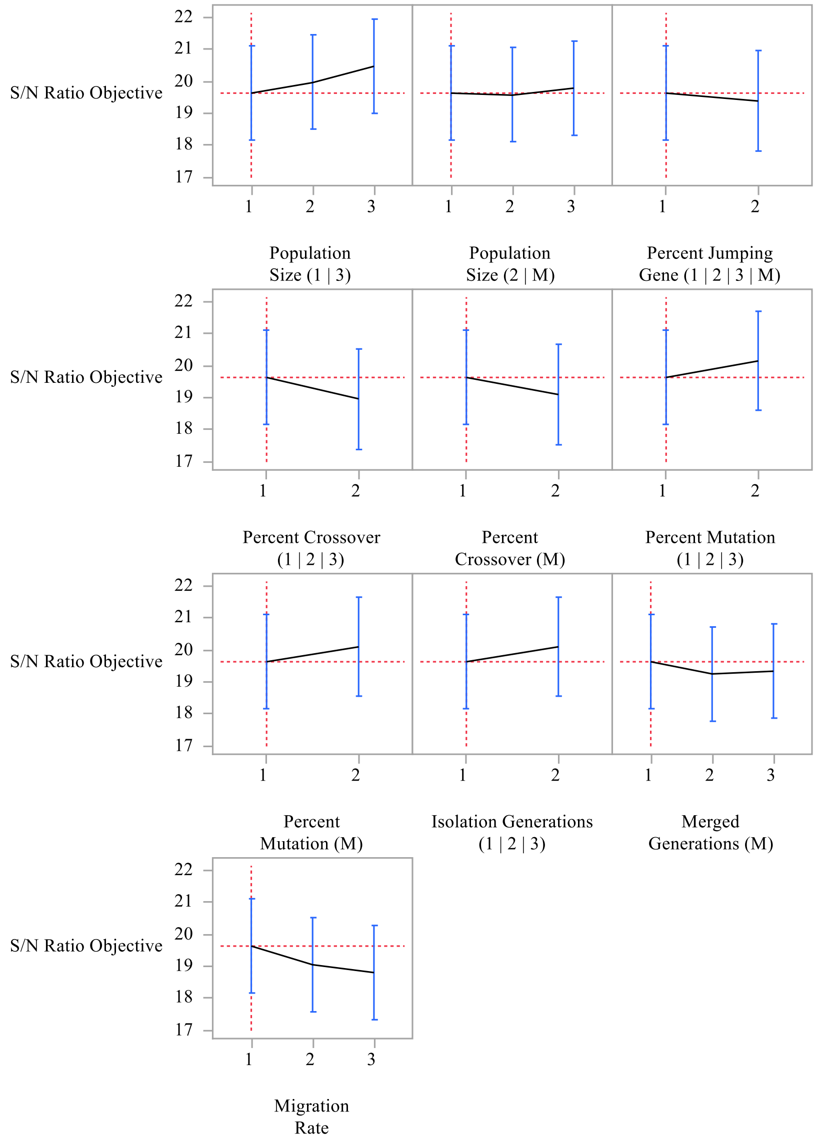


Figure 46 – Mean optimal objective S/N ratio for each level of the control factors for the SLPs in Stage One

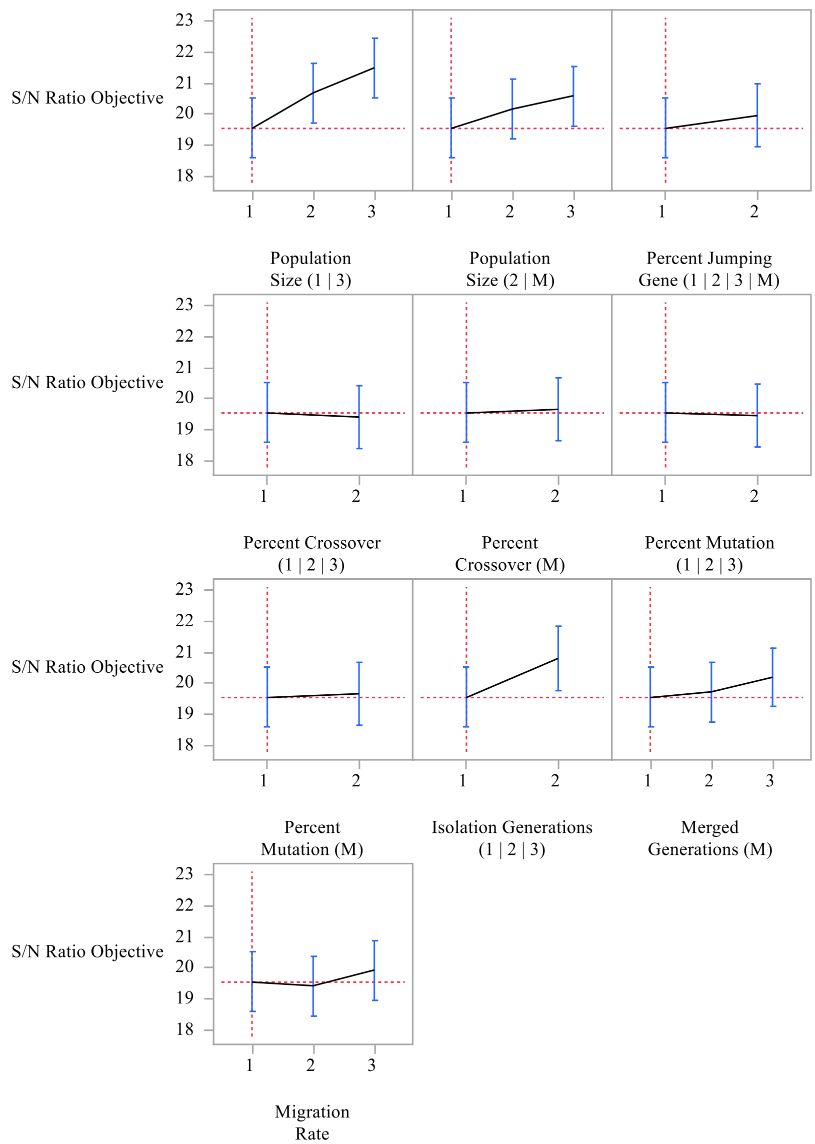


Figure 47 – Mean optimal objective S/N ratio for each level of the control factors for the DLPs in Stage One

Observation of these main effects plots of the control factors indicates very different behaviours across the two problem types. While a distinct positive relationship is observed relative to population size of the 1 | 3 populations in both datasets, the population size of 2 | M have a distinct positive relationship in the case of the DLPs but a rather neutral one in the SLPs. A positive relationship is observed in the DLPs, which is contrary to what was observed in Stage One. A distinct negative relationship is observed in relation to both crossover factors for the SLPs, while in the DLPs they are rather neutral with the merged crossover factor demonstrating a slight positive relationship thereby indicating that more crossover is preferred. Common among both problem types, more mutation is preferred, which generally aligns with what was observed before in Stage One for the SLPs (mostly neutral in the DLPs), and more isolation generations are preferred, which is to be expected.

The number of merged population generations show distinctly different trends. While in the SLPs the relationship is generally negative, the relationship is distinctly positive in the DLPs whereby more generations are preferred. This could be a result of the added generations providing additional exploration and convergence, which is beneficial with the higher design variable dimensionality of the DLPs. The preferred migration rate is also very different between the two problem types. In the case of the SLPs, the first factor level is clearly the preferred one whereby more of the merged population should be composed of designs from the random design isolation population (60% from population 3). On the other hand, factor level three is the preferred in the case of the DLPs. The preference in this case is for the merged population to consist of 60% from that of the best design isolation population (population 1). This result confirms the earlier observation regarding the second population (anti-best designs) contributing the least to the performance of the tri-population algorithm. Observation of these results enables the optimal parameter settings that allow the solution procedures of Stage Two to best discover optimality across both problem types to then be identified. The optimal settings are summarized below in Table 36.

Table 36 – Best control factor settings to achieve optimality in Stage Two

Description	SLP Best Levels	DLP Best Levels
Population Size (1 | 3)	300	300
Population Size (2 | M)	200	200
Percent Jumping Gene (1 | 2 | 3 | M)	0.20	0.4
Percent Crossover (1 | 2 | 3)	0.70	0.7
Percent Crossover (M)	0.70	0.9
Percent Mutation (1 | 2 | 3)	0.15	0.1
Percent Mutation (M)	0.15	0.15
Isolation Generations (1 | 2 | 3)	35	35
Merged Generations (M)	50	100
Migration Rate	(0.1, 0.3, 0.6)	(0.6, 0.1, 0.3)

Not surprising, the optimal settings leverage the highest levels of both the population size control factors across both problem types. Moreover, in general a higher level of the mutation factors and isolation generations are preferred, which meets expectations. Generally, a lower level of crossover is preferred, that is except for the merged populations of the DLPs where a larger amount is desired to best provide optimality. This is likely a result of the higher crossover providing improved alteration across the multiple periods of the DLP layout designs (i.e. higher design variable dimensionality). As mentioned before, the preferred migration rates differ across the two problem types. A merged population consisting of more random population designs is preferred in the SLPs while a merged population consisting of more best population designs is preferred in the case of the DLPs.

These results were further analysed by leveraging a multifactor analysis of variance technique (ANOVA) like was done before in Experiment 4. The results of the

ANOVA analysis for both the SLP and DLP datasets are provided in Table 75 and Table 76 respectively and can be found in Appendix G, Section G.6. Observation of the P-value statistics for the SLPs indicates that the most significant factors are that of the migration rate, followed by the population size of 1 | 3, and then that of the crossover in the isolation populations (1 | 2 | 3). These three factors account for about 50% of the variation in the data. Interestingly, the population size of 2 | M, followed by the merged generation factor, are the two least significant factors. In general, the remaining factors account for the remainder of the variation, each contributing less than 10%. As for the DLP dataset, the population size of 1 | 3 factor accounts for over 45% of the variation followed by the isolation generations (1 | 2 | 3) which account for another 27%. That means that over 70% of the variation in the data is associated with these two factors making them the most significant of those tested. Add to these the contribution of the population size of 2 | M, and over 85% of variation in the data is accounted for by the population size and the isolation generations factors. This result is not at all surprising and aligns well with preceding observations made in this experiment and Experiment 4 from earlier. Interestingly in the DLP dataset, the crossover and mutation factors collectively account for less than 1% of the variation, making them the least significant of the factors tested.

7.2.3 Experiment Summary and Conclusions

As was observed before, it can be concluded that to most effectively solve problems in Stage Two, the problem type should be considered before defining the best parameter settings to deploy. To best solve the problems to optimality, the parameter sets provided in Table 36 ought to be deployed. These configuration of parameter settings yield the best optimality discovery, on average, across the *52 Problem Test Set* examined in this experiment, which as a reminder, covers a range of different problem characteristics and conditions.

7.3 Summarizing Statements of Experiment Set B

Recall, the purpose of Experiment Set B was to test the effectiveness of the solution procedures developed as part of the bi-model multi-stage solution approach proposed for solving the complex layout problem formulation of this dissertation. Additionally, identifying how to most effectively solve said layout problems was another core element of the experiment set. In performing these tests, Hypothesis 2 was in turn directly substantiated. As a reminder Hypothesis 2 was as follows:

> **Hypothesis 2:** **If** the proposed bi-model multi-stage hybrid solution approach is implemented to solve the MIP formulated RDLP, **then** the problem will be solved most effectively, in terms of solution quality.

Without a directly applicable problem formulation available in the literature to compare to, the best alternative was then to set a literature standard and moreover identify how to best tune this solution approach to best solve the uniquely formulated problems of this dissertation. To this extent, an extensive optimization parameter set was performed across the constructed *52 Problem Test Set* of this dissertation. The best settings for the tested parameters were identified as a result of these experiments performed. The results establish that to most effectively, in terms of solution quality (i.e. optimality), solve the uniquely formulated and complex problems of this dissertation, the Stage One and Two parameter settings provided in Table 33 and Table 36 respectively should be deployed. Furthermore, if the simplified formulation, yet still considerably complex and moreover unique one of Stage One is to be solved most effectively, in terms of solution quality (i.e. optimality), then the optimal settings summarized in Table 30 should then be deployed.

CHAPTER 8

EXPERIMENT SET C: CASE STUDY

The goal of this chapter is to present the results of Experiment Set C. This set consists of the final experiment, Experiments 6, which builds upon the previous experiments conducted in an attempt to then provide substantiation to the overarching hypothesis of this dissertation. As a reminder, the purpose of this experiment set is as follows:

Purpose: To test the LIVE methodology by applying it to a real-world layout design problem and to test its ability to enable designers and stakeholders to make more informed and collaborative decisions.

The focus of the experimentation to this point has been in testing and validating the novel methods and solution procedures developed exclusively to handle the unique problem formulation of this dissertation. Having since well tested these, attention is now turned towards examining the LIVE methodology's (enabled by these methods and procedures) ability to provided solution to a real-world layout design problem. In doing so, the sheer power of the methodology and the value it creates for designers and stakeholders alike is then able to be observed.

Now as noted before, Experiment 6 builds upon the previous experiments performed. As such, in solving the problems of this set, the conclusions of Experiment Set A and B are leveraged to ensure the most effective solution is achieved with the proposed bi-model multi-stage solution approach in the LIVE methodology. This means, in stage one solution is achieved by leveraging the FSPPM, performed without FSA infused, deploying the simple rectilinear flow distance method, and finally assigning the optimization parameters best for initialization as identified in Experiment 4. In Stage Two, the advanced flow distance method is deployed and the optimization parameters identified in Experiment 5 to best establish optimality are deployed.

8.1 Experiment 6: A Real-World Case Study

In Experiment 6, the developed LIVE methodology is applied to the redesign of an aerospace parts storage and distribution warehouse layout. The study considers both the restructuring of operations and reconfiguration of the existing layout design. The operations and layout of a 53,000 square foot warehouse area are examined. This warehouse area falls between two attached buildings, each over quadruple the square footage and which act primarily as storage facilities for the parts; stacked in rows of 30-foot-tall racks lining the buildings. At the center of these two buildings, and where this study focuses its efforts, the core receiving, part storage preparation, and distribution operations occur. The baseline configuration of the current layout is provided in Figure 48 to give the reader an appreciation for the scale of the problem examined and the general setup of the facility.

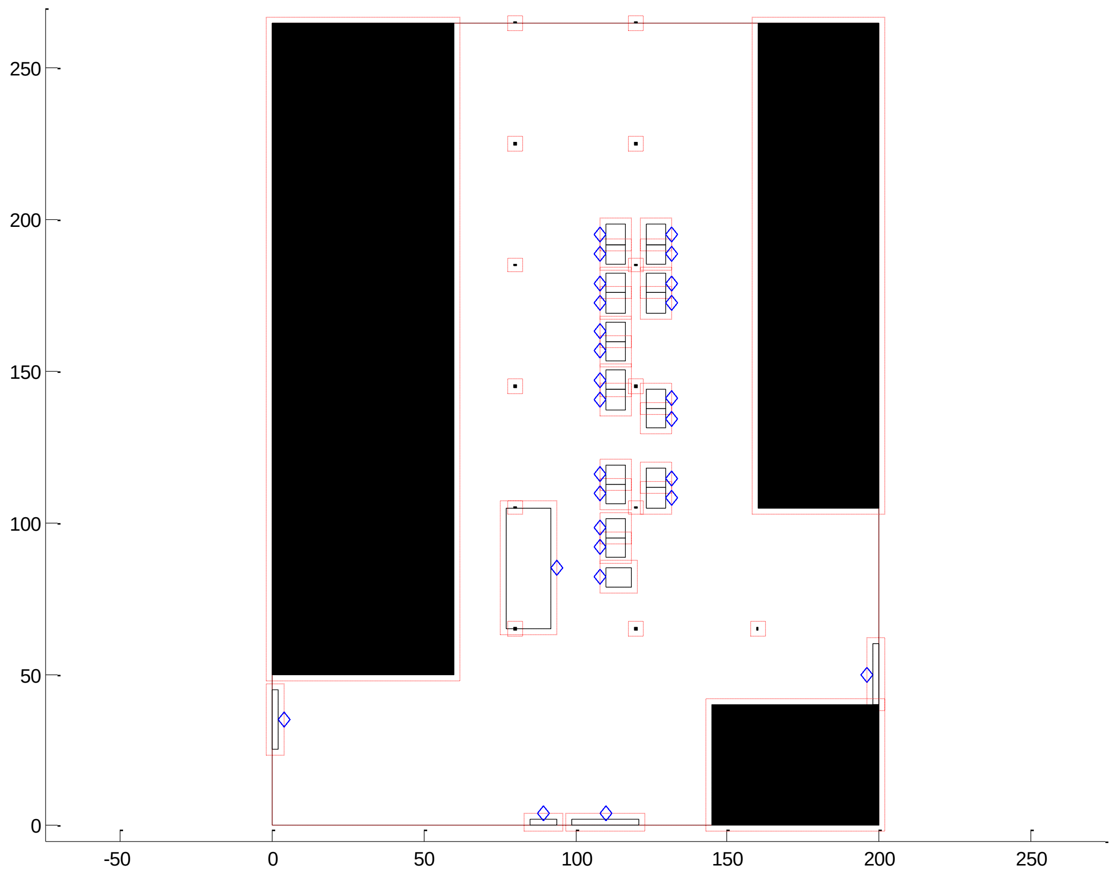


Figure 48 – Baseline configuration of the current layout

The facility consists of a multitude of different sized objects (to be referred to as stations or regions from here forth) along with a large white space, as is observable in Figure 48. Building A and C doors lead to the two storage buildings noted before. The bottom two doors represent the receiving and shipping docks where the parts are load onto or removed from UPS, FEDEX, and other shipping company trucks. The black regions are representative of regions whereby stations cannot be placed. The top-left region is an area of more part racks, the bottom-right and top-right both sales and management office spaces. As can be observed, small (relative to the rest of the space and objects present), structural pillars are littered in a 40’ x 40’ gridded structure throughout the space. At the center of the space lies a horseshoe layout of workstations surrounded by a manual roller conveyor belt. This is where the primary operational activities occur. This horseshoe is referred to as the cross-dock. Finally, a staging area for large parts is situated left of this cross-dock. All remaining white space is unused, but available for use. All doors and regions are assumed non-movable.

8.1.1 Operational Landscape

In this environment, several operations are occurring simultaneously and moreover spread across the twenty-one current workstations of the cross-dock system (excluding the non-manned staging area and door stations). Parts are being received, shipped, inspected, packaged, staged, and in some instances retrieved from Building A (left door in baseline figure presented earlier).

After studying the system and, the cross-dock operations, it was identified that the operations could be decomposed into five distinct processes types. These process types were labelled as purchase orders (PO), sales orders (SO), repair order inbound to be shipped (ROIS), repair order inbound to be inventoried (ROII), and finally repair order outbound (ROO) processes. This decomposition is summarized in Table 37. Going forth each of these processes will be referred to by their acronym. An understanding of each of these five distinct process types and their flow throughout the space enabled a clear grouping to become apparent based on the stations and ordering thereof in each process. It was found that PO and ROIS processes have identical process flow characteristics, whereby parts are received, processed in, inspected, processed out, and then packaged and staged, when applicable, before being shipped. These process flows along with the others present in the system are presented in Figure 49. Due to the nature of this process flow, the synthesis of the PO/ROIS processes were then labelled as pure cross-dock processes as these parts originate from the receiving door and end at the shipping door. In the system under study, the goal was for these parts to have a one-day turnaround, whereby parts were to be received and shipped within one business day.

In parallel to these two processes, it was identified that the SO and ROO processes shared common characteristics, whereby parts were retrieved from Building A, inspected, then packaged and staged, when necessary, before being then being shipped. The ROII process is, in many ways, the reverse of this, whereby the parts are received and then inspected before being sent to the racks in Building A. Due to the nature of these process flows, the synthesis of the SO/ROO/ROII processes were labeled as rack processes since the parts originates or end at the Building A rack door (i.e. in the racks). The SO/ROO processes are further grouped provided they have identical process flows, which are distinctly different from that of the ROII process flow.

Table 37 – Process definitions

Acronym	Process Definition
PO	Purchase Order
SO	Sales Order
ROIS	Repair Order Inbound to be Shipped
ROII	Repair Order Inbound to be Inventoried
ROO	Repair Order Outbound

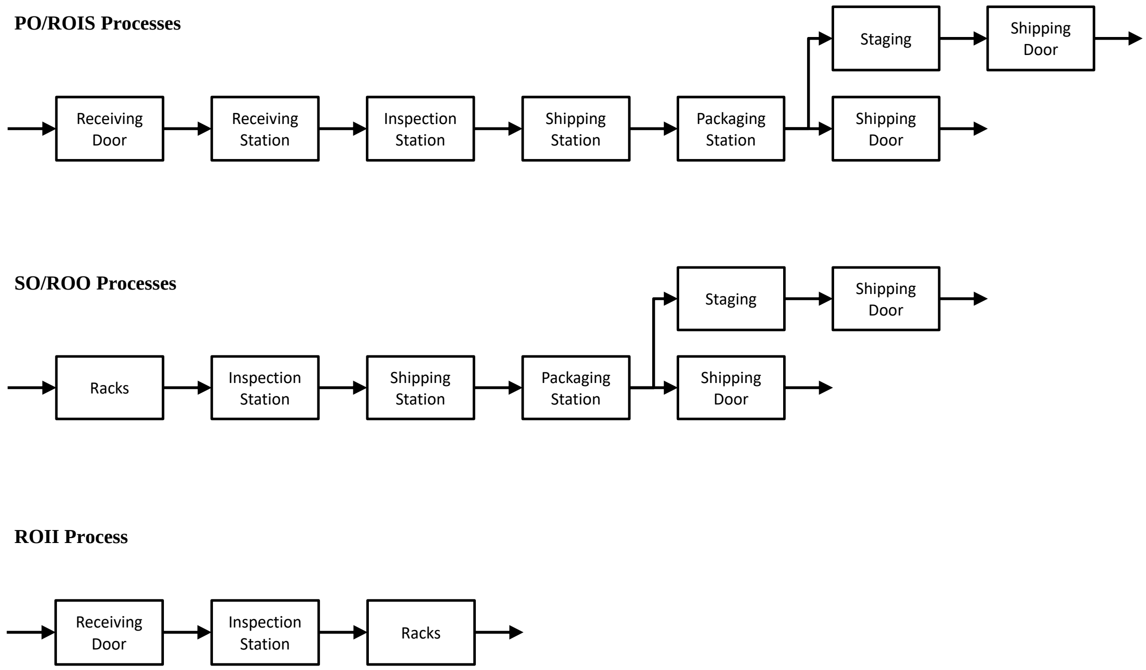


Figure 49 – Processes flows of the various processes present in the system

Going forward the PO/ROIS and SO/ROO process groupings are leveraged to reduce the dimensionality of the problem and provide a more aggregated approach to evaluating the performance of the system. This analysis was required to enable the processes, process flows, and characteristics thereof present in the system to be defined as part of the first step of the LIVE methodology. By the methodology requiring such definition, it effectively facilitated a better understanding of the operations and the process flows present as a by-product. This improved understanding constituted the first of the benefits to be yielded from applying the LIVE methodology to this problem.

8.1.1.1 Baseline Operational Characteristics

Having since observed the general characteristics of the system, those of the current operations are presented. From a human resource point of view, not all twenty-one workstations are manned in the current state of the operations. Eight of the twelve inspection stations are manned; two of the four receiving and shipping stations each are manned, along with the packaging station for a total of 13 actively manned stations. Each station consists of a single worker. Current operations also do not leverage independent material handlers and thus, the workers of each station must retrieve and pass along parts after processing occurs at the worker's station.

In terms of the processes, all five have their volumes (i.e. parts per day) distributed across each of the manned stations. As such, the process flows observed before in Figure 49 are only representative of the general process flow. Each of these becomes enumerated based on the stations present and manned. For example, for the cross-dock processes (PO/ROIS) the number of unique process flows in which parts of this type can take as they pass through the system is totalled at 64 provided the current stations manned. This total is achieved by multiplying the number of receiving stations (2) by the number of inspection stations (8) by the number of shipping stations (2) by the packaging station (1) and finally by the staging vs. non-staging split (2). This latter split constitutes parts that are oversized and need to be staged until pickup. These parts and thus process flows are then assumed to need forklift handling as opposed to a manual handler. This enumeration is present across each of the three condensed groups of processes provided the stated distributed operational approach currently deployed.

8.1.2 Layout Concepts Considered

After observing the three distinct groups of processes (PO/ROIS, SO/ROO, and ROII), three unique operational concepts were formed for subsequent analysis.

8.1.2.1 Concept 0: Baseline Layout Configuration

In the first of these concepts, the current baseline configuration is assumed to remain unchanged over the planning horizon (to be defined in the proceeding section) and thus, only a pure analysis of the layout design going forward would be tested (i.e. optimization was not required). This approach enables a baseline layout design performance to be established, whereby others can then be compared to it to establish either superiority or inferiority. This layout configuration was presented before in Figure 48.

For this baseline configuration, two process distribution approaches are considered. The first is the current one, whereby the process volumes are distributed across all manned stations. This approach effectively represents the case where nothing with the current operational structure or layout is altered over time. The second approach considers splitting the distribution on a cross-dock, rack process-basis. In other words, certain manned stations are dedicated to each of these process groups. For example, with two receiving stations manned, one would be dedicated to processing the PO/ROIS cross-dock process parts while the other would only process SO/ROO/ROII rack-process parts where necessary. This approach was formed after observation of how the two process flows were uniquely distinct from one another. Again, for both approaches, the layout design is assumed unaltered over the entire planning horizon and remains identical to the baseline configuration presented before.

8.1.2.2 Concept 1: Current Operational Space, Revised Operational Strategy

The second concept considered is a derivative of the second approach outlined before. Like this approach, the workload is distributed on a cross-dock, rack process-basis. Contrary to before however, in Concept 1, the layout design is not required to remain fixed across the horizon. As such, in this concept the impact that layout restructuring has on performance is consider and this is achieved through optimizing the layout via the proposed bi-model multi-stage solution approach. It is required however, that the operations remain located in the general area in which they are currently located in the warehouse space. Additionally, to reduce the complexity of the problem, the space was refined to just the area starting at the rack region in the baseline configuration and spanning width-wise to the start of the top-right offices. In reducing the space, the Building A, receiving, and shipping doors were appropriately relocated to be well-representative of where flows would intersect this refined area. This layout space is provided below in Figure 50 such that a clearer understanding of this setup can be gained. The bottom-left station in this space is the relocated Building A door and then working rightward along the bottom edge, the shipping door then receiving door. To the left of the left boundary lies the storage racks of the original layout and to the right of the right boundary, the sales offices.

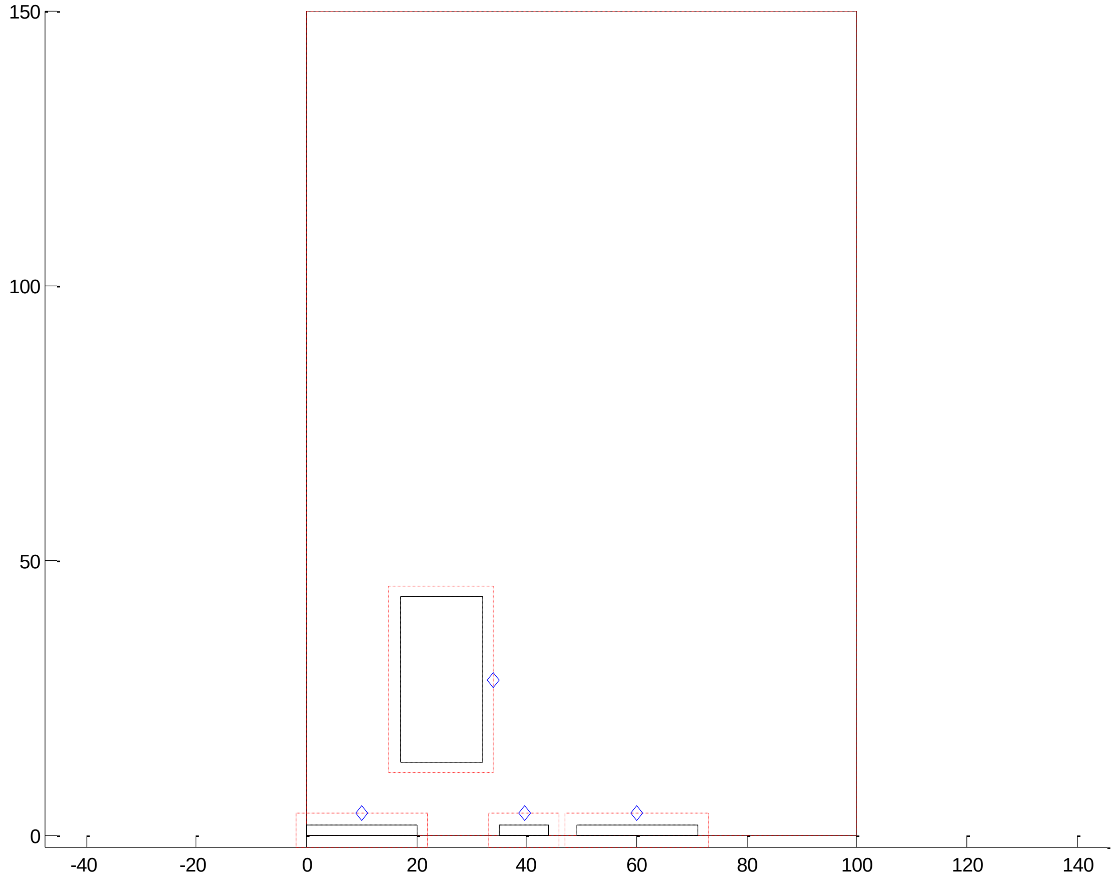


Figure 50 – Concept 1 layout boundaries and setup

8.1.2.3 Concept 2: Revised Operational Space, Revised Operation Strategy

In the third and final concept considered, Concept 2 entertains more than just a splitting of the process distributions on a station-basis. In Concept 2, the operations are split on a layout area-basis as well. Based on the unique difference between the cross-dock and rack processes, it was believed that these two groups could be segregated in the available space to provided improved operational flow. With the rack-process originating from the Building A door in the bottom-left corner of the warehouse area and ending at the shipping door, also at the bottom of the warehouse area, any motion north of the rack door can be viewed as wasted motion. In the current operational landscape, all parts of these rack processes must traverse across and up the space before then backtracking down to the shipping door. This observation motivated the consideration of this approach, whereby this wasted motion could be eliminated by relocating the rack-processes to the currently unutilized space present at the front-left of the warehouse. To enable efficient flow, another receiving door, also currently unutilized at the bottom-left corner near the Build A door, was to then be utilized as part of this operational concept. At the same time, the cross-dock processes were assumed to remain in the current operational area such that an up and back motion for the PO/ROIS processes remained.

To study this concept, the layout was then split according to this premise, whereby the rack-processes and associated stations would be optimized in the front-left white space of the warehouse while in tandem the cross-dock processes and associated stations would be optimized in the current cross-dock area. This split is represented below in Figure 51 and Figure 52 which represent the cross-dock and rack-process layout setups respectively. A synthesis of these two layout spaces, and associated layouts, produces the overall operational performance of the system and layout design. This concept directly considers the costs of relocating the stations from the current cross-dock space to this new unutilized space in the front of the warehouse. The cross-dock area as shown in Figure 51 is similar to that of the one tested in Concept 1, except that now the rack door is removed (as no cross-dock processes include it in their process flows) and in place of this removal, the left boundary is shifted in toward the backside of the staging area to refine the space and reduce the difficulty of the problem solution. As can also be observed in the rack-process layout, Figure 52, the staging area in the rack-process layout was assumed to be located near the receiving and shipping doors along the right management office wall (right of this boundary).

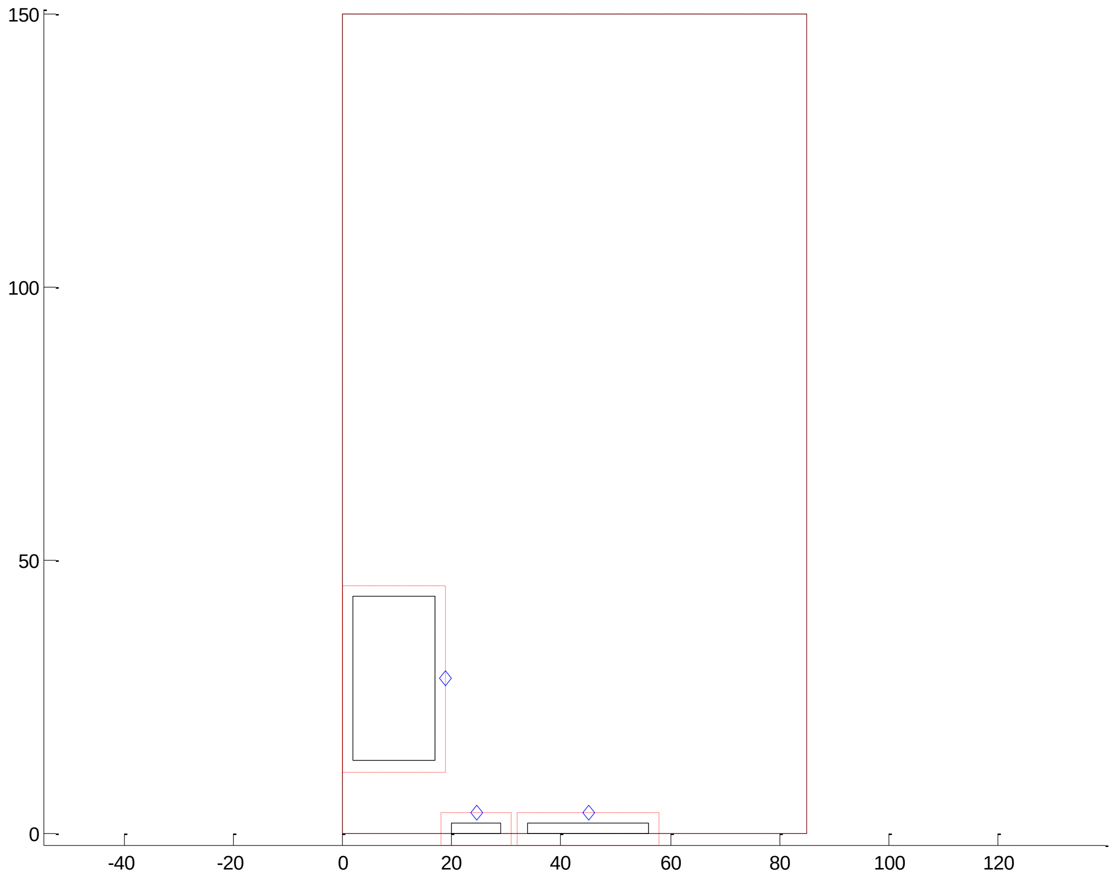


Figure 51 – Concept 2A cross-dock process layout boundaries and fixed stations

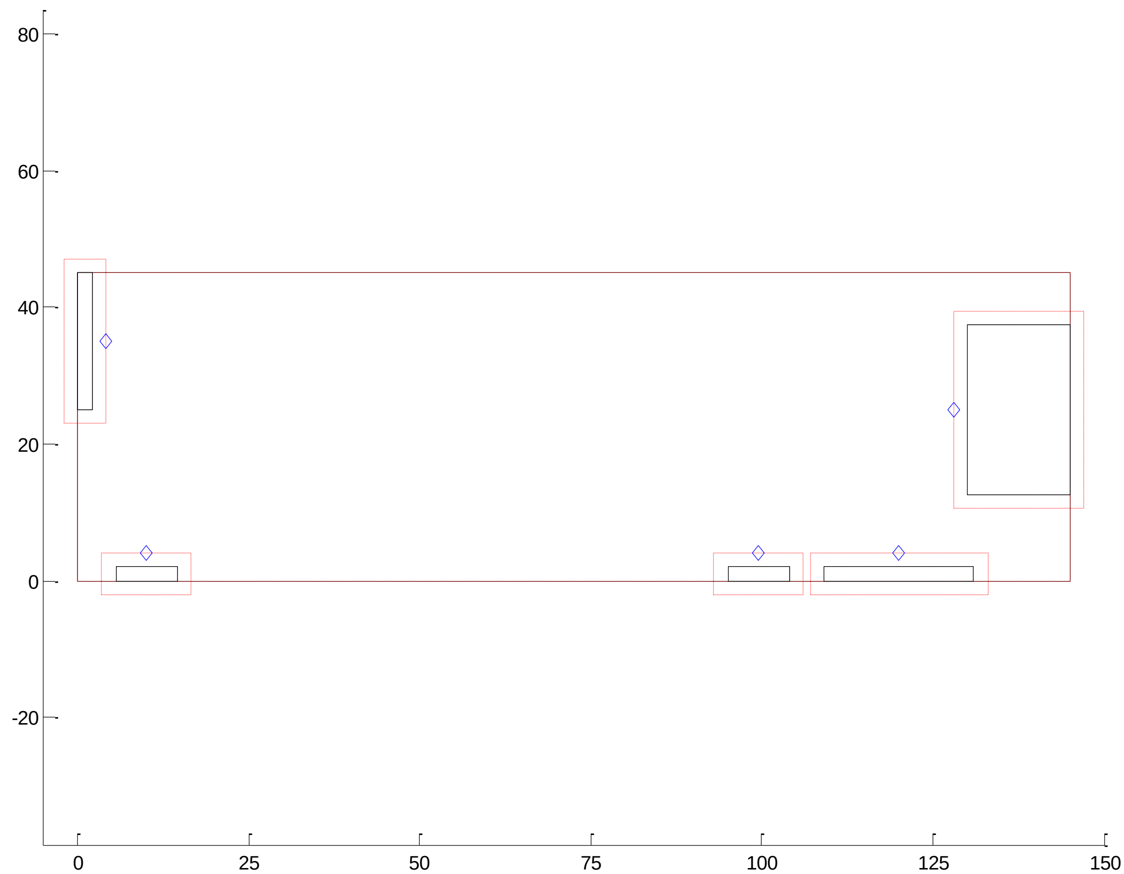


Figure 52 – Concept 2B rack process layout boundaries and fixed stations

Each of these defined concepts and the associated process distribution approaches are summarized below in Table 38. Each row in the table can be thought of as a unique set of problems. In Concept 0 there are two, one with distributed processes and one with split. In Concept 1 a single set while in Concept 2, there are two, one for the cross-dock processes layout and one for the rack processes layout. As a reminder, in Concept 0, only an analysis is to be performed whereas in the others, an optimization is performed for each of these problems of the set to be defined next.

Table 38 – Process distributions vs. concept

Concept	Processes	Approach
0	PO, ROIS, SO, ROO, ROII	Distributed
	PO, ROIS | SO, ROO, ROII	Split
1	PO, ROIS | SO, ROO, ROII	Split
2	PO, ROIS SO, ROO, ROII	Split

8.1.3 Business Model and Market Conditions Examined

With the physical layout properties defined and process compositions established in each, a good portion of the scenario problem definitions have since been established. In addition to these properties, and as one may recall from the problem initialization step of the LIVE methodology outlined in Section 4.1, additional properties still require definition before complete scenario problems can be formed. The major one that remains undefined is the scenario structure. While for this study a fixed planning horizon structured in 12 month forecasting intervals and spanning 36 months (i.e. 3 years) was deployed across all scenarios, the same was not true for the restructuring schedule. This was the first of the business model conditions in this study considered as a design factor. A variety of different restructuring schedules were considered. Among these, static single period layout designs were considered along with two period and three period dynamic layout designs. In addition to these restructuring forms, different implementation timings of these restructures were also considered. Those considered in the study are outlined in

Table 87 provided in Appendix G, Section G.7.2 whereby the timings were chosen in a mostly arbitrary manner.

Two additional human resource related design factors were also considered in this study when defining the scenario problems. The first was the manned stations. This design factor directly considers the addition of labor at the stations over time and thus enables the study to examine how the work force ought to evolve over time. Moreover, when coupled with the restructuring schedule, the timing of these additions can be assessed. Several station manning options, or evolutions, were considered in this study. In all considered, the number of manned stations grew to align with the assumed growth in production over time; however, the distribution of the manned stations across the two process groups (PO/ROIS and SO/ROO/ROII) varied over these different options examined. Given these properties are defined on a period-basis, the evolutions needed definition across the three distinct forms of the restructuring schedules considered. Note that these evolutions were established such that consistency was maintained for the most part for the options across each restructuring option. That is in the sense of distribution between the two process groups and moreover this distribution over the planning horizon. These options span considerations such as an equal distribution of stations as well as skewed options whereby more active stations are assigned to each of the two process groups. This was done to demonstrate, when in conjunction with the PPD and the distribution options studied, the system can become station constrained, whereby there is not enough station capacity to maintain production levels. This then effectively reduces the profitability of the system, as will be observed later. Again, the manned station distributions studied can be found in Appendix G for reference (

Table 87).

In addition to this design factor, the number of material handlers available to move parts throughout the space was also considered. This consideration constituted a major change in the existing operations whereby no dedicated handlers were present to move product. Instead, the workers at the stations were responsible for this duty. This consideration deploys the industry concept of a water-spider, whose sole duty is to facilitate the movement of the product between stations. For this design factor, a couple different handling options were considered. Definition of the required handlers was defined after observing the utilization levels of the baseline configuration during initial testing. In order to demonstrate how handler availability or capacity can impact the design and profitability of the system, these options were chosen strategically such that in some situations, when coupled with a high PPD option, the system may become material handler constrained. In turn, the profitability would then suffer, as will be observed later when the results of this study are analyzed. This design factor, along with the manned station factor, restructuring factor, and operational concepts discussed before, compose the business model conditions considered in this study.

In conjunction with these business model conditions, several market conditions were also considered as design factors in this study. The most notable of these were the year-over-year parts per day (YOY PPD) and distribution option factors. The coupling of these two, establishes the production rate to be considered in each scenario problem. Three YOY PPD options were considered ranging from 60% to 80%, which were established based on observed projected increases in the operations of the system since having been established just three years ago. The starting PPD was set to be 30 PPD, matching the current average volume of the system. Several distribution options were also considered in parallel. Since the PO and SO processes alone compose 95% of the current operations, and this is likely not to change going forward, this percentage of the PPD associated with these two processes was maintained across the horizon. With the current operational distribution being 65% PO and 30% SO, this distribution was maintained as the starting distribution for all distribution options considered. One option considered this ratio maintaining over the horizon while the other two options examined a shift in operations towards SO related parts becoming more prominent. This was done as the goal of the firm was to begin leveraging e-commerce to boost profitability. This shift only impacts SOs, which is why the remaining two options considered in the study examine two different shifts in this distribution, one being more aggressive over this three-year planning horizon than the other. One considered reaching a roughly equal split at 45/50 come year three while the other considered a more dramatic 25/70 shift, both in favor of more SO related parts being processed per day. Coupling these redistributions in process volumes with the human resource design factors of before enables the implemented dynamic production adjustment technique to become active in situations where production cannot be sustained provided the combination of YOY PPD, the number of active stations available, and handler availability. The exact distributions and their evolutions across the planning horizon for these distribution options may be found in Table 92 in Appendix G, Section G.7.2.

In addition to both these factors, the coefficient of variances for the two process groups, PO/ROIS and SO/ROO/ROII, were also considered as design factors. Both these factors are directly associated with the local robustness method implemented and thus directly enable robustness relative to the production uncertainty of each of these to be considered. While only a single level was considered for the PO/ROIS processes and set at 10%, two were considered for the SO/ROO/ROII processes and set at a 10% or 20% YOY increase provided the higher variability associated with the success of the firm in the e-commerce space. Both factors' applicability depends on the design choice factor considered. One choice appropriately defines the percentile ranges to consider just the nominal production levels and thus the system's performance in the absence of uncertainty. This option, if one recalls, effectively renders these coefficient of variance terms irrelevant provided how the local robustness method was implemented in the performance model developed as part of this dissertation. The other option considered robustness over the percentile range of 30 to 70%, whereby the coefficient of variance factors then become relevant as robustness relative to these production uncertainties are considered when optimizing the layout design.

These factors and their variability (options) are summarized in Table 39, whereby the business model related conditions, market related conditions, and single design decision are grouped to provide complete closure for the reader on these noted factor distinctions. Once more, for those factors with options presented in Table 39, one may refer to Appendix G, Section G.7.2 for the expanded definitions of these factor options.

Table 39 – Factor table of conditions considered in the case study

Factor		**Levels**				
Condition	Description	Level 1	Level 2	Level 3	Level 4	Level 5
Business Conditions	Concept	0	1	2A	2B	
	Restructuring Option	1	2	3	4	5

Table 39 (continued)

	Station Manning Option	1	2	3	4
	Handler Option	1	2		
Design Decision	Percentile Option	(0.5, 0.5)	(0.3, 0.7)		
Market Conditions	YOY PO/ROIS C*v* Increase	10%			
	YOY SO/ROO/ROII C*v* Increase	10%	20%		
	YOY Parts/Day Increase	60%	70%	80%	
	Distribution Option	1	2	3	

A full-factorial design of experiments (DOE) was leveraged to cover the entire design space and consider all combinations of these conditions when establishing the scenario problems. In total the number of unique business, market, and operational concepts examined was 3,312. In other words, 3,312 unique scenario problems were considered (solved and subsequently examined) in this study. Each experimental trial of the DOE defines, when coupled with the process and physical layout definitions from before, the necessary input properties to completely define a scenario problem. Note, this complete scenario set quantity is less than what one would compute leveraging the levels provided above and for a full factorial design (5,760 in total). The difference is accounted for by several filters being applied. First, Concept 0 only applies when restructuring option one is chosen, the remainder apply across all restructuring options however. When the design choice is set to level one, the YOY SO/ROO/ROII C*v* increase factor only needs be executed for a single option due to a non-robust evaluation being performed. Furthermore, since Concept 2A (the PO/ROIS layout) does not consider SO/ROO/ROII related processes (Concept 2B does this) it does not need to be evaluated for both

SO/ROO/ROO C*v* increase factor levels. These filters effectively reduce the number of scenario problems that require solution from 5,760 to the noted 3,312.

For the remaining properties leveraged in this study to define the scenarios, one may refer to Appendix G, Section G.7.2 for these property definitions and any accompanying assumptions made regarding their definitions. With that said, one notable property definition, which requires definition here is that of the market values per part. Since such information was not available, it was assumed that on average a part yields a market value of $75 and moreover, this is consistent across all part types (PO, SO, etc.). Though a rather significant underestimation provided the parts being processed by the system (aerospace parts), this assumed market value was enough to outweigh the observed average manufacturing costs per part in the system. This enabled a positive profit margin to then be observed. Note, all subsequent results are based on this market value assumption. Moreover, with it well understood that the average market values for the studied system are far more significant than this assumed value, the results that are to be demonstrated and the differences observed would only become more extenuated with a larger, better representative, market value input provided to the performance model.

As for the optimization and analysis parameters, these have since been established according to the results of the previous experiments performed. Boundary constraints are considered hard in this study while budgetary constraints are soft, allowing for debt financing to occur when required to enable a restructure to occur. At this juncture, the physical layout properties, processes, market and business model conditions, optimization parameters, and analysis options have all been defined across all the scenario problems considered in this study. This complete definition of the scenario problems, and thus scenario set to examine, concludes the first step of the LIVE methodology's application to the problem. The results of the second (solution) and third step of the methodology (performance model analysis) are presented and discussed next. In yielding the results of this study, a few noteworthy observations were made regarding the performance of the developed FSPPM. The observations and actions taken in response are highlighted in Appendix G, Section G.7.1.

8.1.4 Down-Selection of Business Models to Examine Going Forward

Following these actions, effective solution to the scenario problems defined before was then possible. As such, Stage One solution of the scenario problems was then performed leveraging the optimization parameter sets established earlier in Experiment 4.

As noted before, the first stage of the proposed bi-model multi-stage solution approach can act as a way of providing a more rapid formation of the design space all while proceeding towards the final solution to the individual scenario problems. Once formed, the design space can then be evaluated before proceeding into the second stage of solution. This capability of the LIVE methodology, and the solution approach it deploys, was leveraged when performing this study. The design space was able to be previewed and suitable business models, which maximize profit, identified for further consideration going forward in the design process.

This down-selection is advantageous as some business models (i.e. synthesis of business conditions factors considered and discussed before) will consistently be inferior to others and as such do not need to be considered going forward in the design process. Business decisions (i.e. design factor choices in this study) such as hires, restructures, process changes, etc. can be thought of as forming a decision tree, whereby each branch represents a different business model and combination of decisions. Each of these branches will have a length representative of its performance across the market condition factors considered in the study. Those branches that are consistently longer, i.e. better performing, can be viewed as those falling on the pareto optimal frontier. It becomes productive then to prune, or filter out, those business models that fall inside this frontier as they will consistently underperform the others from a profitability standpoint for each of the market condition forecasts. To be capable of performing this down-selection, an intimate understanding of the design space was first required. The results of the Stage One solution to these scenarios and this design space exploration follow.

8.1.4.1 Design Space Exploration

Before the performance results could be compared, the concepts needed to first be post processed, whereby Concepts 1 and 2 were converted back to the original warehouse layout, not the refined areas they were optimized and evaluated for. This was necessary so that a direct comparison could be made between them and the Concept 0 results. The conversion did not change the relative positions of the workstations; rather it only changed the relative position to the doors in these concepts, thereby effectively altering the material handling distances and associated costs. In the case of Concept 2, only Concept 2A required conversion as Concept 2B had the doors located in the same locations they would be in the otherwise converted warehouse area. Moreover, while post-processing, the advanced flow distance method was applied to each of the found optimal designs of the Stage One algorithm to provide a better comparison going forward when such an evaluation would then occur in Stage Two to further optimize the layout designs. Finally, the performance results of Concept 2A and 2B were combined by summing across the performance metrics apart from the utilization levels, which were averaged provided these are unique to the individual layout and those stations. As such, they cannot be justifiably summed like the other metrics to form the Concept 2 results.

8.1.4.1.1 <u>Holistic Overview</u>

With the results post-processed for comparison purposes, the design space was then examined first by leveraging the scatterplot matrix presented in Figure 53 of the relevant nominal system metrics versus the various business model and market conditions considered in the study. These results observe the nominal performance of the designs encapsulated in the solution of the scenario problems. The results of the robustness metrics and relevant design factors follow many of the same trends observed here and thus are omitted to avoid redundancy. Blue dots represent Concept 2 results, green correspond to Concept 1 results, red to Concept 0 with a distributed process approach, while purple indicate results for Concept 0 with a split approach.

As can be observed in Figure 53, strong positive relationships exist between YOY PPD and that of the profit, direct labor costs (DLC), retained earnings (RE), and direct retained earnings (RE DIRECT) metrics. This is to be expected given that as production volume increases, more direct labor would be required to sustain such workloads and more production equates to higher profits and thus retained earnings. Additionally, as production rises, indirect labor costs shift to becoming direct labor costs, which explains the negative relationship observed between that of the indirect labor costs (ILC) and YOY PPD. This trade will become a common theme of discussions going forth and furthermore, becomes a key factor is establishing the most suitable, i.e. best, business model going forward. Also related to the ILCs, it is evident that Concept 1 (green) consistently has lower ILC in comparison to Concept 2 (blue). This is not because Concept 1 is more efficient however; it is quite the contrary. Concept 1 consistently has lower ILCs because overall the layout designs are inferior from a material handling perspective. Large handling distances correlate to less idle handler time and therefore lower ILCs but higher DLCs in return.

As also expected, provided the setups of the concepts, Concept 2 incurs the greatest amount of rearrangement costs (RC) while Concept 0, which remains fixed at the current configuration, experiences the least at zero dollars. Concept 1 falls between these two concepts. The widening spread in the data points with increasing restructuring option can be attributed to both more rearrangements being considered and thus higher RCs as well as the interplay between these higher restructuring options consisting also of associated station options and handler options that change more frequently over the larger number of considered rearrangement periods (both physical layout and operations). This interplay between these factors will become important going forward in understanding the behaviours of some of the later results. In doing so however, great insight can be derived as a by-product, as will be shown.

Also, more green dots (Concept 1) appear at the low end of the DLCs. This does not indicate that Concept 1 is preferred from this stand point. The reason for this is as follows. These cases are those in which, because of how Concept 1 is configured, the production level is being capped by insufficient handler availability, which effectively decreases the production level and thus DLC and in turn profit. The appearance of these at the low end of the profit metric confirms that these low direct labor cost cases are simply those limited by capacity restrictions. Such a case will be presented next.

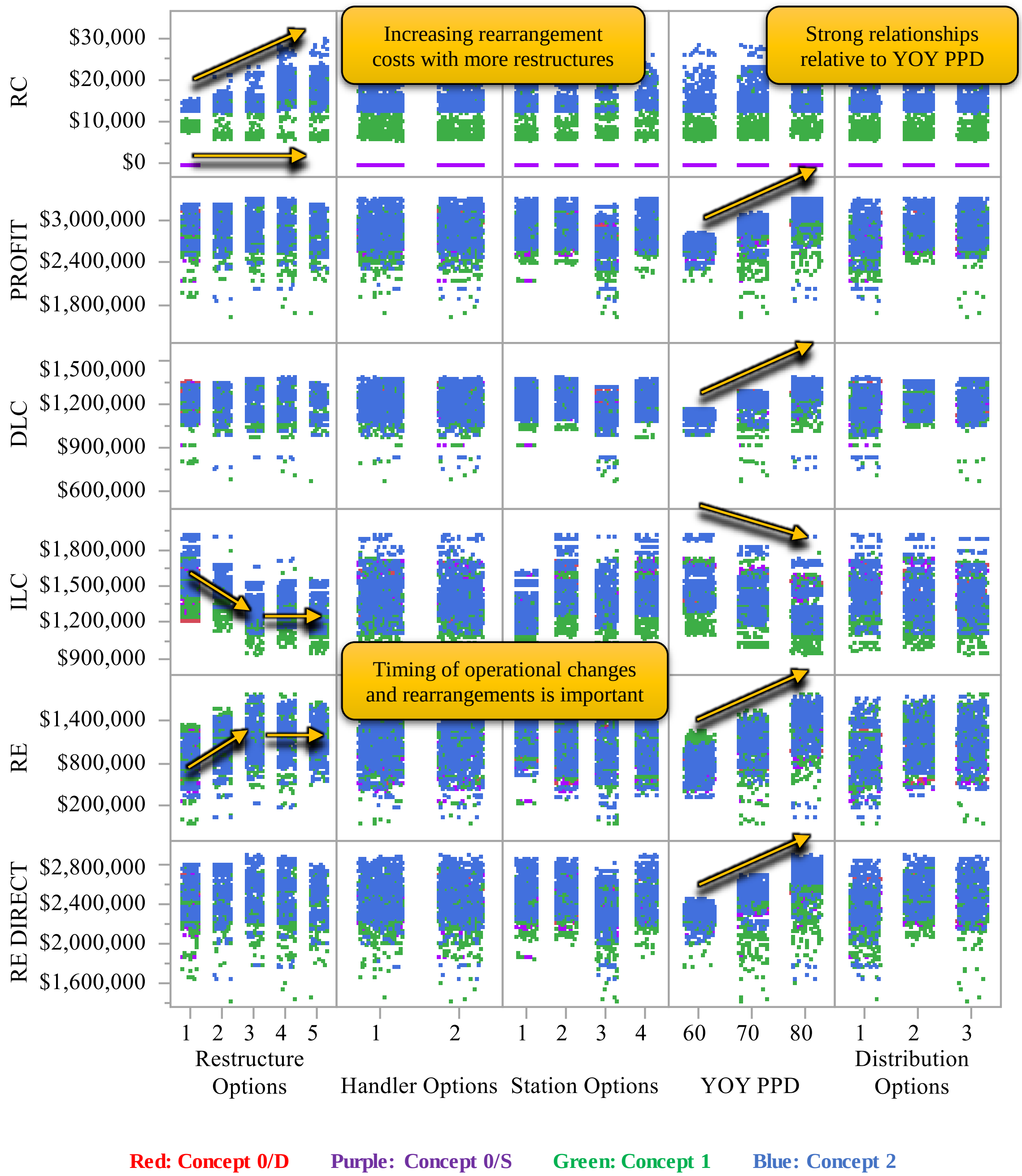


Figure 53 – Nominal system metrics versus conditions

8.1.4.1.2 Utilization Constrained Examples

Having previewed the design space holistically, a deeper dive into the scenarios and some of these behaviours observed before was warranted in order to gain a deeper understanding of the interplay between the business conditions, market conditions, and the layout design itself. In doing so, improved insight into the design problem could then be gained and further value could be derived by deploying the LIVE methodology to solve such a layout design problem.

The first scenario examined was that of handler constrained case. In this case a two-period layout design structure was considered (restructure option two) with the higher handler option deployed, evaluated for a YOY PPD of 80% (highest level) and with the lowest manned station option of 10 inspection stations (5,5 split), and 4 receiving/shipping (2,2 split). Despite this lower active station quantity, the system is material handler constrained. This is demonstrated in Figure 54 below, whereby the utilization level of the handlers (blue bar) has reached 100%, fully utilized by time the end of the planning horizon was reached (Month 36). At the same time, the other stations maintain a margin of available capacity. The inspection stations, to no surprise provided their higher processing times, follow closely at 90% utilization. While the utilizations are relatively high at the end of the planning horizon considered, all stations and handlers are substantially underutilized in the earlier stages of the planning horizon (first year or twelve months of time). Only after the first twelve months do they then experience a more dramatic up rise in utilization as a result of the continued upswing in PPD. In this case, the restructure occurred at the twelve-month mark and came with an increase in active stations, which is why at that point the utilization did not jump then but did afterwards at month twenty-four. This upswing in PPD, but then no further increase in active stations at month twenty-four, lead to the ratio between the stations and handlers shifting from being overly idle (60% of the time) to becoming more active (over 80% in some cases). As one can image, underutilization in the earlier months leads to high ILC experienced early on with these ILC then shifting to DLC latter in the horizon. A more optimal strategy would then be to reduce the number of active stations earlier on when the PPD is lower, and then gradually increase as PPD increases downstream. This would keep utilizations higher earlier on, thereby avoiding high ILC, which only diminish the profitability of the firm. Care would need to be taken to avoid getting behind in adding active stations (i.e. workers) as it would otherwise then lead to a capacity limited system which would also diminish profitability.

Figure 54 – Handler capacity constrained scenario

Another interesting case observed while investigating the design space was the one provided in Figure 55. In this scenario, a three-period restructuring design was considered along with the highest station option, a YOY PPD of 70%, the mid-range shift in PO/SO process distribution levels over time, and the lower handler availability option. This case is interesting as it demonstrates a scenario whereby the system shifts from being capacity constrained by the receiving station to then being handler constrained as a result of changes in the operations going from period two (Month 18-24) to period three. This is a unique example of the interplay between these different design factors. Entering the last period, there are too few receiving station to meet the production demand while there remains just enough handler availability to do so (97% utilized). Once the new hires were made, and therefore new receiving stations became active, the system switched from having such stations as the bottleneck of the system to then the handlers becoming the bottleneck as the PPD continued to rise at a rate of 70% YOY.

Figure 55 – Example of the system bottleneck changing

The next scenario, provided in Figure 56, is important to discuss as it is the first observed that is inspection station constrained. In this case, the inspections stations become the bottleneck of the system despite station option four being deployed, which by all means considers the greatest number of active stations along with option two. The question becomes why are the remaining stations well underutilized while the inspection stations are completely utilized, thereby limiting the production level sustainable by the system?

Recall, that for this Concept 1 example, the utilization levels are the mean values between the two distributed process groups. Since distribution option one was considered in this scenario, effectively indicating the process mix would remain at a 65% PO / 30% SO split in volume. The majority of the PPD, increasing at a YOY PPD of 80% in this case, were then of the PO type. Station option four, while considering the greatest amount of active stations, also considers allocating more of these to the SO group. This effectively pairs the highest PO PPD case with the lowest number of PO active inspection stations. As such, there are far too few active stations in the PO/ROIS line to sustain the high level of production considered for the scenario. This is thus a perfect example of how matching distributions of active stations and the process distribution mixes becomes critical to a well performing system and overall design.

Figure 56 – Inspection station capacity constrained scenario

The final constrained case examined is one of the most severe cases observed while investigating the design space. In this case, a similar scenario to that just described was encountered except, the bottleneck was then that of the receiving stations. In fact, throughout the planning horizon they were the most utilized and therefore bottleneck in the system throughout. In Figure 57, the considered PPD for the scenario, or PPD expected, and the actual PPD are overlaid on top of the utilization bar chart so as to directly observe the implemented dynamic production adjustment technique at work and moreover, the relationship between these utilizations and the PPD that can be processed by the system. As one can see, due to the station decomposition and restructuring schedule considered in this business model coupled with the distribution of the process volumes and YOY PPD of this scenario, the system became capacity constrained within the first eighteen months of operation. In this case, the system was constrained by an insufficient number of active receiving stations to sustain such high production volumes.

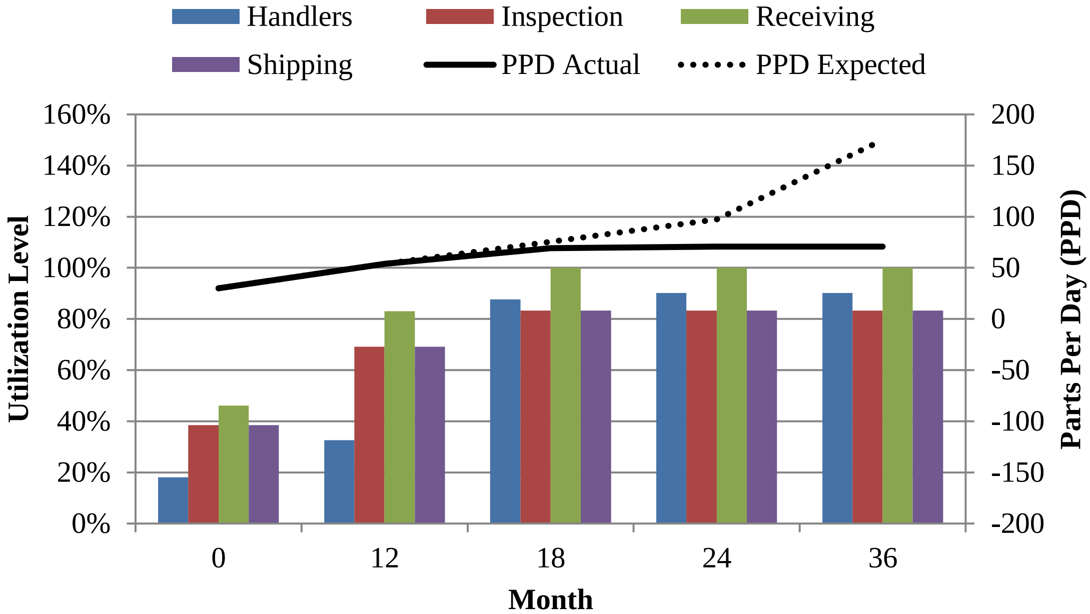


Figure 57 – Example of a severely capacity constrained system

Becoming capacity constrained so early on also severely diminishes the profitability of this business model. As can be observed by the lines of expected PPD and actual PPD, once the receiving stations reached 100% utilization the actual PPD that was sustainable deviated from the expected. At that point, it became constant across the remainder of the horizon given no further changes in active station quantities was to be considered in this business model beyond this point in time. One may consider that the area between the two PPD lines indicates the lost production volume experienced by the system by being capacity constrained by the receiving stations. The receiving stations are not alone in being over-utilized. All other stations and handlers are well above 80% utilized. Given the 80% YOY PPD increase of this scenario, in order to achieve the expected PPD at month thirty-six, all stations would have been over-utilized by that point, not just that of the receiving stations. As such, adding just receiving stations would only have the effect of switching which station the system is then capacity constrained by. Given the graph shown, if receiving stations were to then be added in this business model, the system would then just become handler constrained (next highest utilization level at roughly 88%).

8.1.4.1.3 Concept Comparison

Having since acquired an intimate understanding of the design space and the interplay between different design factors and their impact on the performance metrics of the system, a comparison between the various concepts considered was sought to begin to down-select towards the best business model to consider going forward. This was done first by returning to the scatterplot presented before of the design space and examining it further, but now on a concept-basis.

For the nominal scenario conditions (no robustness considered), it becomes evident, by highlighting the Concept 0 cases (purple and red in Figure 58), that this business model is consistently suboptimal across all metrics except for the rearrangement costs (RC). Recall, that Concept 0 cases are those in which the original layout configuration is maintained across the horizon despite changing conditions. As such, this is not at all surprising since all other concepts consider rearrangement. Provided that these other cases outperform that of those associated with Concept 0, it then becomes clear that the rearrangement of the current layout is needed, and furthermore beneficial. It is also evident that such a business model would result in lower profitability for the business. This is can be observed directly by these highlighted cases in Figure 58 clearly falling below what other concept cases can achieve across the conditions in the profit

(PROFIT) and retained earnings (RE) metrics. One should not be confused by such cases producing notably lower ILCs. The reason for this outcome is as follows. Due to the inefficient nature of the layout configuration, material handlers are more active provided the longer material handling distances they must travel as a result. As such, one can expect to see that the lower ILCs are only a by-product of such costs being reallocated to that of the DLC metric due to material handlers being then less idle and instead more active. By inspection, this is in fact observed in the DLC metric. Such a theme was also observed before when the design space was explored more holistically.

Now, the one situation where Concept 0 does come close to competing with the other concepts considered is for that of the first restructuring option. Under this option the other designs only consider a singular rearrangement at the start of the planning horizon. After which, these designs then remain unaltered much like that of the Concept 0 cases. Provided that these other concepts then incur a rearrangement cost, as demonstrated in the upper-left most box in Figure 58, the gain in having then lower material handling costs via an improved layout design is somewhat mitigated by such costs. As demonstrated though, not entirely, they still maintain a marginal advantage from a direct retained earnings perspective.

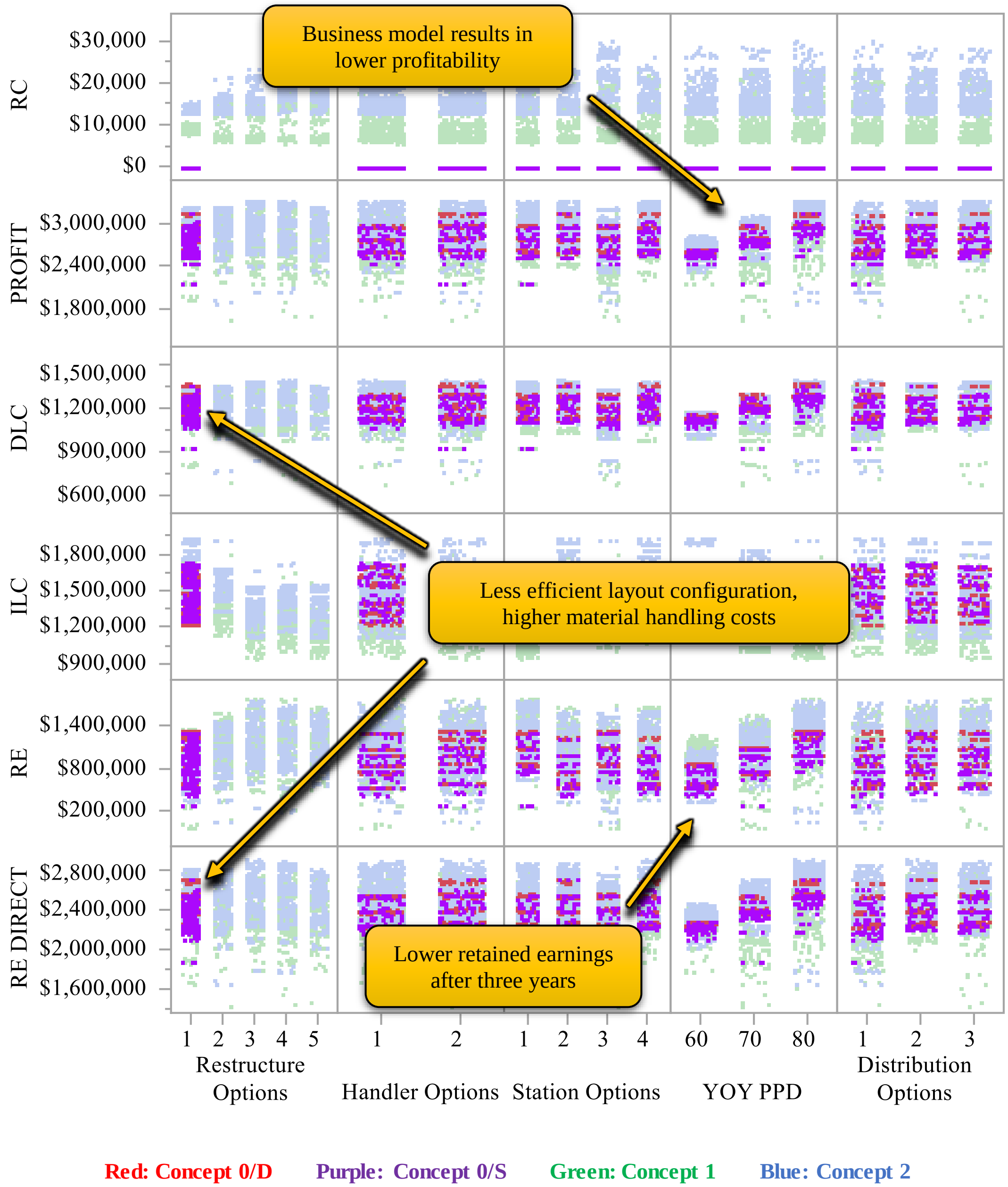


Figure 58 – Performance of Concept 0 across the design space

As for why this performance gap (higher RE, lower ILC, etc.) yielded by other concepts grows as different restructuring options are considered, one must in parallel understand that these restructuring options are also coupled to changes in manned stations and handler availability, as was observed and well discussed before while presenting the utilization constrained examples. As such, this growing difference can be attributed to such changes being more effective than simply maintaining a constant number of manned stations or handlers across the horizon. This outcome demonstrates the need to evolve the system on these fronts over the planning horizon. Moreover, the difference in performances between different restructuring options, whereby the higher options (more rearrangements and changes in human resources) aren't necessarily advantageous, establishes that such operational changes and rearrangements must be strategically chosen and furthermore timed. These are both observations that before would not be observable with the current approaches in the literature to the layout design process. Furthermore, such observations provide substantial insight into the effectiveness of the layout design and additionally the performance of the firm across these different business strategies. *All this insight creates great value for the designer and stakeholders alike during the design process, enabling more informed and data-based decisions to be made and in turn strategies to be formed.* To no surprise, the robustness results demonstrated a similar outcome, whereby the Concept 0 business models were consistently less ideal in comparison to the other two concepts considered in this study.

A direct comparison of the concepts across all other business model design factors and market conditions is provided in Figure 59 on a retained robustness and retained earnings perspective (the main objective functions of the optimization). The chart provides the average performance of each concept across all remaining conditions, both business and market. As demonstrated, Concept 0, to no surprise, is on average the least optimal of the three concepts. While Concept 1 and 2 are comparable, Concept 2 is slightly more optimal, which can be attributed to the redesigned operational landscape it deploys, which enabled it to reduce the material handling costs of the system.

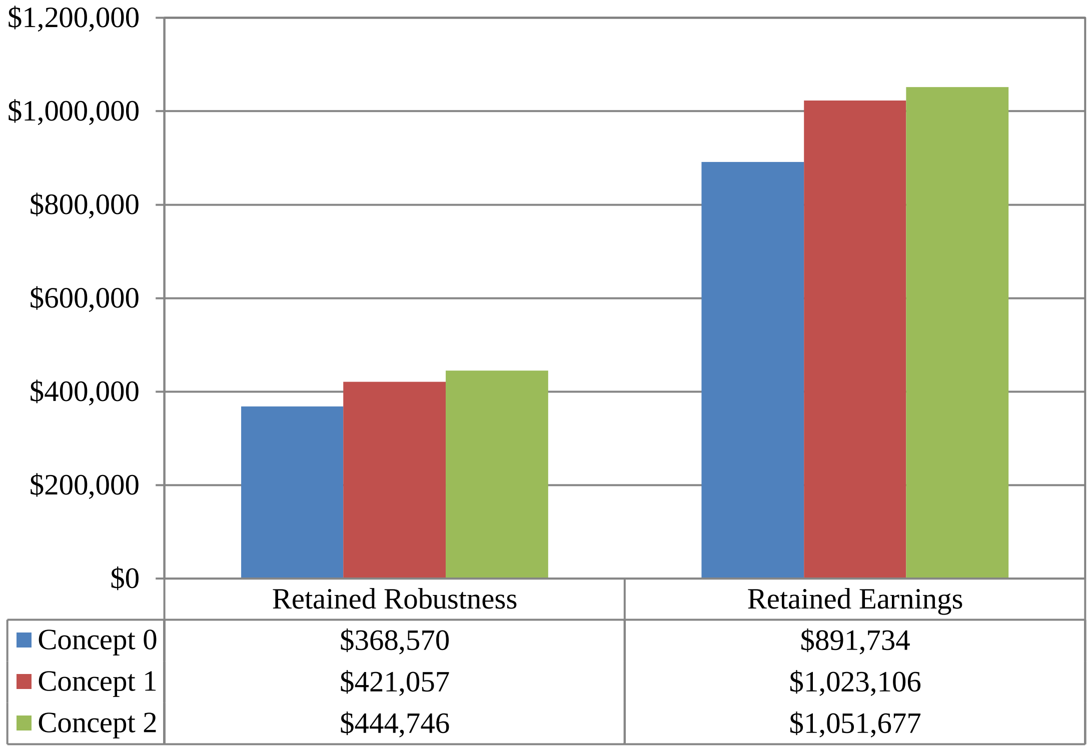


Figure 59 – Average performance of the concepts across all other conditions

Taking these average performance results for each concept and comparing them against each other enabled Table 40 to then be formed. The table is segregated by the diagonal whereby the upper-triangle compares the concepts on a retained earnings-basis while the lower-triangle compares them on a retained robustness-basis. The upper-triangle is read as the cell value indicating the column concept's percent superiority over the row concept while the lower-triangle indicates the row concept's percent superiority over the column concept. As shown, across both metrics, Concept 1 and 2 on average outperform Concept 0 by nearly 15%. In the case of retained earnings, Concept 1 achieves 14.7% higher earnings over the three-year planning horizon when compared to

Concept 0, while Concept 2 achieves even more at nearly 18%. Comparing Concept 1 and Concept 2, one can observe that on average Concept 2 achieves 2.8% greater earnings over the same planning horizon than Concept 1. In the retained robustness metric, this difference effectively doubles to that of 5.6%. Considering these results, it became evident that Concept 2 was the preferred concept of the three.

Table 40 – Comparison of the concept's average relative profitability

		Retained Earnings		
		0	1	2
Retained Robustness	0	…	14.7%	17.9%
	1	14.2%	…	2.8%
	2	20.7%	5.6%	…

8.1.4.2 Final Business Model Selection

Prior observations indicate that the preferred business strategy going forward should consist of an operational landscape described by Concept 2. To establish the remaining strategic business decisions to deploy in the business model (design factor options), a look across the performance landscape relative to the remaining business model conditions was performed to identify the most suitable complete business model, i.e. best performing one, on average, across the market conditions considered. This was performed within this Concept 2 operational landscape since it had since been established as the preferred. To do so, for each unique business model (i.e. combination of business conditions design factors) of the Concept 2 space, the average performances of these potential models across the unique market condition scenarios were then plotted against the remaining business model design factors as demonstrated in Figure 60.

In Figure 60, the robustness results are provided, though note that the nominal results follow suit, leading to the same result that will derived from the analysis that follows. The results are plotted with several of the business models yielding the best RR direct results highlighted. As can be seen, though many of the two and four station options appear at the top of this metric, when translated to the RR metric, they become noticeably inferior compared to other points not highlighted. The reason this occurs is that for these models, they may be effective in sustaining the production levels; however, they are very inefficient when it comes to labor utilization. The noticeably lower RR result, which accounts for indirect labor, indicates that for these models, they effectively are underutilizing human labor, or put alternatively, they have too many workers for what the production level is and as such, too much idle labor. This interplay is important to grasp. As mentioned before, the maximization of human labor utilization (minimizing ILC) without sacrificing production by having insufficient capacity, will be what enables one business model to then stand above the rest and as such, be identified as the best business model to consider going forth in the study.

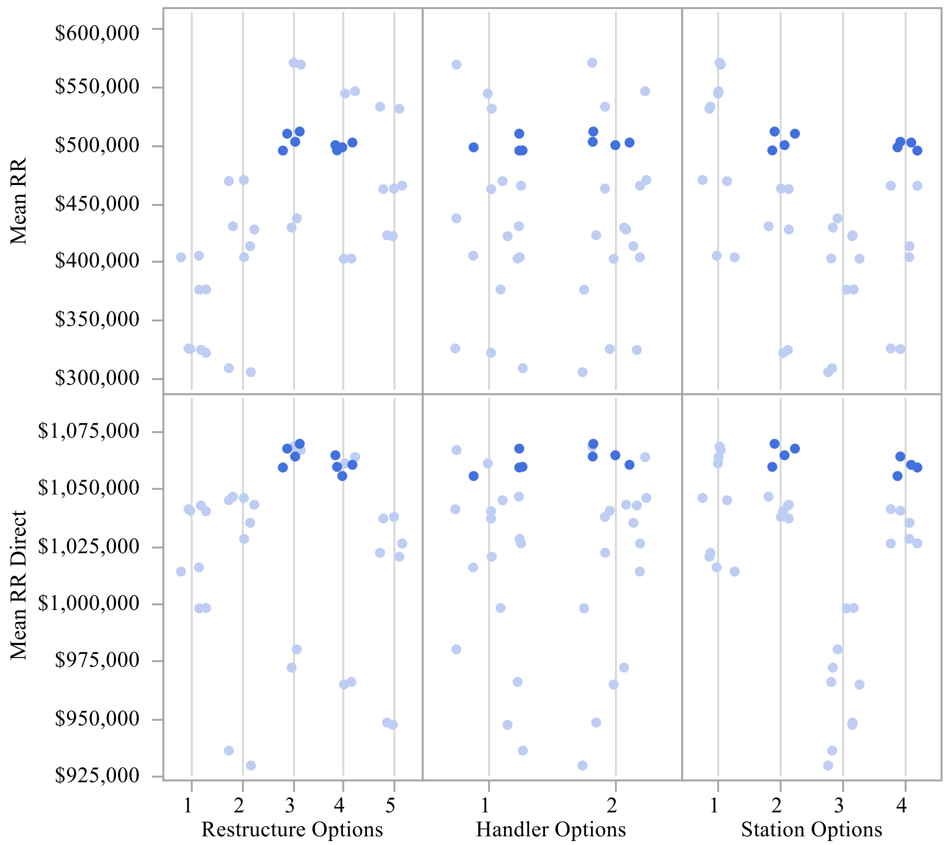


Figure 60 – Concept 2 business model performances

Observing then just the average RR result, the best business model, on average, across those considered was identified as indicated below in Figure 61. As shown, the best two models both deploy restructuring option three, whereby a two-period structure with restructuring occurring at month eighteen (or half way through the planning horizon) is deployed. Moreover, both these deploy the associated station option one for this restructuring schedule deployed. The difference between them lies in the material handler option, whereby it is evident that neither option is all that preferred over the other from a performance perspective (option one is slightly superior). This result mirrors earlier observations where it was observed that, in general, the apex in performance occurred at restructuring option three, with diminishing performance onward at higher restructuring options. This business model was then established as the best model of those considered in this study and recorded as being one to then be consider going forward for eventual solution in Stage Two of the proposed bi-model multi-stage solution approach of the LIVE methodology.

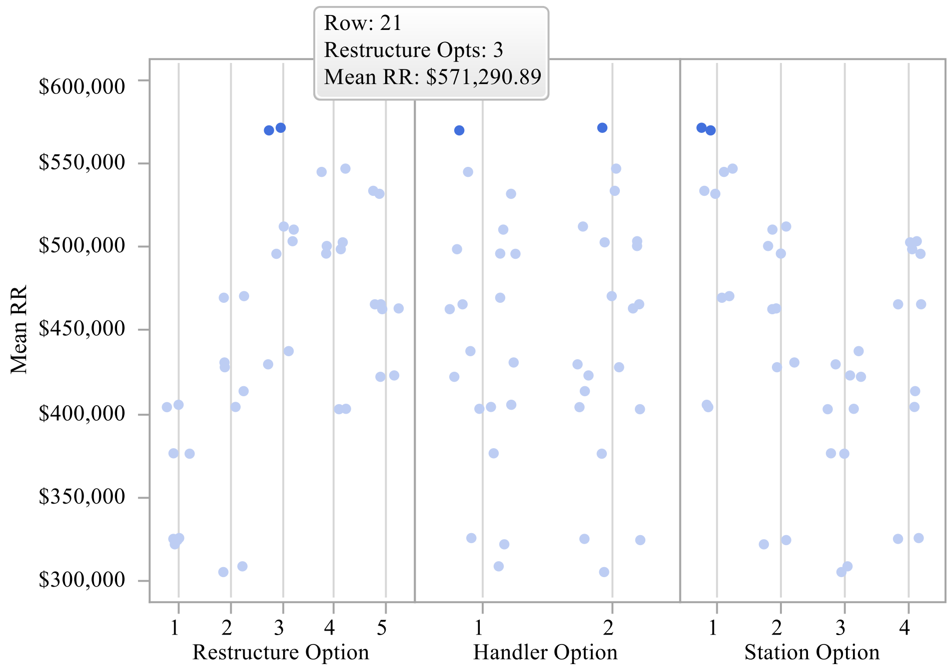


Figure 61 – Overall most robust business model across all market conditions

Though the best design had since been established, a purely static design was also sought, for both comparison purposes and to enable a purely static robust layout design to be further evaluated going forward. Selection of the business models deploying a restructuring option of one, and further identifying the best performing one of those, yielded the result shown in Figure 62. As one can observe, this model also deploys the same handler option and station option of the previously identified best overall business model. These options across the two models deploy effectively an equal distribution of the stations allocated across the two process groups. Provided that the distribution options of the study considered an even split as well as two biased splits in favour of each process group, this result is not at all surprising. Though such an approach may not always be capable of sustaining production levels, on average, it will outperform the others thanks to the human labor utilization being collectively superior. Now it is also clear that the static models noticeably underperform the dynamic ones such as the best one identified before. So why do such models consistently underperform these dynamic ones?

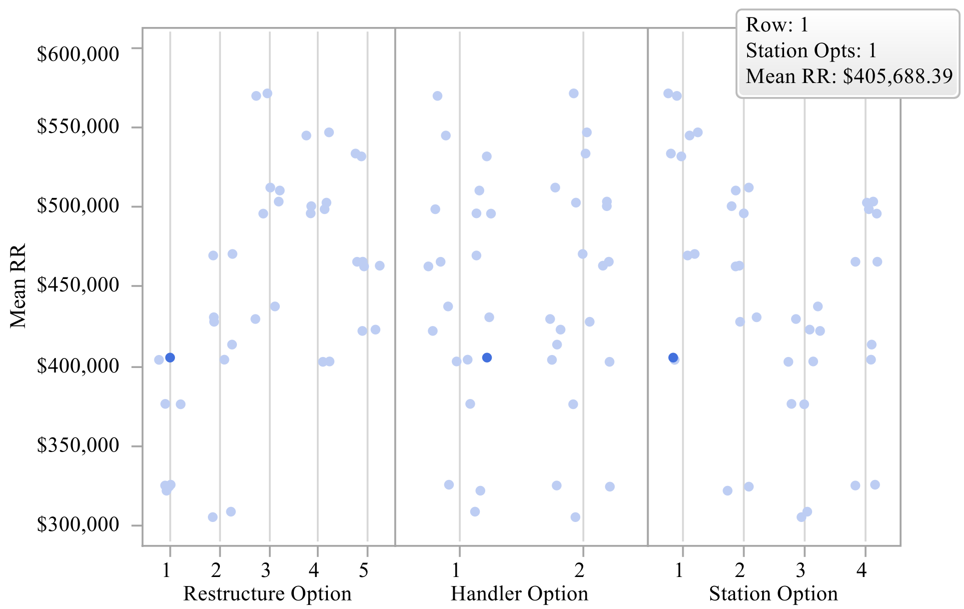


Figure 62 – Most robust static business model across all market conditions

To answer this question, one can understand this outcome by examination of the utilization levels for the two cases that have since been established as those to consider going forward. With one being static and the other dynamic, yet both deploying the same business model factor levels outside of the restructuring schedule, a direct comparison of the two can shed insight into why this is exactly so. The reason has to do with a recurring theme observed in prior discussions of the results. A comparison of the business model's average utilization levels across the considered market conditions is presented in Figure

63. As a general note, the best overall identified business model, the one deploying the option three restructuring schedule, will be referred going forward as the best dynamic business model whereas the other will intuitively then be referred to as the best static business model.

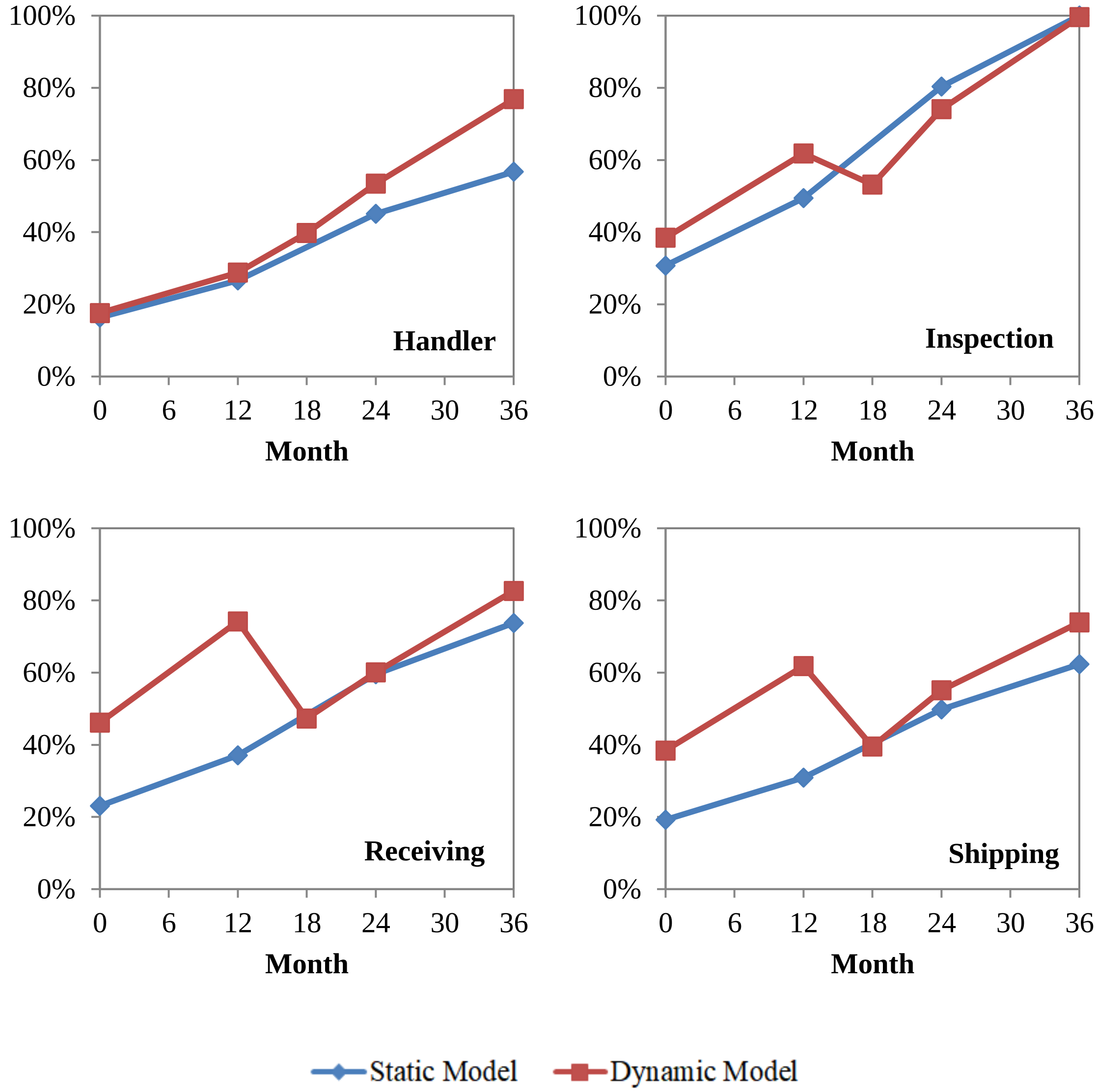


Figure 63 – Average utilization levels across the planning horizon

As can be observed in the plots of Figure 63, except for the inspection stations utilization in the back half of the planning horizon, the dynamic model consistently has higher utilization levels while avoiding becoming over utilized. Most notably, this is so in the early months where the receiving and shipping stations are considerably more utilized. This can be attributed to the dynamic model starting with fewer active stations (i.e. workers) and then adding additional stations downstream to handle the growing PPD the system experiences. By adding workers strategically to the system, the dynamic model can maintain lower indirect labor costs in the earlier months, and as demonstrated in Figure 64. Though in the last year of the planning horizon the static model boasts lower indirect costs (a result of the dynamic model adding additional workers), this is outweighed by the model yielding over $200,000 more in indirect costs over the first year of operation.

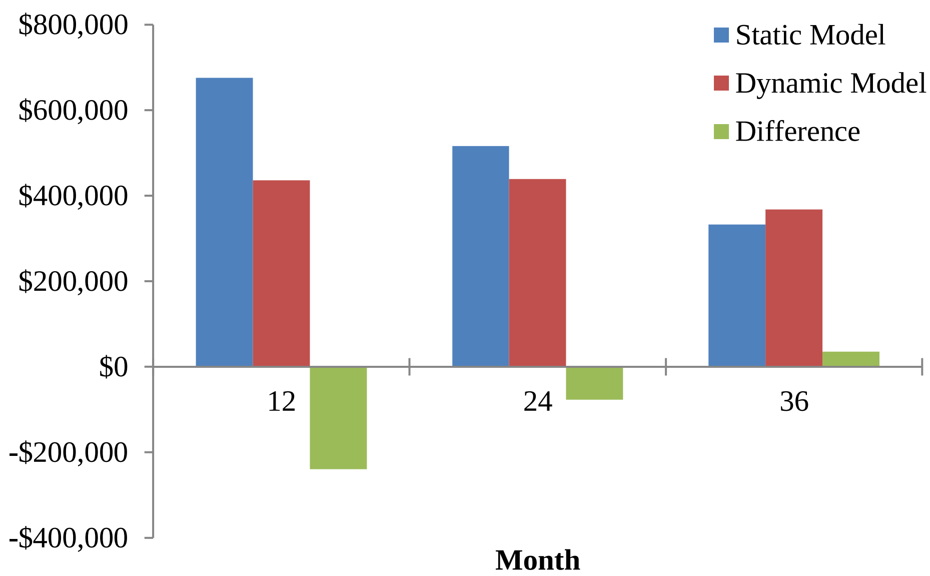


Figure 64 – Indirect labor costs across the planning horizon

The static model produces a largely linear decrease in its indirect costs as a result of not adding any more workers to the system, yet experiencing increases in PPD. At the same time, the dynamic model produces indirect costs across the horizon segments that are far more level. This is a result of this model adding workers to compensate for increases in PPD at the right times. Moreover, it indicates that such a model does a better job at efficiently utilizing human labor in comparison to the static model. At the end of the day, the dynamic business model chosen outperforms the static model thanks to it evolving over time. The ability to observe these differences, behaviours and moreover, understand how such business decisions regarding restructures, adjustments in human resources, and changes in operational design impacts the performance of the system and layout design directly confirms the ability of the LIVE methodology to enable more informed and strategic decisions to be made.

As a reminder, going forward, the business models chosen for further investigation and thus optimization in Stage Two of the solution approach are summarized in Table 41, whereby the business condition design factors associated with the chosen models are established.

Table 41 – Final business models chosen for further study

Factor		**Business Model**	
Condition	Description	Static	Dynamic
Business Conditions	Concept	2	2
	Restructuring Option	1	3
	Station Manning Option	1	1
	Handler Option	1	1

8.1.5 Final Layout Design Results

With the final business models chosen for further study established, the associated scenarios deploying such models were then solved in Stage Two of the proposed bi-model multi-stage solution approach of the LIVE methodology to establish final designs for potential implementation. The optimization parameter set identified in Experiment 5 was deployed while solving said scenario problems. In total, 54 scenario problems were subsequently solved in Stage Two leveraging the results of the earlier executed Stage One of the solution approach.

To establish the best designs for implementation, the designs generated after performing the Stage Two optimization were individually previewed and further post-processed. A unique design was generated from each scenario further examined in Stage Two. This meant one for each unique market condition scenario, i.e. combination of the market condition design factors considered in the study, multiplied by the two business models considered. For each of these generated layout designs, the other market conditions for which it was not originally designed for were then post-applied to this design and the result recorded. The designs to be presented next were those that yielded the best average performance across these market conditions scenarios for each business model and from a retained earnings perspective. A few of the alternative designs generated by the Stage Two optimization are presented in Appendix G, Figure 94 in particular. The best identified designs, after visual inspection and considering their performances across all market conditions of this study, are presented next.

For the static business model, the SO/ROO/ROII and PO/ROIS designs are presented in Figure 65 and Figure 66 respectively. Joining these in the warehouse space can be understood as the area represented in Figure 65 falling at the bottom-left portion of Figure 66. The design of the SO/ROO/ROII space follows expectations where the stations are placed in sequential order relative to the process flows of this group (door, receiving, inspection, shipping, pack, then door in the case of the SO/ROO process). The ROII handling paths can be observed as going over to the inspection station and then back to the Building A Door. This behavior effectively pulls against the stations wanting to come down and to the right towards the shipping door. Also, the advanced flow distance method has forced groups of stations to form (overlapping red boundary lines). Since some flows need to pass between these stations, two pass-through alleys form in the layout as demonstrated by the three bottom-left stations (two receiving and one inspection) and then the two inspection stations in the middle of the layout forming groups and maintaining gaps between themselves and other groups/stations. This is a unique outcome of the advanced flow distance method, one that would not be observed with the traditional rectilinear method deployed. A main artery has also formed down the middle of these stations going left to right, effectively maintaining that the parts continue to flow in a productive manner from their input doors towards the output door of the system. This design also is conducive to enabling the flow of parts coming from the PO/ROIS cross-dock layout, represented in Figure 66. There remains a clear and wide path leading directly to the input/output doors of the system from where this PO/ROIS layout is located up and to the right of the SO/ROO/ROII layout provided in Figure 65.

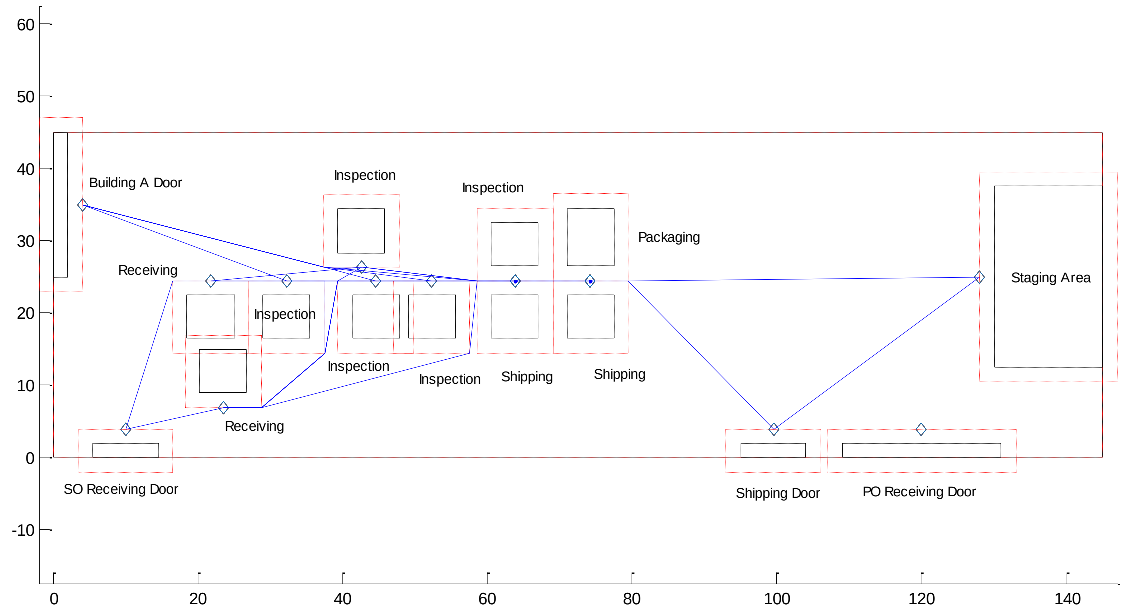


Figure 65 – Layout design for the SO/ROO/ROII operations under the static business model

The PO/ROIS layout, provided in Figure 66, shows a compact configuration whereby the stations are placed in a way that parts can flow about all sides of the stations. It is also observable that the receiving stations fall near the bottom-right corner of the configuration (nearest to the receiving door) while the packaging station falls on the left near the staging area and maintains a direct line of flow towards the shipping door. In this configuration, one can visualize the flow of parts as moving in a counterclockwise motion starting at the bottom-right and ending at the bottom-left. This is like the baseline cross-dock configuration's flow (horseshoe pattern). Inspection stations are understandably then mostly situated in the upper-right corner since they fall in the middle of the process flows of these processes (receiving, inspecting, packaging, ship in that order as a reminder).

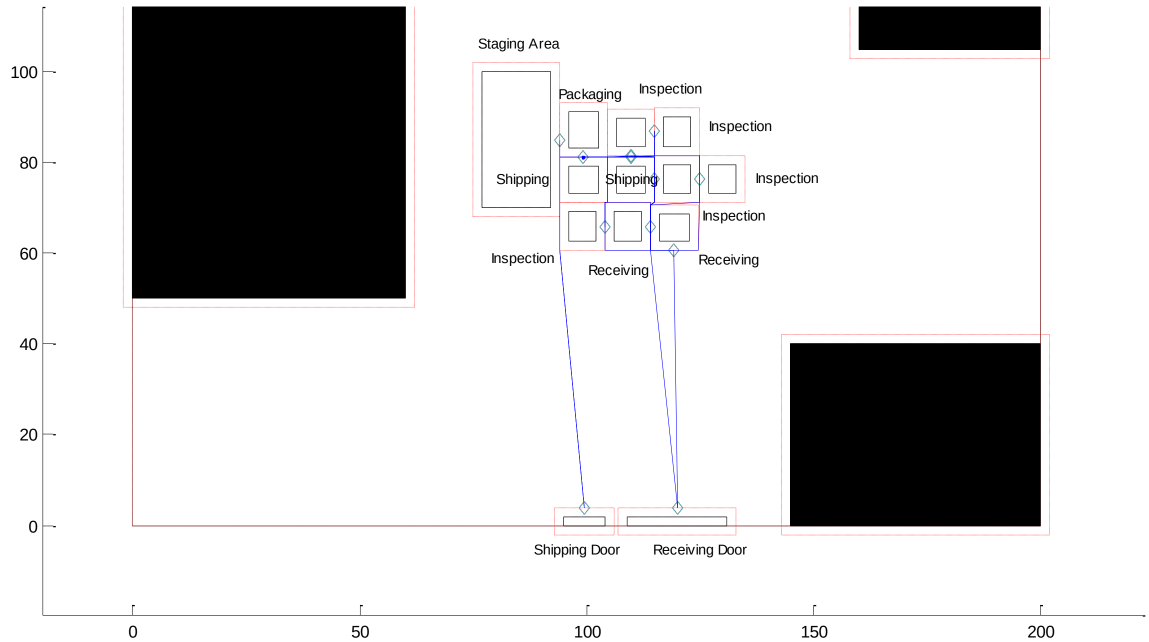


Figure 66 – Layout design for the PO/ ROIS operations under the static business model

In the case of the dynamic business model, the best resulting design can be found in Figure 67 for the SO/ROO/ROII layout design and Figure 68 for the PO/ROIS layout design. For this business model there exists an additional two inspection stations that becomes active in the second period (months eighteen to thirty-six). Despite being inactive in the first period, the stations are strategically placed such that they need not be further moved, nor do that of any of the other stations to accommodate them. In this case, the dynamic layout remained the same from the first period to the next, which is why only a single layout design is presented. In this instance, it was more beneficial to maintain the same layout, but rather construct it to be decent across both periods. The benefit of this approach is that it then avoids the cost of rearrangement. This would explain this result. Given the costs of rearrangement observed before while exploring the design space, this is not surprising. Also, since distributions within the processes flows of this operational space remained proportionate; there is no need to restructure to accommodate such changes.

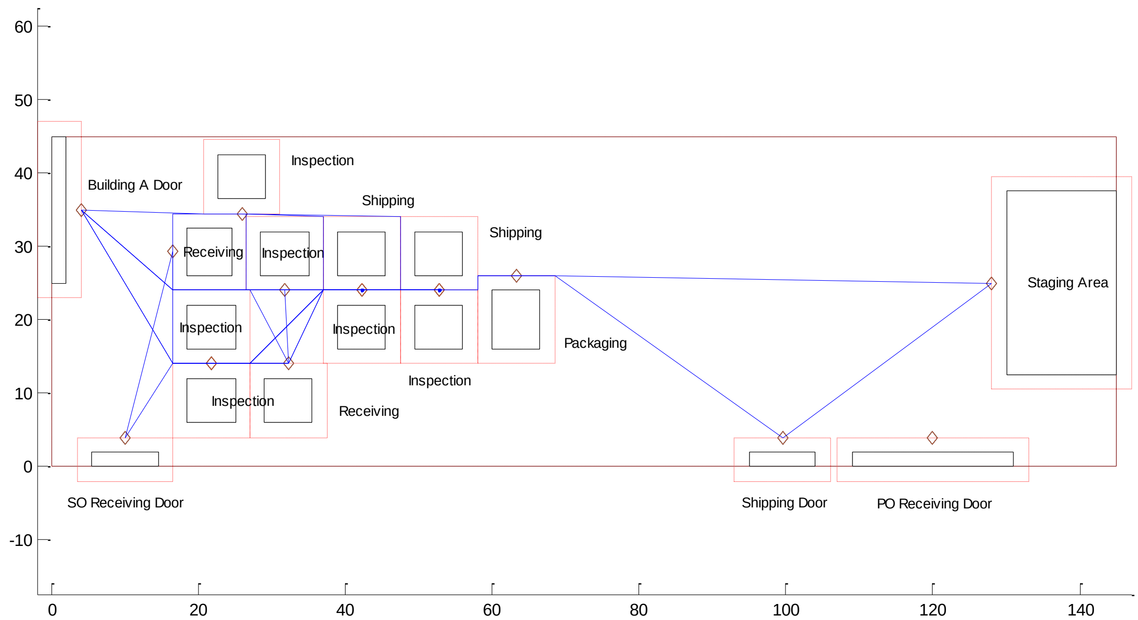


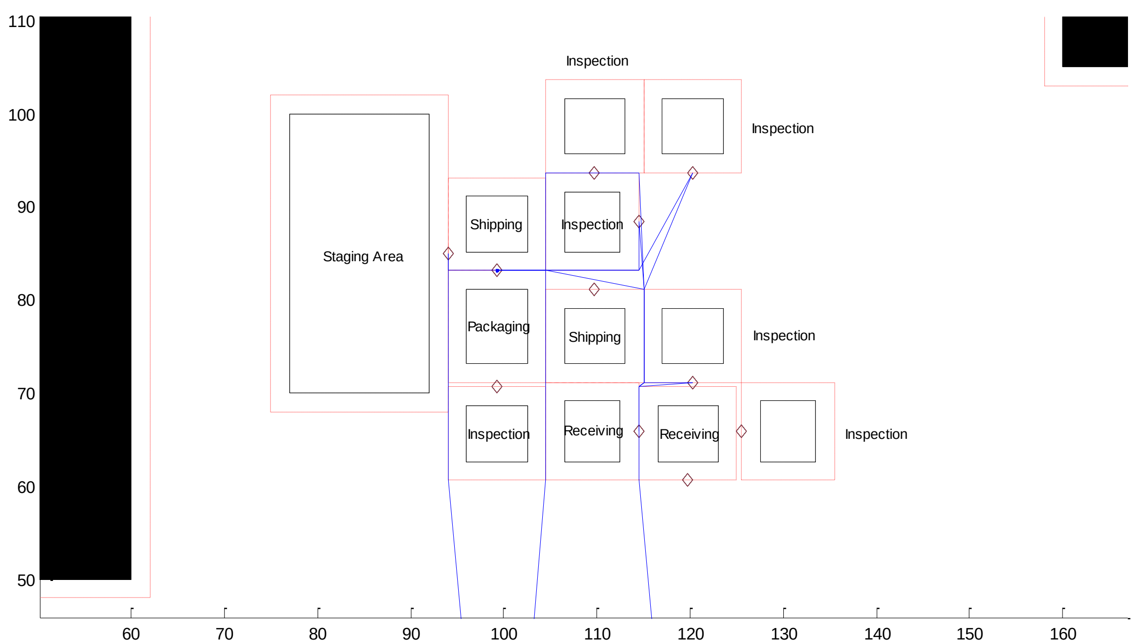
Figure 67 – Layout design for the SO/ROO/ROII operations under the dynamic business model

In general, it can be observed that the resulting design did not come to complete convergence. Just from a visual inspection of the layout, some slight improvements could be made to further improve the layout design. This is a result of the increased dimensionality of the problem and indicates the need to further refine the Stage Two algorithm to provide better convergence. With that said, it remains a good initial design in which could be adjusted further manually if desired. Moreover, this sixteen-object, two-period problem is relatively large for how complex the formulation is and for the number of constraints thus present. Additionally, this problem and all those considered in this study must evaluate well over 100 unique process flows, which has the impact of the algorithm requiring more computational time per generation to evaluate each layout design. This translated to fewer generations being able to be run over the allotted twenty-four hour run time limit set for each scenario problem evaluated in Stage Two. Regardless, the generated design still provides a good initial design to then further improve through designer input.

These observed themes continue into that of the PO/ROIS layout design for the dynamic business model, demonstrated in Figure 68, where the top layout is for the first period and the bottom for the second. Like before, the design remains the same from a configuration stand point. With further inspection though, one can observe the addition of new process flows as new stations become active in the second period. In the first (top layout), the bottom-right most receiving station is inactive during the first eighteen months of the planning horizon before then becoming manned with new hires being made at this point (four in total: one receiver, one shipper, and two inspectors). The same is also true for the shipping station located in the middle of the configuration as well as the bottom-left and right most inspection stations. This can be observed by the new handling paths (blue lines) forming between these stations and others in going from the presented period one layout design (top) to the period two layout design (bottom).

As expected, receiving stations can generally be found at the bottom-right of the configuration while the shipping and packaging station are located closer to the end of the counter-clockwise directional flow loop and near the staging area. Further examination highlights that, for the most part, the initially inactive stations have been strategically placed near the outskirts of the configuration (bottom-left and bottom-right). Since they are inactive in the first half of the planning horizon, placement elsewhere would only lead to them getting in the way of the parts flowing to and from those stations that are active.

Placement in this manner allows the configuration to not change, thereby avoiding rearrangement and loss of production costs, while at the same time remaining relatively effective from a material handling cost perspective. Though it may be hard to see, this configuration in general maintains the horseshoe flow that was present in the baseline configuration. This is particularly true in that of the first period layout design. In this configuration the receiving station comes first, then the four active inspection stations (those four located above the receiving station), then the single active shipping station whose input/output point coincides with the packaging station. From there, the flow can split to either the adjacent staging area or traverse nearly straight down to the shipping door of the system. This configuration facilitates an efficient flow of the parts throughout the system.



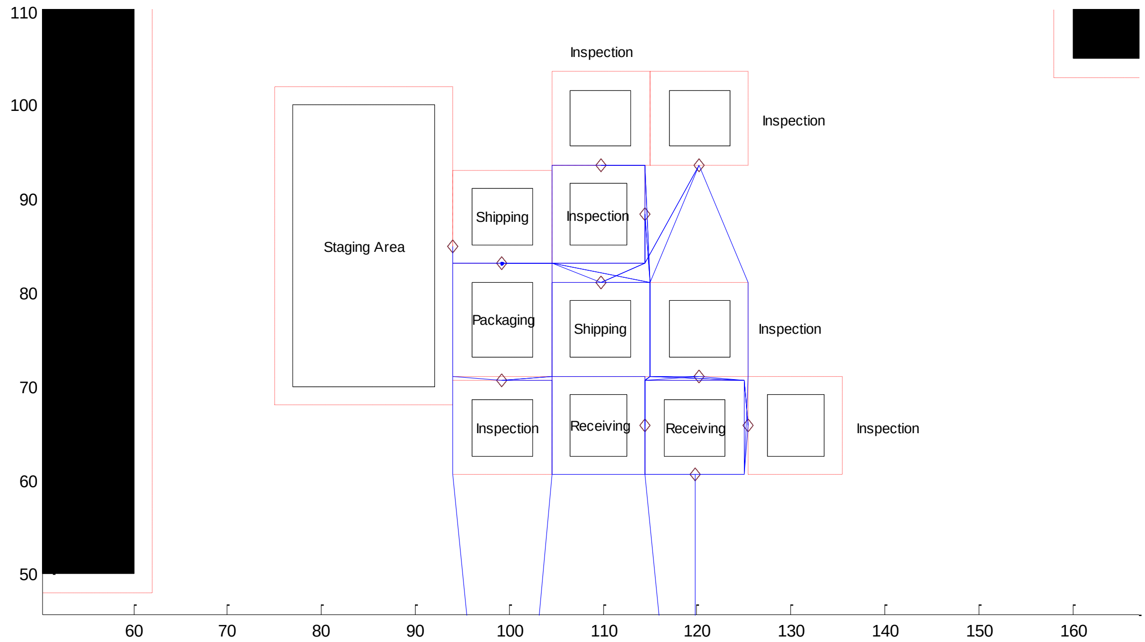


Figure 68 – Layout design for the PO/ROIS operations under the dynamic business model (top – period one, bottom – period two)

Now, had more staging areas been desired they could easily have been placed in the layout as additional staging areas in the process flows of the individual processes (PO and ROIS in this case). Also, if a more distinct horseshoe design was desired by the designer, these stations could have been strategically constrained in the space in such a formation to force the design to then form about these constrained staging areas in such a fashion. With that said and as demonstrated, the same formation may have been inherently captured by the bi-model multi-stage solution approach and deployed performance model of the LIVE methodology had it been considered.

Regardless of which business model is deployed, another notable outcome of these two proposed redesigns of the warehouse layout and operations is the more compact nature of them. As demonstrated in Figure 69, the redesign creates over 10,000 square feet of extra warehouse space that before had encapsulated the more dispersed cross-dock operations (baseline configuration presented in Figure 48). Note that the upper 50 feet of that shown in Figure 69 was already empty as shown in Figure 48, otherwise this area would have been 15,000 square feet in size as opposed to only the noted 10,000. With industry averages for leasing warehouse space falling anywhere from $4 to $7 per square foot, this redesign and extra storage space could by extension establish that this redesign constitutes an additionally savings of anywhere between $40,000 and $70,000 for the firm in just a year of time. Over the three-year planning horizon considered here, that extrapolates to over $200,000 in three years on the upper end of that range. This outcome demonstrates that more than just the benefits of reduced material handling costs and thus higher margins can be realized by redesigning the layout and operations of this system. Recall, this benefit was directly observed before during the beginning of this dissertation while introducing the problem and the importance of the layout design process. Then, it was acknowledged that layout design process could facilitate consolidation efforts such as that experienced by Goodrich Aerostructures [43].

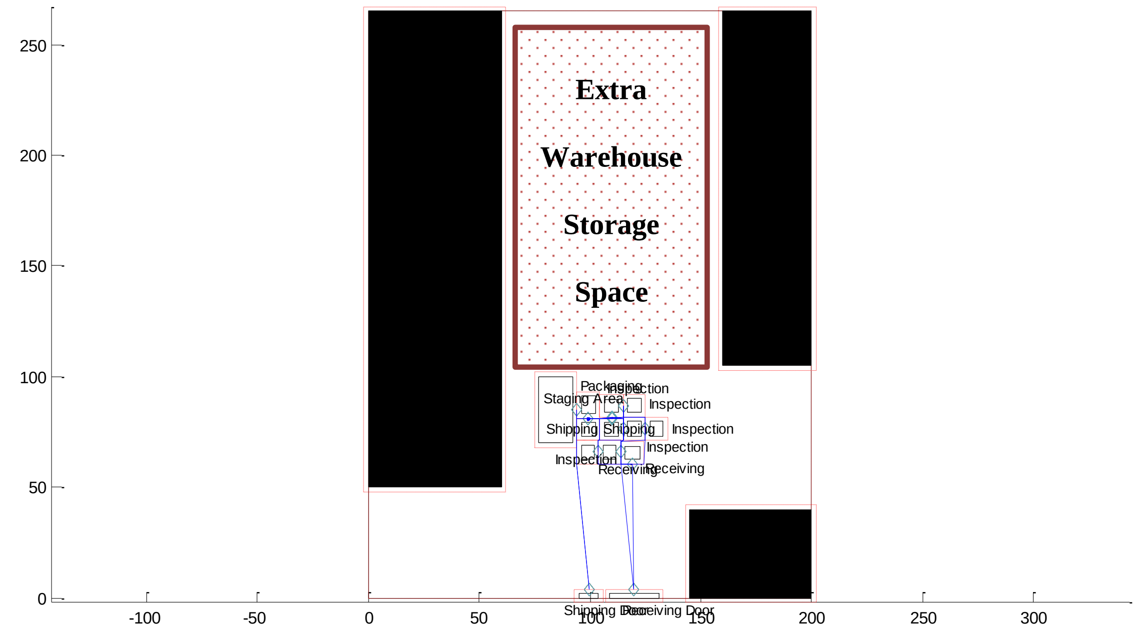


Figure 69 – Extra storage space now available as a result of the redesign

How these discussed redesigns then compare to the baseline configuration and operational landscape can be found in Table 42 and Table 43. The Concept 0, distributed process operational setup results were leveraged to represent the baseline. Again, as a reminder, Concept 0 entertained the unaltered baseline configuration as presented in Figure 48 earlier. In Table 42, a comparison is made from an overall retained earnings perspective, whereby all costs are considered in this metric. As one can observe, while the dynamic model outperforms the baseline configuration dramatically at over 30%, the static model underperforms that of the baseline model. This result can be attributed to the baseline configuration a) not incurring the same rearrangement and loss of production costs the static model is subject to as a result of its initial rearrangement and b) the baseline model considering the distribution of the volumes across all the same active stations, rather than split. This dispersed workload enables it to perform better on average across the different distribution options considered. As the SO PPD increases relative to the PO parts, this has no impact on the utilization of the active stations since both process groups are distributed across all stations. Therefore, the PPD experienced by the stations, regardless of the distribution, will always remain the same. This translates to a design that can maintain higher production rates overall as it is less prone to becoming capacity constrained as a result of the non-distributed volume approach considered in the static and dynamic models.

While this translates then to a higher overall retained earnings achieved by the baseline model, when compared to the static model (10% higher), this advantage disappears when one looks at the direct retained earnings. This occurs since indirect costs are not considered and instead only that of the direct costs are with this metric. The direct retained earnings metric can be thought of as metric in which focuses more on the material handling costs, whereby a less optimally configured layout would then yield higher material handling costs and therefore a lower profit margin per part. This would in turn lead to then lower direct retained earnings. In this case, the higher sustainable production level it can achieve as a by-product of its distributed approach is outweighed by the baseline configuration being less optimally configured in comparison to either the static or dynamic model designs. The baseline is then 4% less optimal than the static model and 11% less optimal than the dynamic model. This translates to roughly $100,000 and $250,000 in reduced costs to the firm over the three-year period of time considered in this study (based on the assumed market value of $75/part).

Table 42 – Comparison of the redesigns to the baseline design on a retained earnings-basis

Business Model	RE	$ Difference	% Difference
Baseline	$1,062,889		
Static	$956,576	-$106,313	-10%
Dynamic	$1,387,111	$324,222	31%

Table 43 – Comparison of the redesigns to the baseline design on a direct retained earnings-basis

Business Model	RE Direct	$ Difference	% Difference
Baseline	$2,382,222		
Static	$2,486,000	$103,778	4%
Dynamic	$2,637,000	$254,778	11%

In either metric, the dynamic model noticeably outperforms the baseline configuration. In the case of the retained earnings, the dynamic business model and adjustments to the operations enable it to then better utilize its human capital, which translates to lower ILCs. This explains why the dynamic model was able to remain advantageous over the baseline configuration in this metric while the static model did not. As such, this difference can then largely be considered a by-product of a better formed business strategy.

It is also important to note, the difference in the retained earnings relative to the static model would be less while the dynamic one would only widen as the current operations do not deploy a material handler, as was considered in Concept 0. This would take capacity away from the system given that the station worker's processing times would need to also then include the time to pick up and drop off products at the next station in the process flow. This would effectively diminish the system's ability to sustain higher levels of production, which would have the effect of then reducing the retained earnings metric for the baseline. As such, this 10% difference would likely be less, if anything at all, if compared exactly to the current operational landscape. Some of this effect may however be mitigated by then such operations not incurring the costs of labor associated with the handler(s).

8.1.6 Concluding Remarks

As was well observed and documented, a redesign of the current configuration and moreover the firm's operations are necessary to ensure profits can be maximized going forward. In the course of the study, it was established that management must be strategic in both the strategies they deploy and even more so the timing in which they are implemented. As was also discovered, the effective utilization of labor is paramount to maximizing profit and thus the firm's retained earnings after the three years of operations considered in this study. Senselessly hiring new labor was proven to be non-beneficial in some instances. Consideration of material handler utilization was also identified as being essential to properly evaluating the system and its capabilities to yield profits. It was also established that a redesign of the operations, which considers segregating the two unique process groups (cross-dock vs. rack processes) in the warehouse, can be of great value to the firm as it enables the material handling costs of the system to be significantly reduced. It also has an ancillary benefit of providing floor foremen with more transparency into operations, given the distinct process lines that are formed. Considering these observations and conclusions, the best business model or set of strategic business decisions was then identified given the considered market conditions the system could potentially be subject to going forward.

In addition to this best business model, being dynamic in nature, the best performing static model across the considered market conditions was also identified. For each of these two business models, two layout redesigns were generated and proposed going forward for implementation to improve the operational effectiveness of the firm. These designs and numerous insights into the operations and performance of the system were facilitated by application of the LIVE methodology to the investigated layout design problem. With that stated, this concludes the presentation of Experiment 6 and thus the experimentation performed in this dissertation.

CHAPTER 9

SUMMARY AND CONCLUSIONS

This dissertation has provided a new method for designing environments, specifically manufacturing environments, subject to evolving and uncertain market and business model conditions. The methodology, referred to as LIVE, provides a new and systematic approach to considering strategic business decisions concurrently with the layout design of a system, thereby enhancing the benefits that can be yielded during the layout design process. The LIVE methodology is represented graphically in Figure 70.

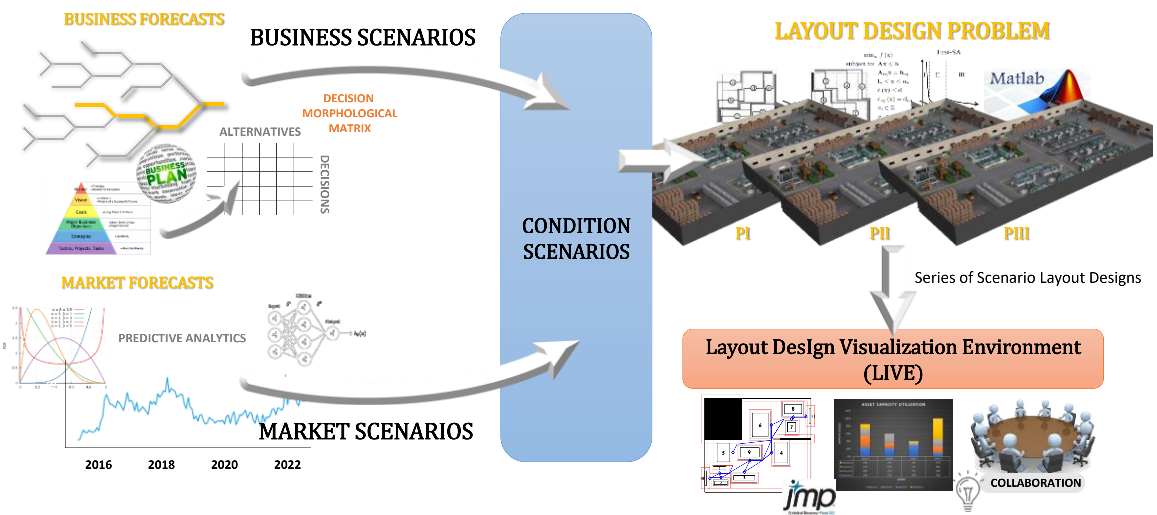


Figure 70 – LIVE methodology framework

The purpose of this research was to continually pose questions that would help facilitate the identification and subsequent formation of an improved way of exploring the design space of a detailed evolving environment such that more informed and collaborative design decisions could be achieved despite the presence of evolving and uncertain market and business model conditions. In the process of doing so, critical gaps in the literature were identified and then systematic approaches formed to provide closure to said gaps. Innovative methods were developed in the course of this research to handle unique challenges encountered while attempting to both provide closure to identified gaps and moreover handle the unique and complex problem characteristics considered in this dissertation. Several hypotheses were formed throughout this process with the overarching hypothesis of this dissertation stated as follows:

> **Overarching Hypothesis:** <u>**If**</u> the problem of designing an environment subject to evolving and uncertain market and business model conditions is solved with the proposed LIVE methodology, <u>**then**</u> designers will be capable of making more informed and collaborative decisions on its design.

In addition to the overarching hypothesis, two other secondary hypotheses were formed in response to the gaps identified during the literature review and formulation of the problem requiring solution. The first was in response to the identified gap in the literature relating to the potential generation of suboptimal layout designs. This was a result of the current flow distance methods failing to consider flow path feasibility when establishing the material handling costs of the system; a major cost factor, as has been well established in the literature. The resulting hypothesis was then presented:

> **Hypothesis 1:** <u>**If**</u> an advanced flow distance method that ensures flow feasibility is implemented to define the MHCs, <u>**then**</u> improved layout designs that are better representative of reality can be established for variable production environments where several interrelated processes are occurring concurrently.

The second of these secondary hypotheses, Hypothesis 2, was then formed in response to both this hypothesis as well as the unique and rather complex problem formulation that was identified as requiring solution in order to accurately evaluate and subsequently establish layout designs in practice. To handle such a problem, Hypothesis 2 was stated as follows:

> **Hypothesis 2:** **If** the proposed bi-model multi-stage hybrid solution approach is implemented to solve the MIP formulated RDLP, **then** the problem will be solved most effectively, in terms of solution quality.

An overview of the experiments then performed to provide substantiation to these hypotheses is presented next, along with the resulting outcomes of the experiments.

9.1 Review of the Experiments

9.1.1 Experiment Set A: Validation of Methods

The goal of Experiment Set A was to test the effectiveness of the developed FSPPM, the novel advanced flow distance method, and the need to infuse FSA into the first stage of the proposed bi-model multi-stage solution approach. Throughout this experiment set, a uniquely designed *52 Problem Test Set* was leveraged as a testing platform. The FSPPM, novel advanced flow distance method, and infusion of FSA in the Stage One GA algorithm were all tested across this problem set.

In Experiment 1, it was demonstrated that the FSPPM performs far better than the traditional random assignment method of the literature. The FSPPM demonstrated its ability to discover feasible layout designs at a far faster rate and as such, made problems before unsolvable to then become solvable. Additionally, it was identified that the best sigma value to deploy, while leveraging the FSPPM, is that of a value of 0.7.

In Experiment 2, it was demonstrated that the novel advanced flow distance method does well in ensuring flow path feasibility is maintained throughout the layout. It was also demonstrated that there exists a distinct difference between the traditional rectilinear result and that generated when deploying the advanced method. Moreover, it was proven that optimizing relative to the rectilinear result yields noticeably inferior layout designs compared to when the advanced method is deployed to establish the material handling distances and subsequently the material handling costs. It was observed that optimizing relative to the advanced method, which considers flow path feasibility, identified layout designs that, on average, we over 12% superior in comparison to those generated using a rectilinear method, only to then consider flow path feasibility after the fact. This outcome in turn provided direct substantiation to **Hypothesis 1** whereby it was proven that such an advanced method, which considers flow path feasibility, is necessary during the layout design process.

In the final experiment, Experiment 3 of this set, it was demonstrated that while infusing the FSA technique provides improved optimality, the substantial time cost associated with its execution deters its application in Stage One when merely initializing the Stage Two GA of the bi-model multi-stage solution approach.

Overall, the testing performed in Experiment Set A, established that the FSPPM should be deployed with a sigma value of 0.7 regardless of the goal of Stage One (initialization or final solution) and in the case of initialization, it was also concluded that the rectilinear method should be deployed and FSA not. With that said, when final solution, and thus optimality, is the goal of Stage One, the advanced flow distance method and FSA should both be deployed to ensure the best layout design is identified by the Stage One algorithm.

9.1.2 Experiment Set B: Optimization Studies

The goal of Experiment Set B was to leverage the outcomes stated before in Experiment Set A to then test the effectiveness of the solution procedures developed as part of the bi-model multi-stage solution approach proposed for solving the complex layout problem formulation of this dissertation. Additionally, identifying how to most effectively solve said layout problems was another core element of the experiment set. In performing these tests, **Hypothesis 2** was in turn directly substantiated.

Without a directly applicable problem formulation available in the literature to compare to, the best alternative was then to set a literature standard and moreover identify how to best tune this solution approach to solve the uniquely formulated problems of this dissertation. To this extent, extensive optimization parameter studies were performed across the constructed *52 Problem Test Set* of this dissertation in Experiment's 4 and 5. In the course of these studies it was identified that of the optimization parameters tested, population size and number of generations contributed most to the performance of the Stage One algorithm while population size, crossover percentage, and migration rate were the major contributors in Stage Two.

The best settings for the tested parameters were also identified as a result of these experiments performed. The results establish that to most effectively, in terms of solution quality (i.e. optimality), solve the uniquely formulated and complex problems of this dissertation, the Stage One and Two parameter settings provided in Table 33 and Table 36 respectively should be deployed. Furthermore, if the simplified formulation, yet still considerably complex and moreover unique one of Stage One is to be solved most effectively, in terms of solution quality (i.e. optimality), then the optimal settings summarized in Table 30 should then be deployed.

9.1.3 Experiment Set C: Case Study

Building on the outcomes of Experiment Set A and B, Experiment Set C sought to test the proposed LIVE methodology by application to a real-world layout design problem. The LIVE methodology was leveraged to examine the operations of an aerospace parts warehouse, the effectiveness of the current layout configuration, and the redesign of it. The study performed, examined the performance over a forecasted three-year period and for a multitude of different market and business model conditions. Robustness relative to production uncertainty was also considered during the process of identifying the business strategies and layout design that would maximize profits over the three-year planning horizon considered in the case study.

In applying the LIVE methodology, the sheer power of it and the value it creates for designers and stakeholders alike was observed consistently throughout the design process. The informative discussions while presenting the results of the Experiment 6 case study were all facilitated and moreover enabled by the LIVE methodology and the solution approach and performance model deployed by it. The observations made throughout the study, all enabled by the LIVE methodology, shed great insight into the layout design problem and additionally the resulting performance of the firm across differing business strategies. In the absence of such a methodology, such would not have been the case. A far more limited understanding of the system and the interplay between both business strategy and the layout design would have been achievable.

The methodology also proved capable of enabling non-uniform unstructured restructuring schedules to be considered in conjunction with human resource adjustments, such as labor allocations and new hires. The ability of the LIVE methodology to facilitate this analysis then enabled the impact that such strategic decisions have on the system to be better understood. All of this created great value during the design process and as was demonstrated, enabled more informed decisions to be made regarding both the design of the firm's layout and operations. Moreover, the integration of the detailed performance model proved exceedingly useful in facilitating enhanced insight into the operations of the firm, which would have otherwise been unachievable by layout design process currently deployed in the literature. Most notably, its inclusion of utilization metrics proved useful in enabling the strategic business decisions to be well understood from a capacity perspective.

The LIVE methodology's ability to provide this enhanced understanding of the operations and layout design thereby enabled more informed decisions to be made in the presence of uncertain market and business model conditions, which directly substantiates the **Overarching Hypothesis** of this dissertation. Furthermore, this case study affirmed an overarching motivation of the dissertation. That being that 10-30% annually can be saved through the reduction of operating costs with effective layout design [71].

9.2 **Forward Looking Future Improvements**

Though the work of this dissertation was extensive, and the LIVE methodology provided to be a significant advancement in the area of how the layout design process is performed, there is always room for further improvement to the methodology and the solution procedures it deploys to facilitate solution to the complex layout problem formulation considered.

9.2.1 Solution Procedure Improvements

While the developed solution procedure of Stage One outperformed expectations, the procedure of Stage Two fell somewhat short. With that said, the solution procedure did perform admirably despite being tasked with solving such a complex and arduous problem formulation. Considering this, a major focus of future improvements to the methodology should be on improving the performance of the Stage Two solution procedures.

It is believed that one such improvement could include pushing the rectilinear method of computing the material handling costs into Stage Two whereby it is leveraged during the isolation period of the GA. The rectilinear method would greatly expedite the solution process in this period where it effectively solves, though only partially, the MIP problem three times in the form of three separate populations. Though it was proven that the rectilinear method can lead to suboptimal solutions, like in Stage One, it could be leveraged to provide partial solution in Stage Two during the isolation period where exploration is more prominent. Further, because the algorithm merges these isolated populations only to continue to search for the best solution leveraging the advanced flow distance method; it should avoid such a pitfall. Even after the conclusion of the first generation it would identify such suboptimal designs and in turn eventually drop them from the population. It would then then instead converge on the true optimal solution that accounts for flow path feasibility. This proposed improvement of the Stage Two solution procedures, at a minimum, could provide significant computational time savings for the overall methodology, but more specifically during Stage Two. With that said, great care would need to be taken to avoid overly converging, thereby eliminating designs that would in fact be better if flow feasibility was considered during the isolation period.

It is also believed that improvement of either the evolutionary process or initialization of the populations in Stage Two could yield improved convergence behavior. It is believed that the advantage of leveraging the QAP/U-SP formulation in Stage One to generate an abundance of diverse and feasible designs may have come at a cost though. Due to the stacking nature of this formulation, altering the designs through the implemented genetic operators of Stage Two proved to be difficult as was observed on occasion throughout the course of this research. For example, if an operator attempts to switch the positions of object a and object b, but object a and b are distinctly different in size or perhaps the same size, but rotated differently (assuming they're non-square), then because of the stacked nature of the configurations, such a switch would effectively cause the objects to then be overlapping. This in turn would then render the design infeasible. Think of it as trying to fit a car through a house door, it simply will not work. This outcome likely led to some of Stage Two's less than stellar performance characteristics, especially from a feasible design discovery and therefore computational time stand point.

As such, it is believed that there are two viable options for remedying this occurrence and they are as follows. One, while initializing the tri-populations, via the layout design set provided by Stage One, only a portion, say 25%, of the population should be established from this set for each population. In other words, if the population size is to be 200, 50 unique designs from the set would be selected. Then the remaining 150 designs would be established by deploying a heuristic technique that randomly selects one of these 50 designs and then alters it by spreading the objects apart. With the QAP/SP-U model effectively packing the objects in a compact fashion, this heuristic would effectively force these objects to be spread apart from one another. The stacking rules outlined earlier in this dissertation could be leveraged to achieve this. One can think of this technique as resembling that of an exploded view of the configuration, much like what you would see in computer-aided design or engineering drawings for an assembly spread apart. Such spreading could be dispersed between the objects randomly. While the 50 selected designs would provide global diversity, this approach would effectively create local diversity about the 50 designs. It is believed that this approach would then enable the genetic operators implemented in the Stage Two algorithm to then evolve the populations more effectively. More effectively evolution would then lead to improved convergence properties and solution quality by the Stage Two algorithm. The second option, by extension would be to apply this same heuristic as a genetic operator in the evolutionary process. The synthesis of the two could yield compounding benefits if considered in the future.

Yet another potential advancement of the current approach is to infuse in the first stage GA a cross pollination technique. It is proposed that during this stage the simultaneous solution of like problems could be considered to enhance the performance of the stage. In early generations, the GA has yet to focus on the region of optimality; instead the time is spent exploring the space. It is believed that this property could be leveraged early on in Stage One's progression, where exploration is more highly valued in comparison to optimality, to enhance the solution process. Because like problems have the same structure, they can easily share layout designs for individual periods or across all periods as all physical conditions of the layout remain the same. Sharing explored designs could enable a larger search space to be explored in a shorter duration of time and furthermore reduce the computational overhead associated with determining flow distances. This would be especially advantageous if the advanced method is employed rather than a rectilinear one. For example, for five like problems, a population size of 10 each can enable 250 designs to be evaluated overall amongst these five problems (50 each problem) while the advanced flow method would only need to be performed 50 times to achieve this analysis.

Provided this, a future proposition is that a cross pollination function, based on the diversity of the individual problem populations, be employed to regulate this cross pollination that occurs between the problems collectively solved. As the diversity of a problem's population decreases (i.e. it narrows its scope towards its own optimal region) the amount of cross pollination occurring would decrease until eventually none would be performed. The problems involved will maintain their own populations, but after each generation the populations of each of the problems will be pooled and evaluated according to everyone's conditions. The cross pollination would then dictate how many, if any, of the other problems supplied population individuals would then be selected for replacement relative to its own. If none of these provide superior solutions, then no cross pollination would result. This population cross pollination technique could prove highly advantageous to reducing solution times and improving the performance of the solution process.

This concept of cross pollination is not a revolutionary one. There exists an optimization technique in the literature referred to as the flower pollination algorithm (FPA). Proposed by Xin-She Yang back in 2012, FPA emulates the pollination process of plants to solve single and multi-objective optimization problems [18,175,176]. Although the pollination process emulated in this algorithm differs rather substantially from the concept behind the proposed technique, inspiration in forming the proposed technique's function could be gained from observing the flower pollination algorithm's fundamental premise.

This technique would however require a large degree of memory and further a revision to the current data management scheme deployed in the current LIVE methodology. As such, implementation of this concept going forth would be a substantial effort if it were to be pursued.

9.2.2 FSPPM Improvements

Another improvement to the developed methods was observed while performing Experiment 6. As was acknowledged then, a future extension of this work should focus on further developing the novel FSPPM. Such efforts should attempt to account for sparse, high white space layout characteristics, constrained objects with small areas relative to the other objects and the space itself, along with the general positioning of the constrained objects relative, not only to the diagonal line, but also their vicinity to the boundaries. Dependencies on these characteristics should be built into the functions that define the expected position and sigma of the individual constrained object distributions. The following forward-looking hypothesis was made considering this:

> *If the FSPPM is further developed to encapsulate an algorithm defining the expected position and sigma values of the constrained objects on an individual-basis and moreover encapsulating the above acknowledged dependencies, then better placement performance by the FSPPM would be observed under such layout characteristics.*

9.2.3 Platform Improvements

Before any of these improvements are likely to be made though, the next step for the author is to develop an extensive GUI that can facilitate the data analysis and layout visualization presented in Experiment 6 in a user-friendly platform. This platform would act in further facilitating collaboration among stakeholders during the layout design process. Moreover, a conversion from MATLAB scripting language to python is desired to enable the LIVE methodology and tools developed to become completely open source.

9.3 Closing Remarks

This dissertation began with the goal of improving the layout design process under uncertain and evolving conditions by providing a more accurate representation and evaluation of the design. In the process of accomplishing this goal, the LIVE methodology was formed. Along with its formation an extensive array of novel methods, performance models, optimization techniques, and new applications of existing genetic operators were developed. The novel QAP/U-SP model developed, the revolutionary application of a genetic algorithm to the DLP variant of the problem, the original application of the jumping gene operator to such a model, and the novel heuristics developed for the various genetic operators and FSA perturbation schemes deployed in Stage One are noteworthy contributions to the literature. The novel FSPPM method developed to more effectively handle the QAP/U-SP formulated problem under constrained object scenarios is yet another notable contribution. Moreover, the novel application of the jumping gene operator and the novel repair processes developed for both the Stage One QAP/U-SP model and Stage Two MINLP model of the layout under evolving asset landscapes provides a substantial advancement of the literature pertaining to the LP. Collectively, these efforts contribute greatly to advancing both that of the optimization portion of the layout problem literature, but also that of the mathematical programming literature. The mathematical programming techniques developed here apply beyond the scope of this problem application and universally advance solution to similarly structured combinatorial optimization problems.

This work also demonstrated that the LIVE methodology effectively facilitates improved insight and potential collaboration into the layout design process. The methodology demonstrated its ability to provide an improved layout design process that can effectively handle design problems subject to uncertain and evolving conditions; enabling strategic business decisions to be considered in parallel to the design of the layout. It is the author’s sincere hope, that the work performed in the dissertation can help advanced not only the field of lean manufacturing and layout design, but also the fields of operations management and mathematical programming.

APPENDIX A

COMPREHENSIVE REVIEW OF THE LAYOUT PROBLEM

This appendix focuses on providing expanded, more comprehensive reviews of various topics covered in Chapter 2 where the background on the overarching topic of this dissertation (i.e. the layout problem) is presented.

A.1 Review of Other Layout Performance Measures

This portion of the appendix focuses on providing a more comprehensive review of the other layout performance measures considered in the literature and briefly cited in Chapter 2.

A.1.1 Flexibility

Neglecting the quantitative measures of material handling costs (MHCs) and rearrangement costs (RCs), flexibility is likely the next most frequently considered measure of performance in the literature. As it pertains to the facility layout problem, flexibility is traditionally defined as the layout's ability to efficiently adapt to evolving and uncertain customer demands and internal disturbances without a substantial degradation in operational performance [98,165,135,171,173]. As observed by Koste and Malhotra, as well as others, there are several important dimensions of flexibility in the facility environment. As they note, those of most importance include, flexibility with respect to volume, product mix, new product introduction, product modification, machine capabilities, labor, material handling, routing, operations (i.e. product process alternatives), and facility expansion [98,165,135]. Explicit definitions of each of these can be found in [98]. Before proceeding, it shall be acknowledged that given its similarity to the concept of layout robustness, the two are often coincidently established in the literature. This is despite the two being subtly different from one another. Considering this, it shall not be surprising that crossover be observed between the following discussion and the one provided in Chapter 2 on robustness.

Establishing a quantitative measurement of layout flexibility has been approached in various ways by researchers in the literature. Many in the literature have measured flexibility as the layout which performs, most consistently on a MHCs basis, the best across a series of scenarios. These scenarios often address flexibility in the product-based dimensions, noted above, by encapsulating variations in product production demand across the planning horizon [171,173,39,84,116]. Rosenblatt and Lee, considered a slightly different form of this flexibility notion. Though still MHC-based, they opted to define the flexibility instead as the layout which most frequently fell within a pre-defined percentage of the layouts with optimal MHC across the scenario set [147].

Malakooti and D'Souza took a rather different approach to defining flexibility. They measured flexibility as the ease by which the layout could be arranged and rearranged. Using a ranking scheme to establish the importance of various department proximities, they were able to capture flexibility in this rearrangement dimension [116].

Raman et. al. took yet a different approach to quantitatively defining flexibility. They argued that defining flexibility solely based on MHCs, as often was done, was unwise as it neglected other significant factors that also contribute to the flexibility of the layout [141,142]. Ultimately, they considered additional significant factors which they then grouped under the three flexibility dimensions of expansion, volume, and routing. Acknowledging that many decisions associated with these factors are knowledge-based and qualitatively made by designers, Raman et. al. opted to implement a complementary knowledge-based measurement approach to solve for the flexibility. Using fuzzy rule-based system (FRBS) to quantify these factors, Raman et. al. were able to capture flexibility with respect to all three dimensions noted above.

Provided here is only a brief overview of the literature pertaining to measuring flexibility in the facilities layout problem. A more comprehensive review would surely uncover additional methods of quantifying flexibility beyond that of those discussed. Not being a focus of this dissertation, the discussion on flexibility ends here and we move on to discussing another often-observed measure of layout performance in the literature.

A.1.2 Spatial Utilization

Spatial utilization, also sometimes referred to as space utilization, area utilization, or productive area utilization (PAU), is another common measure of layout performance considered in the literature. Spatial utilization addresses the concern of layout designers regarding how effectively the layout space is being used by operations. Variation in how this is quantitatively measured exists in the literature.

Often, a measurement of the free space of the layout is leveraged to define the space utilization metric. Lin and Sharp for example, used the ratio between the free area available in the space divided by the total layout area in addition to the concentration of the free space in the layout to characterize space utilization [112]. Raman et. al. however, contest that this approach to quantifying utilization is flawed given it focuses on the notion of free space rather than that of utilized space [141,142]. Though they concede that such an approach would be helpful in assessing future expansion potential (i.e. expansion flexibility discussed in the preceding section), they maintain that it provides a poor characterization of utilization. In fact, they argue that to effectively define space utilization, one must consider the idea of value adding and non-value adding utilized space to sufficiently characterize the space utilization metric. If this concept seems familiar, that is because Raman et. al. drew inspiration from the lean manufacturing concept of waste minimization in establishing this perspective. Furthermore, just like that of lean manufacturing seeking to minimize waste, their approach seeks to minimize the area utilized for non-value adding elements [142]. To define the PAU, Raman et. al. used an Analytical Hierarchy Process (AHP) method to quantify the qualitative estimations of value adding vs non-value adding proportions of a given element in the space. The PAU under their approach is in turn defined as the ratio of value adding space to that of the total utilized productive area excluding all free space.

A.1.3 Work-In-Process

Another frequently addressed measure of layout performance in the literature is work-in-process (WIP). Work-in-process, by definition, is the collective cost of all partially unfinished products in production at a given moment in time [63]. One can easily see the value in addressing such a measure and furthermore seeking to minimize it. WIP, as a byproduct of Little's Law, can be related to the throughput of a product [109,8,114,27]. If operations are ineffective and products are often being held up along the way or taking a long time in transit, WIP will be observably higher than if they were flowing through unimpeded. This alludes to the issue of congestion in the environment. Benjaafar, having acknowledged this dependency between WIP and congestion, sought to mitigate congestion by leveraging WIP as the primary measure of a layout's performance [27].

Building on the work of Fu and Kaku, who years earlier also addressed WIP in the layout design problem [74,75], Benjaafar found that reducing overall distances between departments can, though counterintuitive, increase WIP. Additionally, they discovered that the performance of a layout can be affected by non-material handling factor such as utilization levels, department processing time variations, and further product demand variations. To measure WIP, Benjaafar implemented a probabilistic queuing network approach. Benjaafar's approach was capable of characterizing the expected WIP allocated to each department individually and further to the material handling system by probabilistically assessing product travel times (loaded and empty trips). Leveraging Little's law, Benjaafar was further able to establish total expected flow times of individual products in the system.

A.2 Review of Layout Problem Solution Approaches

This portion of the appendix focuses on providing a comprehensive review of solution approaches deployed in the literature to solve variations of the layout problem (LP). This appendix begins by observing exact methods of solution, followed by heuristic, meta-heuristic (simulated annealing techniques and genetic algorithms), and finally hybrid intelligent approaches deployed in the literature to solve variations of the layout problem.

A.2.1 Exact Methods

As noted in Chapter 2, a few of the more prominent exact methods implemented in the literature to solve the LP include branch and bound, dynamic programming, and direct methods. Each of these methods and relevant works are discussed in detail here in the sections that follow.

A.2.1.1 Branch-and-Bound

The branch-and-bound (B&B) method was originally developed to solve discrete optimization problems [38]. B&B solves combinatorial optimization problems (COPs) like that of the LP through a recursive process where at each iteration the problem is branched intelligently into sub-problems of reduced size [7]. Implementing bounding techniques helps to avoid a complete enumeration of the problem, allowing problems of reasonable size to remain solvable. Without such techniques a structured static layout formulated by QAP/S of size 7 would require 5040 layouts and generalizing to an unstructured static layout formulated by a QAP/U model such as SP would require a burdensome 25.4×10^9 layouts to be evaluated before optimality could be guaranteed. Imagine now extending this to a multi-period dynamic problem and the solution space would become so large that it would be intractable to solve even a small sized DLP without the aforementioned bounding techniques.

Branch-and-bound methods have been implemented to solve a variety of LP formulations, with most being applied in the static QAP domain as solutions remain achievable in a reasonable amount of time for moderately sized problems. One notable application of B&B to the LP was Kim and Kim's use of it to identify the optimal I/O point placement that minimizes the total transportation distance for a given block layout.

Using linear programming and heuristics to establish the lower and upper bounds for their B&B algorithm respectively, they were able to solve problems involving 30 departments effectively [91]. Like most researchers in the literature implementing linear programming to solve sub-problems, Kim and Kim used CPLEX, originally developed by Robert E. Bixby and now continually developed under IBM [32,31]. Although Kim and Kim solved a 30-department problem, their method assumed a given layout was known. In other words, they only solved a sub-problem of the LP with B&B.

In the dynamic domain, very few have entertained the use of B&B due to the solution complexity that accompanies it. One of the few to implement B&B to solve the DLP has been Lacksonen (1994). Lacksonen (1994) proposed a two-stage approach to solve the Montreuil inspired MILP formulation of the DLP [125,106]. The first stage of Lacksonen's (1994) approach solved the QAP formulation of the DLP to generate good approximate layouts by a heuristic cutting plane routine. The second stage then solved a modified version of Montreuil's MILP formulation of the DLP using a B&B approach combined with the optimization subroutine library (OSL) subroutines. The solutions of Stage One enabled the number of integers present in Stage Two to effectively be reduced thereby enabling problems of moderate size to remain tractable. Their approach improved solutions to SLPs and solved DLPs that had yet to be solved in the literature at the time [104].

A.2.1.2 Dynamic Programming

With the exception of Lacksonen and very few others, most researchers have leveraged dynamic programming (DP) to solve the DLP to optimality. Much like that of B&B, DP decomposes the problem into sub-problems, storing their solutions as they are solved recursively in order to gain solution to complex optimization problems [110]. Rosenblatt, being the first to propose and subsequently solve the DLP, leveraged DP to solve the QAP/S formulation of the deterministic (product demands are known and constant for each period) problem. In his formulation each stage of the DP characterized a period of the planning horizon and each layout a state of the DP. Rosenblatt was able to achieve solutions to problems involving six departments and five periods [146].

Like that of B&B, DP can solve the DLP to optimality for only small sized problems due to solution complexity. To emphasize this, consider the QAP/S model case where there are N departments in the layout and a total of T periods in the planning horizon. To solve said DLP to optimality would require $(N!)^T$ layout plans to be explicitly or implicitly evaluated. For a problem of just six departments and five periods, like that solved by Rosenblatt, 1.93×10^{14} layout plans must first be evaluated. Therefore, solving larger problems becomes unrealistic without the use of heuristics to reduce the combinations that must be evaluated. Rosenblatt among others have suggested and since implemented such heuristics to enable problems of larger size to remain tractable [146,13]. It should be noted however, that in doing so, there is no longer a guarantee of optimality for the DLP. The implementation of heuristics to solve the LP and more specifically that of the DLP, becomes the focus of the proceeding section. First though, a discussion of gradient and simplex based direct methods is presented.

A.2.1.3 Direct Methods

The implementation of direct solution methods has largely been motivated by the need to handle problems involving flexible modules (i.e. blocks that can change in size) and/or continuous layout representations. Up until now discussions have, for the most part, focused on solving QAP formulations of the problem. Direct solution methods, however, enable MIP, more specifically MILP formulations of the LP to be solved effectively. As such, most of the proceeding literature applies to MIP formulations of the problem.

The effectiveness of said methods in solving dimensioning problems (i.e. flexible modules LP) was no better demonstrated than by Bhowmik's use of an improved move limit method of sequential linear programming to solve the nonlinear programming building design optimization problem [30]. Others like Sutanthavibul and Shragowitz, more generally solved this same dimensioning LP, but while also simultaneously determining the optimal position of said flexible modules [155]. Their detailed SLP, formulated as a MIP and capturing layout continuity, layout boundary restrictions, flexible module characteristics, and module rotations, was first linearized forming a MILP. The then linear nature of their problem enabled them to implement LINDO, a commercial linear programming software, to solve their problem to optimality [150].

Much like that of Sutanthavibul and Shragowitz's formulation, Barbosa-Póvoa et al. (2000, 2001) also solved the detailed SLP as a MILP with commercial linear programming software [20,21]. Their formulation differed slightly from Sutanthavibul and Shragowitz's in that it did not encapsulate flexible module design. Instead they supplemented the formulation by considering irregular shaped rectangular departments and department I/O points. Furthermore, instead of using LINDO, Barbosa-Póvoa et al. relied on the CPLEX optimization package (v6.5) in conjunction with the Generic

Algebraic Modeling System to solve the problem [37]. Despite the added complexity of their formulation, they were able to solve an 11 department, 22 I/O point design to within 5% margin of optimality in an acceptable duration of time, just over 11 minutes [20,21].

Unlike those previously mentioned, Balakrishnan et al. (1992) sought to solve the QAP formulation of the DLP using a direct method. Furthermore, they were the first to address the often present constraint on financial resources available for rearrangement, by considering the budget constrained problem. Motivated additionally by the desire to compare the ability of network-based algorithms to efficiently solve said problem to that of DP, they proposed a simplex-based constrained shortest path (CSP) algorithm. Their algorithm proved to perform better than DP with heuristics implemented, except for when the problem size was small and/or tightly constrained. Furthermore, they identified that selection of candidate static layouts with a mix of best and random layouts provided solution results that were close to or even surpassed that of a purely best layout population of candidates [17].

Few others such as Zhan, Feng, and Sapatnekar have employed gradient based methods to solve the LP. Zhan et al. implemented a multistage (rough floorplanning and floorplan legalization stage) conjugate gradient method with reasonable success to solve the flexible module boundary constrained SLP [177]. Solution of the SLP via this method requires several challenges to be overcome in order to ensure proper convergence. For example, the initial condition has to be feasible, the solution quality is highly dependent of the initial condition, and constraints (e.g. the object boundaries) must be smooth to ensure proper convergence. Zhan et al. addressed these challenges by using bell shaped functions to represent objects (overlap constraints) and a recursive method of evaluating several different initial layout conditions [177].

This concludes the survey of relevant research implementing direct methods to solve the LP. Next, the heuristics approaches to solving the LP is now discussed in detail.

A.2.2 Heuristic Approaches

As noted in Chapter 2, overcoming the limitations that accompany exact methods has been the principal purpose of implementing heuristics techniques to facilitate more effective solution of the LP. Of these, the ability to solve larger problems more effectively has by far been the most prominent motivator to their implementation in the past. As observed by Rosenblatt however, the implementation of heuristics to reduce the number of states that need be considered by DP or any other approach for that matter, results in a solution method that can no longer guarantee optimality [146]. Reducing the number of states is analogous to reducing the search space. By choosing to consider only a limited number of the possible layouts in each period, there is the possibility that the layout yielding the best collective layout plan could be left out from consideration. In the case of DP however, the computational time is exponentially a function of the number of states. Therefore, reducing the number of states by heuristics procedures can enable larger problems to become solvable. Without heuristics, only relatively small problems on the order of ten to fifteen departments are computationally tractable [146]. As such, researchers like Rosenblatt have had to weigh the tradeoff between guaranteed optimality and computational effectiveness.

Like that of Rosenblatt, researchers have often favored computational effectiveness over that of guaranteed optimality. Rosenblatt's heuristic, which is similar to that of Ballou's procedure applied to the warehouse location problem [19], considers only the best t x n layouts as states for each period, or stage of DP. The best layouts are determined first by solving optimally the SLP for each period and then selecting the best n from each period, t. Any duplicate layouts (states) are discarded accordingly resulting in < t x n states that need be considered. This approach provided reasonably optimal results (within 1.2% of optimal), for the selection of just the best four layouts from each of the five periods for a total of 20 layout states for consideration in each stage (period) of the DP [146]. Rosenblatt's success in implementing heuristics to solve the DLP inspired others like Urban to do the same.

Urban proposed a steepest-descent pair-wise exchange heuristic, a multi-period, rearrangement cost considering equivalent to Buffa et. al.'s Computerized Relative Allocation of Facilities Technique (CRAFT), to solve the QAP/S formulated DLP [168]. CRAFT is a heuristic improvement algorithm that considers all pair-wise exchanges between each department location and every other one in the layout configuration for a, or several, supplied initial layouts [9]. Figure 71 demonstrates graphically the data dependencies of each period layout and these pair-wise exchanges that occur from one period to the next in a forward pass behavior. Urban's heuristics avoided the computational overhead associated with DP approaches, like that implemented by Rosenblatt earlier, while performing better than said DP approaches and only slightly worse than optimal.

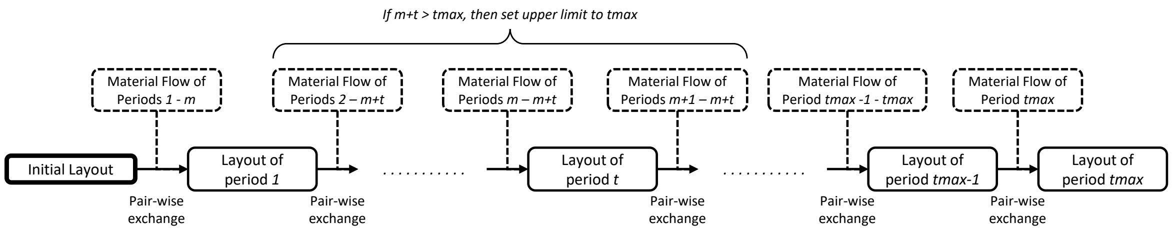


Figure 71 – Urban's pair-wise exchange heuristic [168]

Building on Urban's research, Balakrishnan, Cheng, and Conway proposed two heuristic methods [14]. The first was an improved multi-pass pair-wise exchange heuristic. Urban's heuristic is forward pass in nature. Once a layout for a period is established it never changes downstream. As they observed, the implication of this is the quality of later period layouts is strictly dependent upon its predecessors. This is obviously a significant disadvantage of Urban's approach and Balakrishnan et al. (2000) understood this. To address this shortcoming, they implemented a backward pass pair-wise exchange procedure after first solving the DLP by Urban's heuristic to further improve the solution. Starting from period *tmax-1* and continuing until period one is reached the process as demonstrated in Figure 72 is performed.

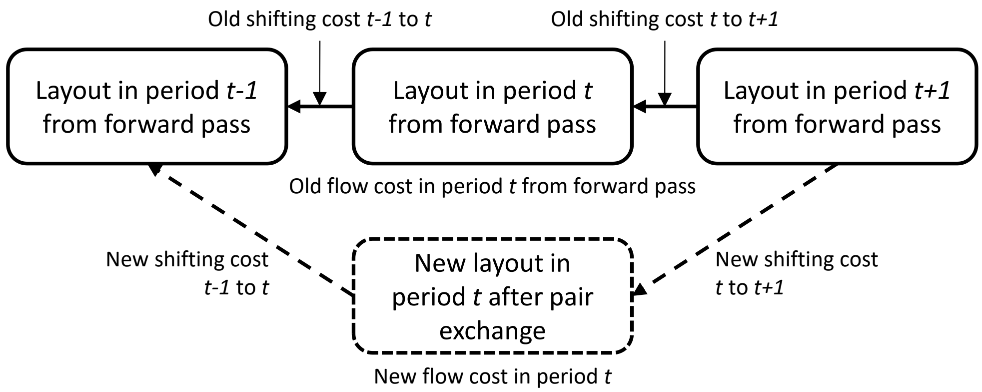


Figure 72 – Balakrishnan et al.'s (2000) backward pass pair-wise exchange procedure [14]

With the backward pass operating on the layout plans generated from Urban's forward pass heuristic, it is guaranteed that the backward pass will never produce a plan worse than that of the forward pass plan.

Balakrishnan et al.'s (2000) second method fused Urban's heuristic with Rosenblatt's DP procedure. The layouts generated for each of the *m* forecast windows (where *m* equals the number of periods, *t*, in the planning horizon) by Urban's heuristic become the states in Rosenblatt's DP procedure discussed earlier [14]. With Urban's heuristic embedded in the DP, it is guaranteed to produce a solution at least as good as that in which Urban's heuristic alone can provide. Furthermore, the overhead associated with using DP is not too substantial since the states that must be considered remains reasonable (at most t^2). For a series of problems, both of Balakrishnan et al.'s approaches implementing heuristics, to solve the QAP/S formulated DLP demonstrated improved results over that of Urban's approach alone. Included was a problem of thirty departments and ten periods, a respectable sized problem [14].

Erel, Ghosh, and Simon later proposed a novel heuristic scheme based on the idea of viable layouts to solve the QAP/S formulated DLP introduced by Rosenblatt nearly a decade earlier [67]. Much like that of those before, Erel et al.'s approach arrives at the best layout plan by implicitly enumerating over a subset of the possible layouts. Erel et al. observed that the DLP could be regarded as a shortest path problem on a multi-stage (each planning period), directed, acyclic network with costs on both nodes (all possible layouts) and arcs (moves between one period to the next). Leveraging this, Erel et al. proposed converting the DLP into a shortest path problem before then using a DP procedure to solve it. Furthermore, they proposed a new heuristic that relied on weighted flow data from each period to generate viable layouts to constitute the states for the DP procedure. Their implementation of heuristics in conjunction with DP proved a viable one, demonstrating results that were computationally competitive to other solution methods found in the literature at the time. Erel et al. also acknowledged that the shortest path problem could be solved by network programming if desired.

Balakrishnan and Cheng more recently extended the work of Balakrishnan et al.'s (2003) by addressing the DLP with rolling planning horizons and under uncertainty [16]. Prior, only fixed planning horizons, which are not representative of how companies actually plan, had been considered. Balakrishnan and Cheng observed no significant difference between using a five versus a ten-period planning horizon. They in turn concluded that when the acquisition of additional data is too costly or too difficult it may be more beneficial to implement just a five-period planning horizon. It was also discovered that algorithms with self-adjusting capabilities performed best when rolling horizons were considered. This is not a characteristic that DP poses, making it a less than ideal solution method for such problems. Furthermore, it was observed that forecast uncertainty does not significantly affect the performance of heuristic/DP-based algorithms developed without consideration of uncertainty. In some cases, this impact was even observed to have been beneficial to their performance. This outcome is an important one to note as it indicates that an algorithm applied to the non-stochastic DLP may remain effective should it be applied to the DLP under uncertainty.

One of the few researchers to have applied heuristics to the MIP formulation of the DLP has been Yang and Peters [173]. Yang and Peters sought to address the flexible LP, a problem that differs from that of a DLP in that the periods forming the planning horizon are not apriori defined. Instead, the period lengths are chosen by the solution procedure to be such that the total costs are minimized. The heuristic balances the trade between increasing the planning window time thereby reducing rearrangement costs and increasing MHCs that are a result of a less efficient layout. As observed by Lacksonen and Enscore and stated before, the requirement to formulate the problem as a DLP is driven by the need to balance these two attributes [104]. The relationship between these two attributes is demonstrated in Figure 73 and is verified by experimental data collected by Afentakis, Millen, and Solomon [6]. The longer the duration between rearrangement occurs, the higher the MHC becomes as a result of the layout become far less suited (i.e. effective) for the changing conditions.

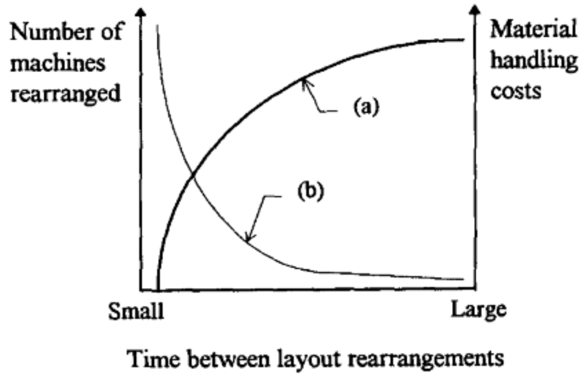


Figure 73 – Correlation between number of rearrangements and MHCs [173]

The implementation of this heuristic significantly improves the computational efficiency of the solution procedure as it effectively reduces the dimensionality of the overarching problem by reducing the number of layout plans that need be considered. To illustrate this point, consider the scenario where the heuristic identifies that the layout can be maintained for two additional periods following the first for a five-period problem. In this case, the heuristic reduces the DLP to one of just three planning windows with one being three periods in length. A DP procedure, if chosen to solve the problem, would then have to solve a problem with just three stages as opposed to five. From earlier discussion, it is understood that this would provide substantial computational time savings. To solve their problem, Yang and Peters adapted their previously developed adjacency graph/integer program formulation instead of using DP [104,136]. Their approach implemented a structured hexagonal adjacency graph adopted from the Spiral Procedure proposed by Goetschalckx [79]. Next, a review of the literature pertaining to the application of metaheuristic approaches is presented.

A.2.3 Metaheuristic Approaches

Of the metaheuristics implemented in the literature, simulated annealing (SA) and genetic algorithms (GA) are the two most prominent methods of solution to the LP. As such a comprehensive review of these approaches are presented here. First a review of the simulated annealing algorithm and the literature pertaining to the application of simulated annealing to the solution of the layout problem is presented.

A.2.4 Simulated Annealing

A.2.4.1 Fundamental Premise of the Algorithm

The simulated annealing algorithm is based on two fundamental outcomes of statistical mechanics. These observed outcomes are as follows [59]:

1. When the thermodynamic balance is achieved at a given temperature, T, in the physical system, the distribution of the energy states becomes a Boltzmann distribution at this temperature.

2. As absolute zero temperature is approached, the physical system approaches its minimum energy configuration.

A.2.4.1.1 <u>The Metropolis Algorithm</u>

The algorithm emulates the first of these outcomes through the implementation of the Metropolis algorithm [122]. At each iteration the Metropolis algorithm produces a sequence of solely predecessor-dependent configurations (i.e. a Markov chain) that approach the system's thermodynamic balance at the current temperature, T, being considered. Applied to the LP, this is analogous to the generation of a sequence of layouts or layout plans that approach a Boltzmann distribution of the layouts or layout plans for that temperature [59]. From here forth, anytime that configuration appears it can be understood as being analogous to stating layout or layout plan.

The Metropolis algorithm achieves this sequence using a probabilistic hill-climbing technique, which enables it to escape from regions of local optima by accepting neighboring solutions (S'), inferior or superior, with the following probability:

$$P\{S \rightarrow S'\} = \begin{cases} 1 & \text{if } \Delta\mathrm{Q} \leq 0 \quad \text{(downhill move)} \\ e^{-\Delta Q/T} & \text{if } \Delta\mathrm{Q} > 0 \quad \text{(uphill move)} \end{cases}$$

where Q is the objective function that defines a configuration's quality, ΔQ the difference between the neighboring configuration and the current configuration's (i.e. neighbors predecessor) quality, and T the current temperature [170]. The sequence is constructed through the repeated perturbation of the predecessor configuration and subsequent observation of this Metropolis rule of acceptance [59].

According to this rule of acceptance, any neighbor configuration that is of superior quality to the current configuration (i.e a downhill move) is accepted. The rule also establishes that the probability of accepting a neighbor configuration of inferior quality (i.e. an uphill move) is non-zero and instead a function of both the current annealing temperature, T, and the quality degradation, ΔQ, that would result from such a move.

The rule's dependency on quality degradation is as follows. For a neighbor configuration yielding a large degradation in quality, the probability of its acceptance is diminished. This bears a strong resemblance to the philosophy of natural selection where a less inferior configuration would have a better chance at reproducing offspring than would a more inferior one.

Its temperature dependency on the other hand is as follows. A high temperature would yield a probability of accepting an inferior configuration close to unity. As such, the algorithm would then accept the majority of moves with only a marginal bias towards accepting superior configurations. Under this temperature condition, the algorithm randomly walks through the configuration space, therefore encouraging exploration. On the contrary, a low temperature produces a near zero probability of accepting an inferior configuration. Under this temperature condition, the algorithm discourages exploration and instead encourages the continual improvement of the configuration's quality. At intermediate temperatures, the algorithm alternates between exploring the space and refining the configuration allowing for the algorithm to identify and escape regions of local optima effectively. From this discussion, it can be understood that the evolution of the temperature will contribute significantly to how effective the algorithm is at searching the configuration space for the global optimum.

A.2.4.1.2 The Role of the Annealing Schedule

The second of these statistical mechanic outcomes is emulated through the implementation of an annealing schedule, which also addresses how the aforementioned temperature should evolve. The annealing schedule, or temperature control scheme, provides convergence towards the global optimum configuration by adjusting the temperature intelligently as the algorithm proceeds. The search time required for this convergence and furthermore its accuracy are direct byproducts of how the annealing schedule is constructed.

A.2.4.2 Annealing Schedules

A.2.4.2.1 The Classical Annealing Schedule

The initial formulation of the SA algorithm incorporated the classical annealing schedule, which applies a linear temperature reduction scheme. In its basic form, the annealing temperature evolves according to the following relationship: $T_i = \lambda T_{i-1}$ where λ is a fixed ratio (always less than unity to ensure a monotonically decreasing temperature) and is often recommended by the literature to be set to a value of 0.85 [170]. For the initial temperature, T_0, it is set according to the following relationship: $T_0 = \Delta_{UHavg}/\ln P$ where Δ_{UHavg} is the average uphill quality change for the initial series of uphill configuration moves and P is the initial probability of accepting inferior configurations, which is chosen to be a value close to unity, but certainly not unity [34].

The larger the fixed ratio is, the longer the annealing process will take. This presents a dilemma. On the one hand, a longer annealing process will often produce more accuracy convergence results as it has more time at higher temperatures to explore the space and avoid becoming trapped in a local optima region as discussed earlier. On the other hand, a longer annealing process is analogous to a longer running time before convergence. The excessive convergence times produced by the classical annealing schedule is its major drawback and why researchers have since sought annealing schedules that improve the SA algorithms overall effectiveness [170]. Two of the more proponent schedules proposed in the literature for controlling the temperature include the TimberWolf SA algorithm and the Fast-SA algorithm.

A.2.4.2.2 TimberWolf SA Algorithm

Sechen and Sangiovanni-Vincentelli improved upon the classical approach with his TimberWolf SA algorithm, which implements a non-linear version of the classical annealing scheme [151]. They generalized the aforementioned relationship to $T_i = \lambda_i T_{i-1}$, allowing the ratio defining the rate of temperature reduction to be dynamically altered as the algorithm proceeds towards convergence. By increasing this ratio from its lowest value of 0.8 to its highest value of around 0.95 when the objective function was decreasing most rapidly before then progressively returning it to its lowest value, Sechen and Sangiovanni-Vincentelli were able to achieve substantial improvements in both convergence speed and accuracy [151]. This scheme effectively reduces convergence times, yet maintains accuracy by reducing the time spent at the extremes of the temperature range (high and low), and instead spending more time in the intermediate temperatures where there is a favorable balance between both exploring the space and improving the configuration. The success and robustness of this scheme have made it one of the more popular annealing schedules implemented in the general application of SA to the solution of combinatorial optimization problems.

A.2.4.2.3 Fast-SA Algorithm

More recently, Chen and Chang proposed a Fast-SA scheme (FSA) that has three annealing stages to further improve the convergence performance of SA. Their scheme was originally developed while attempting to solve the QAP-B*Tree formulated LP more efficiently by reducing the number of accepted uphill moves in the early stages. The three stages of their scheme are as follows [47]:

1. An initial high-temperature random search stage

2. An intermediate pseudogreedy local-search stage

3. A concluding hill-climbing search stage

The first stage of their algorithm sets the temperature to a large value as to avoid the algorithm from becoming trapped in a local optima region [47]. This allows it to perform a random search of the configuration space to more effectively discover the global optimum. In the second stage, the temperature is allowed to approach zero so as to perform a pseudogreedy local-search of the configuration space [47]. By accepting increasingly fewer inferior configurations it promotes the improvement of the configuration towards that of the global optimum. In the third and final stage, the temperature is abruptly increased then gradually reduced until convergence. It has already been well-established that at higher temperatures the algorithm is more capable of exploration as it will accept more inferior configurations. As such, the goal of this abrupt temperature rise is to facilitate the search for better configurations. With the initial stages reducing the number of iterations to explore the configuration space, more time can be sent in this third and final stage to help improve convergence accuracy while also reducing convergence times [47]. To capture this three-stage scheme, Chen and Chang defined the annealing schedule as follows:

$$T_n = \begin{cases} \dfrac{\Delta_{UHavg}}{\ln P} & n = 1 \\ \dfrac{T_1\langle\Delta_{avg}\rangle}{nc} & 2 \leq n \leq k \\ \dfrac{T_1\langle\Delta_{avg}\rangle}{n} & n > k \end{cases}$$

where n is the number of iterations, Δ_{UHavg} and P (the latter set as 0.9 in Chen and Chang's study) are defined identically to before, T_1 is the initial temperature, $\langle\Delta_{avg}\rangle$ is the normalized average quality change for the current temperature, and c and k are user-specified parameters.

The first iteration ($n = 1$) makes up the first stage, which is none other than the classical and TimberWolf SA method of defining the initial temperature. The second stage follows and proceeds until the kth iteration ($2 \leq n \leq k$). The function of c is to control how low the temperature is during this stage. Since a temperature that approaches zero is desired, for reasons stated before, its value should be chosen to be large (a value of 100 was used in their research). The duration of this stage is dictated by the users choice of k. Its value is directly proportional to the problem size and therefore can be determined accordingly. The smaller the problem size is, the smaller the k value can be, such that it doesn't impact the algorithm effectiveness. In Chen and Chang's application of the algorithm to the LP, they set $k = 7$ with great success for problem sizes ranging from 100 to 300 blocks [47].

Upon completion of the second stage, the temperature jumps up as the temperature control parameter, c is dropped from the temperature updating function. In the last two stages the $\langle\Delta_{avg}\rangle$ acts as the temperature reduction ratio. When this average change is significant, the ratio becomes larger and the temperature reduces at a slower rate. On the contrary, when the average change is smaller the ratio is reduced and the temperature reduction accelerated. This is a favorable behavior as it enables more time to be spent during periods of large improvement and less time during those with little.

Since Chen and Chang's introduction of FSA, it has become a particularly popular choice by researchers solving the QAP formulated LP, due in large part to its frequently observed improved performance over the classical and TimberWolf SA algorithms in solving said problem. Although its original application was in association with the QAP-B*Tree formulation of the LP, it has since been applied to other formulations of the LP and other COPs in general with great success.

A.2.4.3 Perturbation Schemes for Generating Neighboring Configurations

In addition to the cooling schedule, the heuristic rules implemented to generate neighboring configurations also play an important role in the algorithms effectiveness in discovering the global optimum configuration. The heuristics implemented vary slightly and according to the underlying model structure and problem formulation.

All perturbation schemes implemented by researchers in the literature include swapping of some form. Swapping refers to the act of interchanging the position of any two blocks. In a structured formulation, the cells in which blocks appear are swapped whereas in the more generic unstructured formulation, the position in which the two blocks appear in the representation sequence are swapped. Two of the more relevant implementations in the literature that demonstrate the latter are Chen and Chang's and Tang's applications of SA to the SLP [47,159]. Chen and Chang's QAP-B*Tree formulation implemented a swapping procedure that involved node swaps of two randomly selected nodes in the representation. Tang's QAP/U-SP formulation implemented a similar swapping procedure for the SP structure that first required one of the sequences in the pair to be selected with equal probability, before a block was then randomly selected and it and an adjacent block swapped. As observed, the heuristics are much the same, but the procedures differ slightly as a result of the underlying model structure. Both performed this swapping procedure with a predefined fixed probability for swapping two selected blocks whose selections were performed without bias (i.e. each block had an equal probability of selection).

To further facilitate perturbation of the configuration, both also implemented a rotation procedure. Rotations are unique to unstructured layout formulations involving non-square blocks and/or I/O point present designs. The relevance of rotation in the former case is easy to understand by inversion. Rotating a square does not result in a spatial change in the layout as all sides are equal. As such, the centroid will remain in the same position and all other blocks present uninfluenced. In the case of a non-square block, this no longer holds true. The latter's relevance can be understood by recognizing that when an I/O point is present each rotated position, regardless of whether it is square or not, will result in a unique I/O point position and therefore MHC value. Both Tang and Chen and Chang sought to handle VLSI designs with non-square blocks present, therefore rotation procedures were also required for complete optimization of their respective problems [47,159]. Rotation for both involved randomly selecting a block, and in the case of Tang's formulation selecting an unconstrained block, for rotation and then rotating said block with a predefined fixed probability. Tang also established that the probability of swap should be greater than that of rotation [159].

Handling a DLP problem requires the above procedures to be extended to encapsulate the now multi-period nature of the problem. Baykasoglu and Gindy, being the first to adapt SA to the DLP, established the standard swapping procedure for the QAP/S formulation of the DLP. They proposed a random descent pairwise exchange swapping procedure. In this procedure, an unbiased period selection forgoes the then unbiased random selection of two blocks for a predefined fixed probability of swap [23]. Sahin et. al. also implemented such a perturbation scheme for the budget constrained problem [148].

McKendall, Shang, and Kuppusamy proposed a more complex derivative of this perturbation scheme that implemented a look-ahead/look-back strategy. In the scenario that the neighbor configuration is accepted by the algorithm, the procedure then proceeds to consider accepting the same block swap in preceding and succeeding periods (t-1 and t+1), once more according to the metropolis rule of acceptance. For each swap that is accepted the process continues backward or forward in time respectively until either a swap is not accepted or the first or last period is reached [121].

Their perturbation scheme demonstrated best in class performance for the literature's standard 48 test problems at the time, consistently outperforming SAs implementing the basic scheme noted before. Furthermore, they observed that the improved convergence properties did not come at a computational cost. The reduced randomness of their heuristic enabled it to outperform the basic heuristic perturbation scheme while requiring the same or less computational time despite the added computational overhead associated with the additional interchange evaluations required at each iteration by the look-ahead/look-back procedure. The superior performance of McKendall et al. perturbation scheme for the DLP formulated as a QAP and relative ease of implementation makes it an attractive option for future implementation in this dissertation.

Due to the increasing difficulty of solution, little research has applied SA to the QAP/U DLP problem and less so to the MIP DLP problem. The former can be relatively easily handled by fusing the work of Tang and Chen and Chang discussed earlier with that of the more recently discussed work performed by McKendall et al. to capture both the unstructured and dynamic nature of the problem. The latter on the other hand requires a fresh set of heuristics to handle the continuous nature of the layout. Dong et al. have been one of a few to tackle the development of the necessary heuristics required to solve such a MIP DLP problem. They proposed a free-space searching rule that identifies space available for machine placement by searching the space from the top-left to the bottom-right corner according to fixed step lengths no larger than the machine edge length. Furthermore, they implemented machine adding and removal heuristics to address new asset integration scenarios [57]. Their procedure proved to be a viable approach to solving the MIP DLP problem. With few researchers having studied the MIP DLP problem in the literature, this gap is one the current research intends to address and further explore.

A.2.4.4 Applications of Simulated Annealing to the LP

Having since developed a thorough understanding of the simulated annealing algorithm, the role that the annealing schedule and perturbation scheme have in the optimization process, and furthermore the various forms of these implemented in the literature, a survey of the more notable applications of SA to the LP will be presented. On the static side of the problem, Tang used a SA approach implementing a classical annealing schedule to solve the VLSI problem formulated as a QAP/U-SP. Although using the less effective classical annealing schedule, Tang achieved acceptable convergence for a boundary constrained problem size of 49 blocks with range fixed blocks in under 30 seconds [159]. Chen and Chang similarly solved the VLSI problem formulated as a QAP/U, but with a B*Tree representation and in the absence of fixed blocks. They introduced the Fast-SA algorithm to solve the VLSI problem with great success, demonstrating the superior performance of the Fast-SA algorithm over existing annealing schedules [47].

As noted earlier, Baykasoglu and Gindy were the first to adapt SA to the DLP. They adopted a variant of the TimberWolf's annealing schedule that employed Bennage and Dhingra's definition for the cooling rate [28]. Like the TimberWolf annealing schedule, the cooling rate as defined by Bennage and Dhingra changes as the optimization progresses. The major difference is that theirs does so dynamically according to the final acceptance probability after each iteration. In other words, the more neighboring configurations that are accepted after each iteration, the higher the final acceptance probability will be. At the time, Baykasoglu and Gindy's work based on Lacksonen and Enscore's formulation of the non-budget constrained LP [106] was best in class, outperforming both Rosenblatt’s DP and Conway’s GA methods for the literature’s benchmark 48 DLP test problems [23].

Later McKendall et al. employed the same annealing schedule, but with updated perturbation heuristics. These updated perturbation heuristics enabled them to achieve best solutions for 35 of the 48 DLP test problems, 12 more than the next best heuristic solution method at the time (2006). It outperformed, in terms of solution quality, the likes of Baykasoglu and Gindy’s SA and Erel et al.’s DP methods as well as the hybrid GA and hybrid ACO methods used by Balakrishnan et al. (2003) and McKendall and Shang respectively [121].

Sahin et. al extended the Lacksonen and Enscore formulated DLP to encapsulate budget constraints thereby matching Baykasoglu et al.’s formulation of the same problem [148]. Being one of only a few to have addressed the budget constrained DLP and after consistently outperforming Bayaksoglu et al.’s ACO solution method for the same problem, it remains best in class in solving the 48 DLP test problems under budget constraints. As such, their method will become the QAP formulated baseline for the current research.

Dong et al. extended the DLP, however in a different capacity from Sahin et al. and Bayaksoglu et al. Instead of incorporating budget constraints, they addressed two additional gaps in the literature. They first addressed the absence of machine addition/removal in each period, thereby accounting for scenarios of asset expansion or downsizing. Furthermore, they addressed solving the DLP as a MIP. This in turn enabled them to evaluate continuous layout representations. Both these characteristics furthered the complexity of the DLP significantly, requiring them to develop advanced heuristics for perturbing the layout configuration. Converting the problem into a shortest path problem and then solving it with an auction algorithm internal to the SA algorithm allowed them to achieve reasonable results for the non-budget constrained MIP DLP [57]. Dong et al.'s research concludes those applications of SA to the LP that are of most relevance to the problem being addressed in this research. This concludes the discussion of applications of SA to the LP. Next, a review of the genetic algorithm and the literature pertaining to its application to the solution of the layout problem is presented.

A.2.5 Genetic Algorithm

A.2.5.1 Fundamental Premise of the Algorithm

To solve optimization problems, genetic algorithms evolve a population of individuals belonging to the problem's search space. Evolution of the individuals in the population emulates Darwin's principle of natural selection. In other words, individuals in the population that are of superior fitness have a better chance of being selected for reproduction of offspring individuals or of surviving to become a member of the next generation's population. This process of evolution is simulated as successive iterations, called generations, until a termination, or convergence, criterion is met. If evolution of the population is performed effectively the algorithm will arrive at the fittest (i.e. best) individual in the problem's search space.

To facilitate the evolution of the population from one generation to the next, a series of genetic operations are performed on the individuals of the population. Individuals that these genetic operators are performed on are referred to as parents and individuals produced as a byproduct are referred to as its/their offspring. As operations are applied, resulting offspring can potentially become parents of the next operation. This process constitutes the procedure, often referred to as genetic reproduction, for facilitating the evolution of the population by the algorithm. The genetic algorithm as a whole can be decomposed into three phases. The phases composing the GA are as follows:

Basic Stages of a Genetic Algorithm

1. Population Initialization
2. Genetic Reproduction (Genetic Operators)
    a. Selection Operator
    b. Variation Operator
        i. Crossover
        ii. Mutation
3. Termination

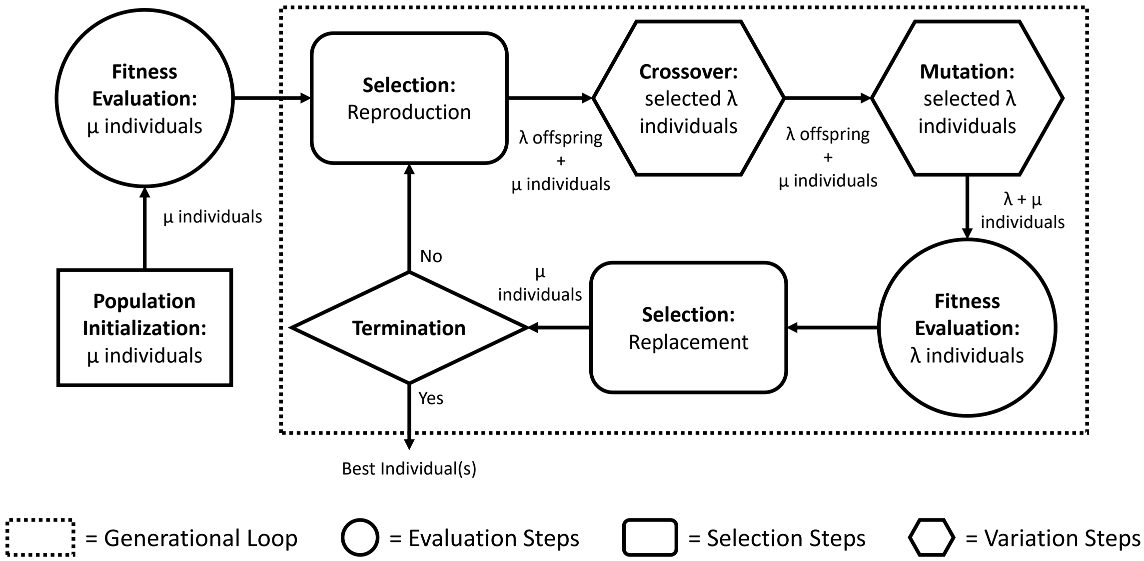


Figure 74 – Notional genetic algorithm [58]

The middle phase of the algorithm constitutes the evolutionary process. In conjunction with the termination phase they comprise what is often referred to as the generational loop. This loop applies the genetic operators to the population in an iterative process, where each iteration establishes a new evolutionary generation, until a desired termination criterion is met. Figure 74 graphically demonstrates the algorithms structure including the generational loop, which is represented by all that falls within the dotted rectangle [58].

A.2.5.1.1 <u>The Importance of the Data Structure Representing an Individual</u>

Before the above described procedure can be performed to solve the problem, the data structure representing an individual in the population, or also referred to as encoding must be defined intelligently. Conway and Venkataramanan stressed the importance of establishing this encoding by stating that "the data structure chosen to represent the population member (i.e. individual) is essential for effective convergence of the method"

[48]. Furthermore, the encoding of the individual closely dictates how the variation operators modify an individual of the population [58]. As such, selection of the encoding often precedes defining the variation operations methods of alteration.

Encoding of an individual usually takes the form of either a binary string or vector of integers or real numbers. A binary string representation is the popular choice as it provides the maximum number of schemata per bit [137]. The choice of researchers solving the LP is often the vector representation however, as it is most compatible with the format of the data that must be represented. In solving the QAP formulation of the LP, a vector of integers is often implemented, with each bit of the vector representing the placement in the layout and the integer value assigned to it the object placed there. With a QAP/U formulation of the LP where rotations of the object may need to be accounted for, additional bits representing the rotational representation of the objects are included. When handling the MIP formulation of the LP, the vector may become a mixture of integers and real numbers, where the integer bits may correspond to rotational representations and the real numbers, the continuous positions of the objects in the layout. Observing how the underlying problem modeling impacts the encoding, a better understanding of the prior stated variation operators' dependency on the encoding structure can be established.

A.2.5.1.2 Phase I: Population Initialization

Initialization of the population is also essential to effective convergence by the GA. When little is known of the problem, the initial population is often formed by randomly generated individuals [137]. Diversity of the population is an important quality and the random generation method inherently excels at this. In evaluating the impact that population diversity has on the performance of GA, Diaz-Gomez and Hougen have observed that the major drawback of this approach is that when most of the solutions fall in the neighborhood of a local optimum the GA will become trapped in this region. As such, they declared that diversity alone is not enough, but rather a healthy or “good” diversity in the search space is essential to avoid poor performance by the GA [56].

Diversity in the initial population is not the sole attribute of importance. In fact, too much diversity can lead to excessive run times by the GA. Individual viability and global optimality are also essential attributes to consider. With a purely random generation method being less than effective at ensuring these, many researchers have sought to employ a priori knowledge of the problem to generate an initial population. The major limitation of this approach is that the necessary knowledge of the problem must first be known, but in the presence of said knowledge the benefit of leveraging it to generate a well-formed initial population can be substantial. As has been observed, convergence is often faster for populations generated by this method due to the GA being supplied with a better formed initial population [137]. Care must be taken in ensuring that this population does not become too diversity-deprived by generating a population that is dominated by a particular region of the search space.

Some researchers have implemented a hybrid approach that combines these two methods in an attempt to further improve the performance of the GA. Leveraging the random methods superior diversity property and the improved viability/global optimality properties of the knowledge-based methods, some researchers have diverged from the conventional single population GA in favor of a multi-population one. This approach both initializes with multiple populations and also evolves each of these sub-populations independently making the GA more capable of exploring far reaching solutions in the search space [46]. The effectiveness of this approach over other methods was proven by Toledo, Ribeiro de Oliveira, and Franca while implementing such a hybrid multi-population GA to solve sets of hard and large lot sizing problems with backlogging [162]. Despite its notable advantages (superior solution quality and convergence times), few, with the exception of Pourvaziri and Naderi, have applied a hybrid multi-population to the DLP [140]. A contributing reason for this scarcity of application to the problem can be acknowledged as being the result of the concepts relatively short history, being introduced only recently.

A.2.5.1.3 Phase 2: Genetic Reproduction (Genetic Operations)

Evolution of the population from its initial state by the application of genetic operations plays a significant role in the GA's effectiveness in solving the problem. To facilitate this evolution, selection and variation operators establish how this evolution will proceed and therefore directly impact the GA's ability to converge on the global optimum and do so efficiently.

A.2.5.1.3.1 Selection Operators

The primary functions of selection operators are to establish the individuals for reproduction and survival [58]. The former function, more commonly referred to as just *selection*, determines which individuals in the population will reproduce and how frequently. The latter function, defined as simply *replacement*, regulates the population size from generation to generation by determining which individuals will become members of the next generation [58].

Selection operations (selection or replacement) are generally driven by the fitness of an individual relative to all others in the population. As such, an individual's tendency of being chosen for reproduction or survival is dependent on its relative fitness to the remainder of the population. How this tendency is determined depends on the selection operation technique implemented. Three common selection techniques implemented in the literature are proportionate (roulette wheel selection and stochastic universal sampling methods), tournament (deterministic and stochastic), and truncation selection with the latter also being capable of acting as a replacement technique. Replacement techniques include, generational, truncation, and steady state replacement. Elitism also often falls under this category as it functions in a selective manner. Many elitism strategies exist, where the common practice among them is to copy a desired amount of most fit individuals to the next generation. If desired, one may refer to the appendix for a discussion summarizing the various types of selection and replacement techniques generically employed in the literature.

A.2.5.1.3.2 Variation Operators

The function of variation operators is to promote effective evolution by transforming the individuals of the population such that the search space is sufficiently explored and areas of optimality are thoroughly exploited. In their absence, the population would converge to a population of identical individuals of just that of the best individual present in the initially supplied population. To facilitate transformation in an attempt to more thoroughly search the space for better individuals, variation operators that are classified as either crossover or mutation operators are implemented.

Crossover operators emulate the biological process of reproduction by generating one or multiple offspring through a combination of two or several parents chosen by the earlier described selection operations. The offspring generated inherit partial characteristics from each parent involved. Mutation operators encourage diversity in the population as these operators modify an individual to form another that is often within its proximity [58]. Mutation can considerably improve the quality of individuals discovered compared to crossover when in the vicinity of the optimum. Furthermore, its preservation of diversity enables it to stave off the negative effects that both a strong selection pressure and genetic drift can have. Establishing an effective mutation rate is essential as too high a rate can lead to an evolution that is no more guided than a random walk of the space. In the opposite extreme, too little diversity in the population can greatly reduce the chances of sufficiently searching the design space and accurately identifying the global optimum.

As noted before, the method of variation implemented is dependent on the encoding structure of the individuals in the population. For binary or integer vectors there exists three crossover techniques. These include, single point, two points, and uniform crossover. On the mutation front, bits of the string are modified randomly with a low probability on the order of 0.01 to 0.1 per individual [58]. Commonly implemented mutation techniques include deterministic or bit-flip mutation. When handling real vectors, the techniques become more involved. In the general sense the binary crossovers above can be extended by exchanging the real components of the parents involved. These are often referred to as discrete recombination methods. The limitation of this would be that the domains of both those exchanged would need to be the same, if not the unit value in each domain could be exchanged instead. Additional crossover techniques include voluminal BLX-α and linear BLX-α (also referred to arithmetic crossover or intermediary recombination). Mutation approaches include uniform, Gaussian, Gaussian and the 1/5 rule, and self-adaptive Gaussian mutation with the latter being the preferred method [58]. Ordinal representations (integer vectors), much like that in which would be observed while handling the QAP/U-SP formulation of the problem, implement methods of crossover and mutation similar to the binary representation procedures, but with the added requirement of sequence uniqueness. Ensuring that a vector sequence is feasible is an important task as will be discussed later in this dissertation

A.2.5.1.4 Phase 3: Termination

Termination of the GA simply involves identifying when the GA has converged sufficiently close to the expected optimum value. Many methods have been implemented in the literature to establish convergence.

A.2.5.2 Genetic Reproduction Methods Applied in the Literature

Several genetic reproduction methods have been applied to facilitate the solution of the LP by GA. For the most part the majority of these reproduction methods implement both crossover and mutation variation operators to promote the effective evolution of the population and in turn the convergence of the algorithm towards the global optimum solution for the LP being studied. As will be observed, with time these methods have themselves improved, becoming more effective at evolving the population.

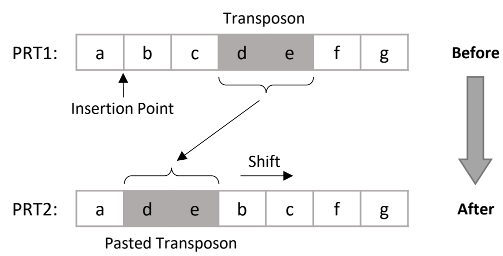
Conway and Venkataramanan, being the first to adapt GA to the QAP/S formulated DLP, set the standard for evolving the population. Conway and Venkataramanan implemented a genetic reproduction scheme employing crossover as the primary variation operator. To select individuals to perform crossover on, roulette wheel proportionate selection was implemented. Once individuals were selected, a splicing position was randomly chosen and the string split. Substrings to the right of the splice point were then swapped. If the split occurred in the middle of a period layout, then the stronger of the two strings was retained with the weaker string filling in the unassigned bits of the string such that layout feasibility remained unviolated. Mutation was also implemented, but in a rather limited capacity. For every so many cross-breedings one of the parents was slightly altered before breeding occurred. Furthermore, Conway and Venkataramanan implemented elitism by allowing for the most fit individual, or as they called it the queen bee, of the population to survive into the next generation [48].

Balakrishnan and Cheng's approach to reproduction sought to increase search space exploration and population diversity. To increase search space exploration Balakrishnan and Cheng implemented a point-to-point crossover technique. After randomly selecting two individuals in the population to become parents, $2(nt-1)$ offspring are generated by interchanging the departments in the parents position by position in a successive compounding manner until the last position in the vector is reached. Here n defines the number of departments and t the number of periods. After eliminating all illegal offspring (those that have duplicate departments in the same period), the best viable child is then selected to replace the worst parent in the population. To improve diversity, mutation of this best cross-bred child is performed with a low probability. A random selection of the period and two departments for interchange encompass their method of mutation. To further facilitate diversity in the population, a replacement scheme that periodically replaces the least fit individuals in the population with randomly created new individuals before proceeding onto the next generation is implemented [13].

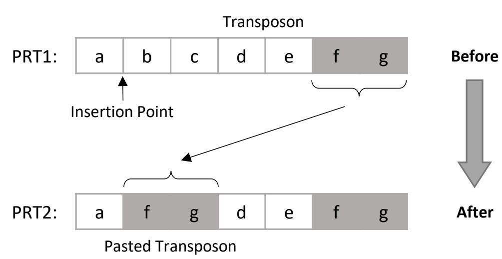
Balakrishnan, Cheng, Conway, and Lau proposed a reproduction scheme for the QAP/S LP that infused heuristics and DP to facilitate improved evolutionary properties. In Balakrishnan et al.'s (2003) implementation, tournament selection was employed to first select *s* individuals from the population to become members of a crossover parent pool. They observed that an *s* equal to ten was sufficient for effective performance. After selection, each individual's string was then cut at the period joints into *P* (# of periods) parts (i.e. layouts) and any duplicates were discarded. After collecting all the unique parts, DP was implemented to generate the best combination from these parts, the result becoming the offspring of the crossbreeding. To promote diversity, the offspring was mutated according to a random Bernoulli test and with a low probability. To mutate the offspring, Computerized Relative Allocation of Facilities Technique (CRAFT) was used to heuristically improve a randomly chosen period of the offspring by performing pair-wise exchanges of the department locations. The replacement technique implemented by Balakrishnan et al. (2003) considered uniqueness by only replacing the weakest parent in the parent pool with offspring if it were to be a unique individual in the parent pool [15].

Much like that of Conway and Venkataramanan, Ripon et al. employed a single point crossover method and furthermore a mutation technique resembling that of Balakrishnan and Cheng's implementation. In addition to these two variation operations, Ripon et al introduced the concept of jumping genes to the solution of the QAP/S DLP

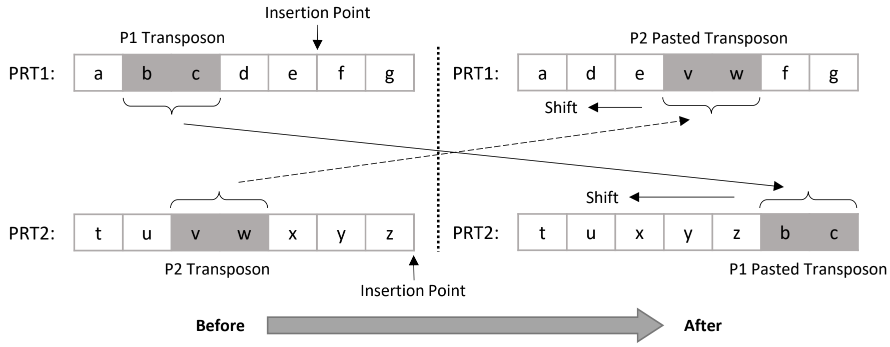
[143]. Emulating the biological occurrence observed in chromosomes by Nobel Laureate Barbara McClintock and first proposed as another variation operator of GAs by Man et al. [119,117], this operator employs cut and paste and copy and paste transposition operations to enable the population to evolve more effectively. To apply these operations, two individuals in the population are first selected randomly along with which of the transposition operations will be applied. Additionally, the selection of the individuals is not limited to two unique individuals, the same individual can be chosen twice. In the latter case, the operations are simplified given they are applied to just one individual as depicted in Figure 75. Figure 75 depicts the general process involved depending on which of the operations is chosen [143].



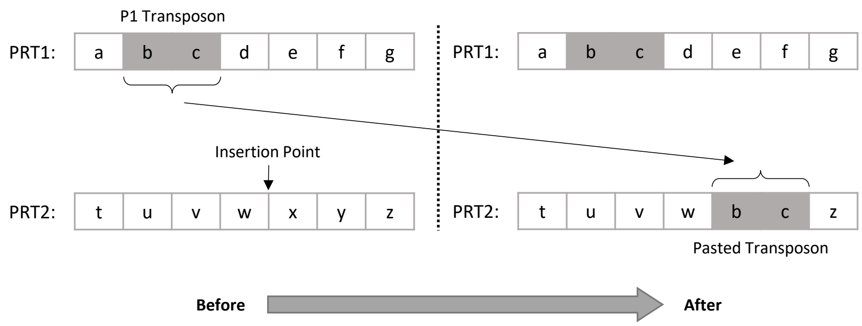
a) Cut Paste – Same Chromosome



b) Copy Paste – Same Chromosome



c) Cut Paste – Different Chromosome



d) Copy Paste – Different Chromosome

Figure 75 – Jumping gene transpositions [143]

Pourvaziri and Naderi proposed a novel crossover method that generates only feasible offspring. Their method has the significant advantage of no longer needing to perform a check of feasibility, as has always been required by those before. To achieve this, first two individuals and a cross point for each period are randomly selected. Then at each of these cross points the sequence preceding the cross point are copied from the first parent to the offspring. Then the remaining bits in the offspring are assigned according to the order that the departments not copied over from parent one appear in parent two. This crossover is depicted in Figure 76 and demonstrates its ability to guarantee feasibility. To induce diversity in the population, Pourvaziri and Naderi implement a mutation operation that interchanges two randomly chosen periods [140].

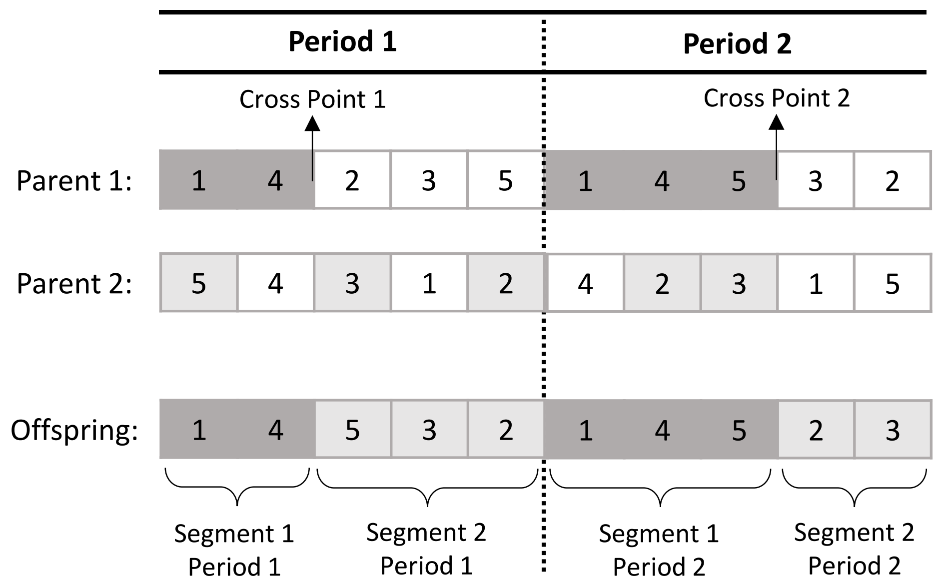


Figure 76 – Example of Pourvaziri and Naderi's feasible offspring guaranteeing crossover method [140]

Liu and Meller, in solving a MIP formulation of the SLP by combining it with a QAP/U-SP model, proposed a modified order-based uniform crossover operator to handle the position-pair based encoding scheme of the QAP/U sequence pair model representation that the GA would operate upon [115]. The first stage of the method follows that of the unmodified uniform cross operator where a randomly generated binary bit string is used to distinguish which pairs from parent one are copied to the first offspring (those with a bit value of one) and which from parent two are copied to the second offspring (those with a bit value of zero). The second stage diverges from the unmodified operator in order to account for the position-pair encoding structure. Just like that of the unmodified procedure, the new position-pairs for the remaining genes in each offspring are determined by their relative orders in the other parent. Instead of directly copying these to the voids in the offspring, the remaining position-pair genes are defined according to its parent's positions for these relative orders before then being assigned. The process is quite involved; as such, to avoid an exhaustive discussion of its nuances one may refer to Liu and Meller's original paper, if desired. Furthermore, a modified form of this process is adopted as part of this dissertation's implementation so one may also refer to this later discussion for some further insight into Liu and Meller's process. Liu and Meller also implemented a mutation operator by randomly exchanging two chosen position-pairs for a selected individual [115].

Up until this point, reproduction methods applied to the QAP formulated DLP have been discussed. GAs have also been applied to the MIP formulation of the LP. For that reason, researchers have developed methods of handling such a problem formulation and the real vector representations that accompany them. Dunker, Radons, and Westkamper are one such example of researchers who applied GA to the MIP formulated DLP. To evolve the population, Dunker et al. (2003) applied a version of Chan and Tansri's order crossover operator [45]. After selecting two individuals as parents, two cut positions are randomly selected and the genes falling between these positions are copied from one parent to one of the offspring. The remainder of the genes are filled with the numbers from the other parent not already in the offspring and according to the order they appear in the other parent. The same procedure, but with opposite parents, is performed to create the other offspring. To improve diversity of the population Dunker et al. (2003) implement a mutation operator that exchanges two random genes in a randomly selected parent for each of the position vectors independently (x and y position vector strings) [61].

Mazinani, Abedzadeh and Mohebali also developed reproduction methods for a GA applied to a MIP DLP. Mazinani et al. applied a continuous uniform crossover operator to evolve the population. Using a roulette wheel selection method two parents are selected to perform crossover on. After the random generation of bit string whose length matches that of the individuals and whose bits are uniformly distributed between zero and unity, the two children are created. The first child is assigned its genes as the positions that are at the bit value between the two parent positions and the second at the position one minus the bit value between the two parents. For mutation, Mazinani et al. applied three separate mutation operators. The first replaces a randomly selected gene with a value in its domain according to a predefined function. The second involves randomly selecting three genes in a period and performing binary and triple exchanges until a most fit individual is established. This resembles that of the CRAFT heuristic, but with triple exchanges. The third method randomly selects a period and performs pair-wise exchange, much like that of CRAFT, to establish the most improved individual [118].

The preceding sections summarize several different reproduction techniques employed in the literature to solve the LP for a variety of formulations (QAP/S, QAP/U, and MIP). As was established prior and supported here, the method of variation is driven by the problem formulation, more specifically the encoding structure employed to represent an individual in the population. The sub-section to follow will explore those applications of GA as the sole method of solution to the LP and that employ some of the aforementioned reproduction techniques. These are not an expansive list of those that have been proposed in the literature, but these are those that have demonstrated superior performance and, at the time, novelty to solving the LP.

A.2.5.3 Applications of Genetic Algorithm to the LP

Several researchers over the years have applied a genetic search heuristic, or simply a GA, to solve the LP. Motivated by the need to handle multiple constraints (e.g. budget constraints) and non-linear or non-convex objective functions, Conway and Venkataramanan became the first to adapt GA to the DLP. Conway and Venkataramanan sought to solve the budget constrained DLP formulated as a QAP/S using a GA employing roulette wheel selection, single point crossover, mutation, and elitism. This genetic reproduction scheme along with GA proved to be an effective method of solving such a LP, after producing solutions within 1.05% of optimality on average using a randomly generated initial population. This was on par with Rosenblatt's DP reported 1.1% for the same experiment involving nine departments [48].

Balakrishnan and Cheng sought improved GA performance in solving the QAP/S formulated non-budget constrained DLP using mutation in a larger capacity compared to Conway and Venkataramanan and by implementing a diversity promoting generational replacement approach. Furthermore, a new crossover operator was proposed to promote better search space exploration. Implementing a point-to-point crossover technique, a best cross-bred offspring mutation method, and a periodic replacement technique, they were able to effectively solve the DLP for a set of 48 DLP test problems. These 48 problems consisted of 6 series of 8 problems, which consisted of 6, 15, and 30 departments by 5 and 10 periods [13]. These 48 problems proposed by Balakrishnan et al. (2000) have become the benchmark for QAP formulated DLP solution methods in the literature. For these 48 problems, Balakrishnan et al.'s (2000) GA outperformed Conway and Venkataramanan on most of the problems with the exception of the 15 department, 10 period problem set, interestingly [13]. Overall though, Balakrishnan et al.'s GA proved to be quite effective in solving the problem.

Dunker et al. (2003) were a few of the earlier researchers to have applied GA to a MIP formulation of the DLP. Employing a two-point crossover operator and basic mutation operator for the DLP, Dunker et al. (2003) were able to outperform previous methods solving such a problem at the time. They also, further extended the problem and their GA to encapsulate grouping of departments into areas by solving both the grouping problem and the sub-problem of determining the best layout for the departments in each group. Dunker et al. (2003) concluded that the approach had a genuine ability to solve large size problems within a reasonable duration of time [61].

Liu and Meller later combined a QAP/U-SP model with GA to efficiently solve the MIP formulation, based on Sherali et al.'s model, of the SLP. The design of Liu and Meller's GA evolved the SP model population by genetic variation operators formulated specifically for the position-pair structure of the SP encoding. For each of these SP encoded individuals searched by the GA, the binary variable values in the MIP model are set [115]. This greatly reduces the effort then required to solve the MIP formulation of the problem as with all binary variables set, the problem can then be reduced to a linear programming problem. As observed earlier while discussing exact methods, solving a MIP formulation of the LP with linear programming is a highly effective approach. Solving the linear programming model to optimality then provides the optimal layout for the SP that is also MIP compliant and optimal for the set binary variables (optimal on a positioning basis). By searching the SP space with GA and solving the MIP problem as described above, Liu and Meller were able to achieve reasonable optimality results for small problems while greatly improving the solution quality and times for larger problems when compared to other heuristic approaches available at the time [115].

In addition to Liu and Meller's and Dunker et al.'s (2003) applications of GA to a MIP formulation of the problem, Mazinani et al. also applied GA to such a formulation. Modeled as an extension of Konak et al.'s MIP SLP formulation with flexible bays and by implementing a continuous uniform crossover and three mutation operators, Mazinani et al.'s GA based solution procedure was able to outperform Konak et al. [95] and others for a series of SLPs. Furthermore, their GA performed as well or better in solving the discrete DLP that Conway and Venkataramanan's GA and Baykasoglu and Gindy's SA algorithm. Comparisons to the General Algebraic Modeling System (GAMS) solution to the flexible bay MIP DLPs further demonstrated the sheer effectiveness of their GA to solve MIP formulated DLPs [118].

The work of Mazinani et al. concludes the research of direct relevance to the problem being studied in this dissertation and for which the GA was the sole method of solution. As can be observed, some of the research discussed before in establishing GA reproduction methods to solve the LP were not encapsulated in this section. The reason for this is that these reproduction methods were applied to a GA that was also coupled with another solution method. This coupling results in it becoming categorized as a hybrid approach, which is the subject of the section that follows. As such, the accomplishments of the research in which these reproduction methods were implemented will be encapsulated in said section.

A.2.6 Hybrid and Intelligent Approaches

A.2.6.1 Applications of Hybrid and Intelligent Approaches to the LP

Early on Balakrishnan et al. (2003) observed the need to hybridize the GA in order to achieve improved performance in solving the QAP/S formulated DLP. To improve the effectiveness of the GA as the main solution method, Balakrishnan et al. (2003) infused DP and heuristics into the reproduction process to facilitate better evolution of the population. Using DP to discover the best combination of parents from a pool as the crossover operator and leveraging CRAFT to heuristically improve the period layout of an offspring as the mutation operator, Balakrishnan et al. (2003) were able to achieve results on par with Baykasoglu and Gindy's SA for a series of DLP. They were also able to solve a 30 department 10 period problem in a fifth of the time of their SA. With that said, Balakrishnan et al.'s (2003) approach did not outperform Baykasoglu and Gindy's SA from a solution quality perspective for these problems. Though true, it's solutions, where the initial population was generated using Urban's method, continued to outperform considerably Conway and Venkataramanan and Balakrishnan and Cheng's (2000) purely GA approaches [15]. This result demonstrates the benefit of combining GA with another solution method, DP in this case, to form a hybrid approach to solve the DLP. Although it did not outperform Baykasoglu and Gindy's SA for the larger problems, it did consistently provide superior results for the small and medium sized problems. Furthermore, during their investigation, Balakrishnan et al. (2003)

acknowledged Grefenstette's [81] suggestion that a mutation rate of 5% or less be implemented in order to provide a healthy balance of diversity in the population [15].

Baykasoglu, Dereli, and Sabuncu sought to solve the budget constrained DLP formulated as a QAP/S using an ant colony optimization (ACO) algorithm. ACO is an optimization technique that emulates the natural behavior of ants in finding their way from their nest to food sources in the most efficient manner allowable by a collective knowledge of the pheromone trails present. To simulate the artificial ants search of food sources (i.e. local optima), Baykasoglu et al. opted to choose a period and the first department for interchange, randomly. The second department was chosen according to a method resembling that of roulette wheel proportionate selection in GA. A comparison of their ACO approach to solving the non-budget constrained DLP to the literature for the 48 DLP test problems introduced by Balakrishnan and Cheng was performed. For these problems their approach provided competitive solutions and, from a solution time perspective, outperformed SA approaches it was compared to for larger problems. Since budget constrained DLP results were unavailable to them at the time, no comparison to the literature was possible [22].

A refined evolutionary approach called colonal selection algorithm (CSA) has also been applied to the solution of the QAP/S formulated DLP. Ulutas and Islier entertained applying CSA to such a DLP after observing Engin and Doyen's successful application of CSA to COPs [66]. CSA is a biological random search method inspired by the human immune system's self-organizing and distributed response to affinity maturation, or in other words, its production of antibodies in response to antigens [55]. To artificially simulate this type of response in the setting of the DLP, Ulutas and Islier applied a generational cloning process that employed a roulette wheel selection method based on the antibody's (i.e. individual's) affinity, or fitness, value to identify two antibodies for cloning. Following cloning, the affinity maturation process is emulated by subjecting the antibodies of this cloned population to an inverse mutation and subsequently a pair-wise interchange mutation operator. The former simply inverts the order of the individual's sequence between two randomly chosen bits and then, if this is not accepted, the latter mutation operator is applied, which simply swaps two random bits in the sequence [167]. Mutation is performed at a higher rate in this application than in the application of mutation in the earlier GA applications [54]. Comparison of this approach to the literature at the time demonstrated superior performance while solving larger problems, improving upon the best known results for 15 of the 16, 30 department problems of the 48 DLP test problem set. After observing reasonable solution times to these larger problems, Ulutas and Islier concluded that the CSA is not only an effective, but also fast method of solution to the DLP [167].

Hoping to improve the effectiveness of GA as the primary solution method to the QAP/S formulated DLP, Ripon et al. proposed a novel hybrid GA. Their approach incorporated jumping gene operations in addition to the standard crossover and mutation variation operators and furthermore a modified backward pass pair-wise exchange heuristic influenced by Urban's forward pass heuristic [143]. The implementation of the jumping gene operation and benefits of it were acknowledged earlier while discussing GA reproduction methods implemented in the literature. As such to avoid reiteration one may refer to Section A.2.5.2 for more details on the reproduction scheme they employed. Ripon et al.'s approach first employs GA to generate the initial solutions of what would have been before those produced in the forward pass by Urban's method. Then, the modified backward pass heuristic uses these solutions to enhance the solution by only exchanging genes with a common boundary if they provide improvement. Just like that of Balakrishnan et al.'s (2000) backward-pass approach such an approach can only yield a solution as good as, or better than, that in which they achieved using their GA with the JGO implemented. Their hybrid GA demonstrated that it is capable of outperforming other hybrid and evolutionary approaches with respect to solution quality for the 48 DLP test problems [143].

Azimi and Charmchi addressed the solution of the QAP/S formulated budget constrained DLP in a different manner than those discussed to this point. They proposed solving said problem using discrete event simulation. By first converting the nonlinear DLP to a linear pure integer problem (PIP), the PIP could then solved to optimality using Lingo 8.0 [113]. The solutions to this sub-problem then provided the empirical distributions for assigning a department to a location during each period. These distributions were then used to discover the best solution to the original problem formulation using discrete event simulation, which was facilitated by Enterprise Dynamics 8.1 [89]. Azimi and Charmchi concluded that the method of solution was a viable approach to solving the budget constrained DLP and does so without requiring any nonrealistic assumption regarding the problem be made [11].

Hosseinin-Nasab and Emami employ yet a different evolution based intelligent algorithm to solve the QAP/S formulated DLP. They proposed a hybrid particle swarm optimization (PSO) approach to solve said problem [88]. PSO is, like other evolution based algorithms, a population based stochastic optimization technique developed by Dr.

Eberhart and Dr. Kennedy in 1995, which solves COPs by emulating the social behavior of a birds in a flock [90]. Since PSO operates in a continuous space, Hosseinin-Nasab and Emami first formulated a way of uniquely mapping the discrete QAP space into a continuous one. Internal to the PSO, they also included SA to further improve the best solution found so far at the conclusion of each iteration of PSO. The implementation of PSO with SA applied to enhance its effectiveness proved to be successful as they were able to achieve the best solution to 37 of the 48 DLP test problems. Compared to the literature on the DLP up until 2006 they outperformed all for the most part, especially for larger problems [88].

Pourvaziri and Naderi also employed SA as a means of improving the best found solution at each iteration of an evolutionary algorithm. Instead of PSO as the evolutionary algorithm, they implemented GA to solve the QAP/S formulated DLP modeled after Mckendall et al.'s formulation. Pourvaziri and Naderi for the first time introduce to the DLP the concept of evolving multiple populations. They employ a tri-population scheme where one population is generated randomly, one from individuals of the best region, and the other from individuals from the non-promising region. The latter two populations are generated by first solving the problem converted to a nonlinear programming model and then solving with CONOPT [60]. The solutions gathered then form empirical distributions for assigning departments to locations in each period. The best region population is then generated directly from these probability distributions and the non-promising from the inverse distributions. These populations are then evolved independently by the GA for a duration of generations referred to as the isolation time before being combined to form a main population that is then further evolved by the GA.

At each iteration of GA, SA is applied to the fittest individual in the population so as to more exhaustively search the local region for further improvement. Testing of their hybrid multi-population GA with SA enhancement demonstrated the sheer robustness that it poses in solving a wide range of period sizes and this robustness compared to other known methods in the literature becomes greater as the problem size increases. These other methods include that of Baykasoglu et al.'s ACO, Baykasoglu and Gindy's SA, McKendall et al.'s SA, Balakrishnan et al.'s (2003) hybrid GA, as well as some others employing Tabu and DP to solve the DLP [140].

Until now, the literature observed has been that in application to the QAP formulation of the DLP. The majority of research on the topic is on such a formulation of the problem due in large part to the already difficult nature of the problem as addressed before. Solving the MIP formulation of the problem only further increases the difficultly. This is why few have tackled such a formulated DLP. Although true, some have in fact tackled this challenging problem using hybrid and evolutionary approaches. A summary of a few notable works in this domain of the LP will now be provided.

Improving upon their work a couple years earlier, Dunker et al. (2005) hybridized their GA by incorporating DP to solve the Yang and Peters (1998) MILP formulated DLP. Employing DP in both a forward and backward direction, the fitness of the individuals in the population are evaluated. Their approach improved upon the results obtained by Yang and Peters and has the major advantage over the work of Montreuil and Laforge [126] and Lacksonen (1997) [105] in not restricting the layout to a skeleton structure [62].

Although not a DLP, in this same year Yang, Peters, and Tu proposed a hybrid approach for solving the detailed flexible SLP formulated as a MILP. They first solved the traditional QAP/S formulation of the problem using a combination of SA and spacefiling curve (SFC). Their SA employed a random interchange of two departments to facilitate perturbation of the layout. The resulting flow sequence and relative positions from the solution of this simpler problem were then used to solve the detailed flexible SLP formulated as a MILP using CPLEX [174]. Yang et al. in many ways solve the MILP SLP in a similar fashion to Liu and Meller, just with different methods of solution. Their implementation demonstrated effective solution to the detailed flexible SLP.

McKendall Jr. and Hakobyan, in addition to Dunker et al. (2005), also sought solution to the MILP formulation of the DLP with un-equal sized departments. To solve the problem they employed a boundary search technique accompanied by tabu search (TS) for further solution improvement. First a boundary search (construction) heuristic (BSH) places the departments on the boundaries of already placed ones with the placement determined according to the flow data. Departments with a higher cumulative flow get placement priority. Once an initial layout plan is obtained using BSH, TS is employed to improve the layout plan. McKendall and Hakobyan's BSH/TS approach performed well, especially for larger problems, relative to the literature for both solution to the DLP and SLP formulation of the problem, proving that such a method is a viable one [120].

APPENDIX B

MAPPING A PHYSICAL LAYOUT TO A SEQUENCE PAIR

Mapping a physical layout to a sequence-pair requires the employment of gridding rules, which dictate the generation of the object step-lines. In the literature [129,129], these are defined as module step-lines, however, to maintain consistency with the terminology used in this document these will be referred to as object step-lines from here forth. With that said, there are two object step-lines (one that is denoted as positive and another as negative) just as there are two sequences in the sequence-pair. The composition of the positive and negative object step-lines form the sequences, Γ_+ and Γ_-, form the sequence-pair. To generate these object step-lines the following rules are enforced: 1) boundaries of objects cannot be crossed by step-lines nor can the layout boundary itself be crossed and 2) no two positive or negative step-lines can cross one another, but they may both run parallel.

B.1 Generating an Object's Step-lines

The process of forming the object step-lines according to the aforementioned gridding rules is as follows. The positive object step-line of an arbitrary object is defined as the union of the horizontal bisecting line of the object of interest and its up-right (UR) and down-left (DL) step-lines. The UR step-line, beginning at the right end of the horizontal bisecting object line, moves in an alternating up and right direction, as the name implies, until reaching the top-left corner of the space all while adhering to the previously noted rules. Likewise, the DL step-line is formed by starting at the left end of the bisecting line and moving in an alternating down and left direction until reaching the bottom-left corner of the space. Joining these positive object step-lines forms the positive sequence. The negative sequence can likewise be formed through the composition of the negative object step-lines, which may be formed in a similar manner. The negative object step-lines are instead formed through the union of the vertical bisecting lines of each object and their corresponding left-up (LU) and right-down (RD) step-lines. Figure 77 provides an example of the outcome of this process for a notional layout composed of six objects.

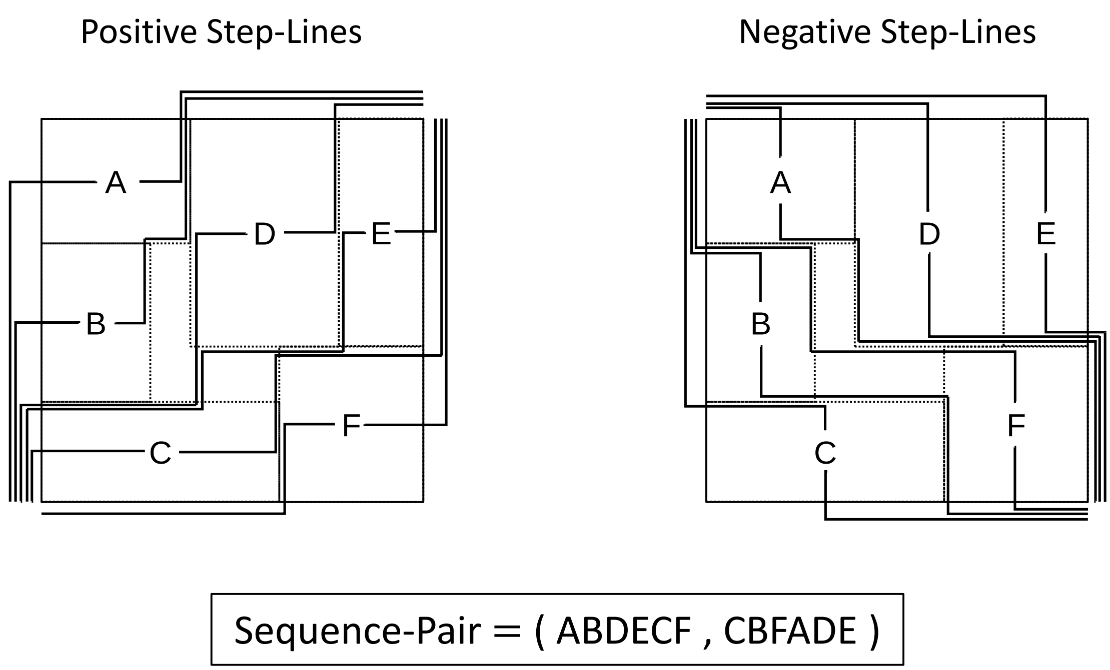


Figure 77 – Object step-lines of a physical layout and their correlation to the sequence-pair of the layout

B.2 Implication of the Gridding Rules

Until now a significant implication of the gridding rules has been overlooked. The aforementioned rules have the effect of producing positive and negative object step-lines that form linearly-ordered pairs of object sequences (Γ_+ and Γ_-), alternatively called a sequence-pair. The positive sequence (Γ_+) is ordered according to the positive object step-lines and starting from the upper-left whereas the negative sequence (Γ_-) is ordered according to the negative object step-lines and starting from the lower-left. This ordering and composition of the object step-lines to form the sequences is demonstrated in Figure 77 where the object step-lines are drawn out and the corresponding ordered sequences are provided. As demonstrated in the positive sequence diagram, object a's step-line is the upper, left most step-line relative to the rest. As such, it becomes the first object in the positive sequence. In the negative sequence, it is fourth from the lower left and as such falls in the fourth position of the negative sequence.

A couple noteworthy outcomes of the aforementioned implication and general observations shall now be discussed as they become important to the derivation of the FSPPM. First, sequence-pairs are structured such that the objects located at the corners of the physical layout have a higher probability of appearing at the ends of the appropriate sequence. Objects at the top-left and bottom-right corners of the space will always appear near the beginning and end of the positive sequence respectively. Similarly, objects at the bottom-left and top-right will always appear near the beginning and end of the negative sequences respectively. In both these scenarios, if an object is located at an absolute corner of the physical layout space, it won't just *likely* appear near the end/beginning of the appropriate sequence, but rather is will *definitely* appear at the absolute end/beginning of the sequence. These outcomes are a direct result of the gridding rules and thereby the linearly-order nature of each sequence.

The second outcome is such, the further the object is from the bisecting diagonals (i.e. line connecting complete opposite corners of the physical space, two of these diagonals), the more likely it will appear at the beginning/end of the corresponding sequence in the sequence-pair. This mirrors the previous observed outcome, but in a more generalized sense. First, let's consider the bisecting diagonal that traverses from the top-left to the bottom-right corner of the space. The further an object's centroid is from this diagonal (normal distance) the closer it will appear to the beginning/end of the negative sequence. If the object falls on this diagonal in the space or near it, then it is likely to appear in or near the middle of the negative sequence. The same is true for the opposite bisecting-diagonal and the positive sequence. Let's also return to the observation that objects at the absolute corners of the physical space will appear, and without uncertainty, at the absolute ends of the sequences. Relating this to the second outcome stated, this situation can be conceptually thought of being the result of the object's placement probability becoming hundred percent for its placement at the absolute beginning/end of the sequence. Being at the absolute corner of the physical space, the object is in theory the furthest it can be from one of the bisecting diagonals. Not only is it the furthest possible, but it is also the furthest amongst all other objects. As such, it has the highest probability of appearing at either the end or beginning of the appropriate sequence (all depending on which bisecting-diagonal is being considered).

Previous experience with the SP formulation led to several other key observations. These observations, in addition to the aforementioned outcomes, were instrumental to the construction of the FSPPM and thus highlighted here. The first notable observation is in regard to the nature of an object's appearance in the sequences. This observation was such; there is no discernable bias as to where the object appears relative to the most frequent position it appears in the sequence (mean position). Put alternatively, the appearance of the object in the sequence behaves normally about its expected position.

Another observation made was the following, identifying feasible sequence-pairs is often more difficult when solving highly constrained problems, such as those considered in this research. This difficulty was observed to be dependent upon three dominating factors. The first factor was the amount of white space available in the layout. The less white space there was, or in other words the more restricting the boundary constraints were, the more difficult it was to identify feasible sequence-pairs. This is quite intuitive. With less white space available there is less placement and orientation freedom and therefore fewer combinations of orientation and sequence-pairs that will yield a feasible design (i.e. one that when stacked remains within the bounds of the space). The more white space, the more flexibility there is to place and orient the objects in the layout without violating the boundary constraints. The second factor was the presence of objects in the layout with fixed placements. Like that of the white space, the presence of objects with fixed placements greatly restricts the number of orientation and sequence-pair combinations that result in feasible layouts (i.e. those that do not violate the fixed placement constraints and thus have the constrained objects placed appropriately in the space). Furthermore, the difficulty of discovering such feasible layouts (i.e. combinations of orientation and sequence-pairs), is directly proportional to the number of these constrained objects. The more that were present, the fewer feasible combinations there were and thus the more difficult it was for traditional placement procedures to discover them. In parallel to this, the third and final factor had to do with the total number of objects in the space. With more objects to place in the space, the greater number of potential combinations of orientations and sequence-pairs there were. This meant having to search through more combinations before finding those that yielded feasible designs.

APPENDIX C

PROBLEM INITIALIZATION INPUT CONDITION DATA

C.1 Input Station, Region, and Personnel Data

C.1.1 Station Data

The designer is required to establish the following properties provided in Table 44 for each of the potential stations. The table provides the property and its units or, where relevant, an example string.

Table 44 – Station input data

Station ID	“Station 1”
Type	“WORKSTATION”
Width (x)	feet
Height (y)	feet
Depth (z)	feet
Maintenance Spacing	feet
I/O X-offset	feet
I/O Y-offset	feet

Table 44 (continued)

Installation Time	hours
Uninstallation Time	hours
Move Rate	feet / hour

The width and height depict the non-rotated dimensions of the stations in the x and y-dimensions respectively. The depth is provided only if in the future it is desired for the layouts to be visualized in 3D. The maintenance spacing characterizes the closest distance another object can be placed to it. The I/O offset dimensions are automatically calculated based on the height and width. By default, the x-offset is zero while the y-offset is half the height. Installation and uninstallation times are in hours and are required to compute the rearrangement times in the performance model. Likewise, the move rate is to be provided in feet/hour. Definition of this property, like the installation times, contributes to the definition of the rearrangement time in the performance model.

C.1.2 Region Data

The designer is also required to establish the following properties provided in Table 45 for each of the regions. The table provides the property and its units or, where relevant, an example string.

Table 45 – Region input data

Region ID	“Region A”
Width (x)	feet
Height (y)	feet
Depth (z)	feet
Maintenance Spacing	feet

Width and height are defined identically to that of the station data whereby they are the non-rotated dimensions. The maintenance spacing is also the same as before when defining the station data.

C.1.3 Personnel Data

The designer is also required to establish the following personnel properties provided in Table 46. The table provides the property and its units, or where relevant, an example string for each condition.

Table 46 – Personnel data

Personnel ID	“John Doe”
Pay Rate	$ / hour
Unit	“Receiving Station”

The personnel data includes: the worker title, their nominal pay rate, and the station type they are associated with. The latter aligns with the station type provided in the station data table outlined above.

C.2 Problem Initialization Input Conditions

C.2.1 Horizon-Based Discrete Conditions

Each of the condition inputs presented in Table 47 are defined at each forecasting point of the horizon. In other words, the condition is defined at each of the dots shown in Figure 78 for a notional example problem. This example is the same one leveraged throughout the implementation chapter of this dissertation to reinforce key concepts.

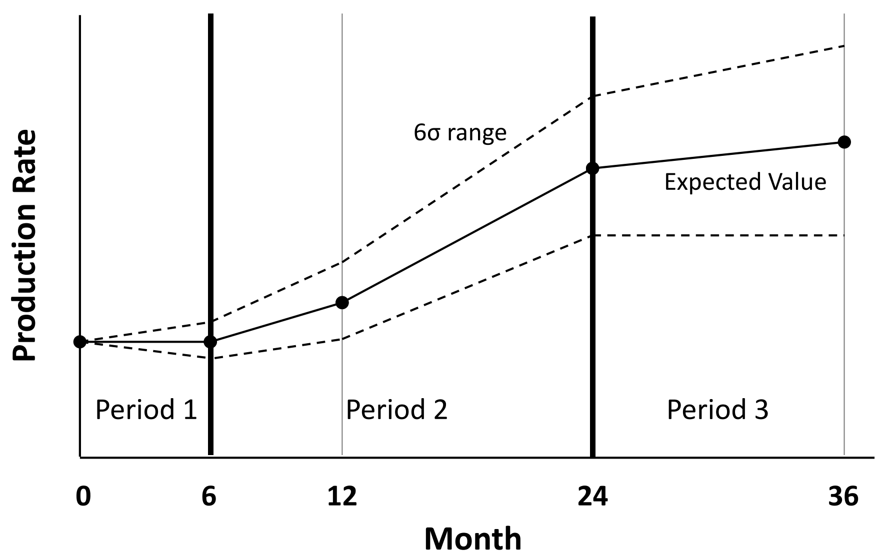


Figure 78 – Example of a horizon-based condition definition

Table 47 – Horizon-based discrete condition inputs

Labor Cost Adjustment Factor	[ ]
Work Days	days / week
Work Hours	hours / day

C.2.1.1 Labor Cost Adjustment Factor

The labor cost adjustment factor in the implemented model represents an input that enables the labor rates of the workers to be adjusted as time passes. This factor, when assigned a value of one, indicates no adjustment while a value greater than one, a positive inflation of the pay rates. This adjustment factor was implemented to enable the designer to have control over the pay rate with time and therefore be able to simulate worker pay raises. Simulating this can then enable the designer to evaluate how raising pay rates may impact their bottom-line.

C.2.1.2 Work Days and Hours

The work days and work hours provide the designer with the capability of adjusting how many days a week the average worker works and the average number of hours per day worked. If a decision to change the work days from five to seven along with an increase in work hours from eight to ten is to be evaluated, these inputs enable such a decision to be simulated.

C.2.2 Horizon-Product-Based Linear Conditions

Each of the condition inputs presented in Table 48 are defined at each forecasting point of the horizon and for each product. Therefore, it there are four unique products present in a scenario problem then there would be four condition forecasts established, where each forecast is composed of however many forecasting points are to be defined.

Table 48 – Horizon-product-based linear condition inputs

Desired Expected Production Rate	units / day
Production Rate Coefficient of Variance	%
Setup Rate	units / setup
Market Value	$ / unit
Estimated Manufacturing Cost	$ / unit
Direct Consumable Cost	$ / unit

C.2.2.1 Desired Production Rate

The desired production rate is implemented in this methodology to enable the designer to adjust the production rates of each of the products. With direct control over these rates, the designer can analyse a variety of different production scenarios. Production mix, production quantity, and production inclusion/exclusion can be assessed by the designer in the scenario set as a result of this input.

C.2.2.2 Standard Deviation of the Production Rate

The standard deviation input provides, for each product and at each forecasting point, the standard deviation of the production rate. This input enables the uncertainty about the expected desired production rate to be modelled, but also control across the entire horizon on a product-basis. The designer can control how the uncertainty grows with time and vary the magnitude across the different products. This input is directly leveraged by the performance model and local robustness method.

C.2.2.3 Setup Rate

The setup rate input captures costs and production time associated with setups at the stations on a product-basis. Not all products require a setup as frequently as others and some not at all, which is why this setup rate is defined on a product-basis. If no setups are required, the setup rate can be set to zero which forces the setup costs to be neglected in the performance model.

C.2.2.4 Market Value

The market value input provides the primary means, along with the production rate, of establishing the revenue in the performance model. The market value is the value of each product in the market, i.e. what it sells for. Adjustment of these enables a designer to consider scenarios where a product may become more valuable or also less value and how this occurrence may impact the bottom-line.

C.2.2.5 Estimated Manufacturing Cost

The estimated manufacturing cost of a single unit of a product is just that. It only provides an estimate from which the relative profit margin ratios can be established.

These are potentially leveraged to dynamically adjust the prescribed production rates if needed during the process analysis portion of the performance model. These manufacturing costs will effectively be updated once the performance model has been executed at which point, they are rendered mute.

C.2.2.6 Direct Consumable Cost

The direct consumable cost per unit of each product is a provided estimate of the consumables used during the process of transforming a product from raw inputs to its finished state. This input accounts for consumables such as electricity and materials.

C.2.3 Horizon-Station-Based Discrete Conditions

Each of the condition inputs presented in Table 49 are defined at each forecasting point of the horizon and for each station.

Table 49 – Horizon-station-based discrete condition inputs

Installation Cost	$
Displacement Cost	$ / unit distance
Support Conduit Displacement Cost	$ / unit distance

C.2.3.1 Installation Cost

The installation cost input defines, for each station, the fixed cost of installing a station. This can account for the cost to level a machine, bolt it down, etc. or for the cost to setup a workstation after it has been broken down and moved. Any station that experiences a change in position after a rearrangement will incur this cost.

C.2.3.2 Displacement Cost

The displacement cost provides the cost to displace a station over a specified distance, as would be known after rearrangement occurs. This can account for the cost of hiring a forklift or large machinery moving company to move a machine from one spot in the layout to another.

C.2.3.3 Support Conduit Displacement Cost

The support conduit displacement cost is one that is often overlooked. This cost input accounts for the costs of rerouting supporting conduits that are required for a moved station to operate in its new location. These can include electrical lines, water lines, HVAC, network lines, etc. This input is defined on a distance-basis to account for the cost per foot of pipe or cable, both of which are easy value to establish from market research.

C.2.4 Process-Based Conditions

Each of the condition inputs presented in Table 50 are defined for each station or each segment of each product-process flow. These inputs become instrumental in analysing the system's capacity in the performance model.

Table 50 – Process-based condition inputs

Station Capacity	units / hour
Setup Capacity	setups / hour
Handler Flow-Rate Capacity	units dist / hour
Handler Labor Rate	$ / hour
Number of Handlers	
Other Handling Costs	$ / dist unit

C.2.4.1 Station Capacity

The station capacity input establishes the number of units of the product that can be produced in an hour at each of the stations that it visits. Each station encompasses a different activity which adds value to the product and transforms it towards being a finished good. As such, each activity is likely to have different processing times and therefore capacities. This definition of the station capacities for producing the product enables the different stages of the process (i.e. activities at the stations) to be accurately modelled.

C.2.4.2 Setup Capacity

The setup capacity establishes the rate at which the activities at each station can be setup. Inverting this property enables the setup time to be defined. More often it is this property that the designer will easily be able to estimate. As such, inverting it enables the designer to define this input.

C.2.4.3 Handler Flow-Rate Capacity

The handler capacity is much like that of the station capacity, however, instead of being applied on a station-basis it is applied on a between-station segment-basis. The handler flow rate capacity establishes the number of units that can be moved one-unit distance per hour. Dividing this by the distance the handler must travel for each segment yields the conventional capacity for each segment, in other words, the number of products that can be transferred in an hour. This input becomes relevant when the material handler utilizations of the system are evaluated.

C.2.4.4 Handler Labor Rate

The handler labor rate defines the average labor rate for a segment's handler.

C.2.4.5 Number of Handlers

This input defines, for each segment, the number of handlers that are moving products. Multiplying this by the handler labor rate above establishes the total handling cost for the segment.

C.2.4.6 Other Handling Costs

The other handling costs input accounts for additional costs of handling that do not relate to the labor costs. This could be leveraged to account for the cost of operating a fork lift. This other handling cost can also be leveraged by the designer to simulate a traditional

MHC objective function. To do this, the designer would need to default many of the other inputs, such as the labor rates and market values, to zero.

C.2.5 Optimization Parameters

The optimization parameters available in this dissertation's implementation are provides below in Table 51, Table 52, and Table 53.

Table 51 – Optimization parameters of Stage One

Stage One	Population Size
	Elite Percentage
	Jumping Gene Probability
	Crossover Probability
	Mutation Probability
	Percent Feasible of Initial Population
	Maximum Population Initialization Time
	Number of Generations
	Time Limit
	Maximum Stall Generations

Table 52 – FSA parameters of Stage One

FSA	Maximum Number of Iterations
	Sample Size
	Probability of Uphill Move Acceptance

Table 52 (continued)

	Probability of Reassigning Fixed Object
	Probability of Swapping Adjacent Objects
	Probability of Rotating Unconstrained Objects
	c Coefficient (higher = more greedy search)
	k Coefficient (higher for larger problems)
	McKendall Method Option (on/off)

Table 53 – Optimization parameters of Stage Two

Stage Two	Population 1 Size
	Population 2 Size
	Population 3 Size
	Population Merged Size
	Elite Percentage
	Jumping Gene Coefficient - Population 1
	Jumping Gene Coefficient - Population 2
	Jumping Gene Coefficient - Population 3
	Jumping Gene Coefficient - Population Merged
	Jumping Gene Probability - Population 1
	Jumping Gene Probability - Population 2
	Jumping Gene Probability - Population 3
	Jumping Gene Probability - Population Merged

Table 53 (continued)

	Crossover Probability - Population 1
	Crossover Probability - Population 2
	Crossover Probability - Population 3
	Crossover Probability - Population Merged
	Mutation Probability - Population 1
	Mutation Probability - Population 2
	Mutation Probability - Population 3
	Mutation Probability - Population Merged
	Mutation Adjustment Coefficient - Population 1
	Mutation Adjustment Coefficient - Population 2
	Mutation Adjustment Coefficient - Population 3
	Mutation Adjustment Coefficient - Population Merged
	Number of Isolation Generations
	Number of Merged Generations
	Maximum Isolation Stall Generations
	Maximum Merged Stall Generations
	Isolation Time Limit
	Merged Time Limit
	Migration Rate

APPENDIX D

DESIGN VARIABLE DERIVED PROPERTIES

Several station properties are dependent on the definition of the design variables. For example, the orientations of the stations are established using the combination of the binary orientation design variables o_{s1} and o_{s2}. Four unique combinations are represented which map to the four discrete orientations of each station (0, 90, 180, 270°), as demonstrated in Figure 80. The physical orientation impacts the placement of the input/output point positions of the station and moreover, the length (x-coordinate direction) and depth (y-coordinate direction) of the station, where the length and depth properties are that of the rotated dimensions.

Mapping the binary variable combinations to the I/O points is performed for every s station (i.e., $\forall\, s = 1, \dots, N_s$) as follows, where $\Delta x_{s,oi}$ and $\Delta y_{s,oi}$ are the I/O points relative position with respect to its geometrical center as shown in Figure 79:

$$s_s = (ds)ws + (1 - ds)ms_s \tag{68}$$

$$\begin{aligned} x_{s,oi} = x_s + o_{s1}\big[\big(\Delta y_{s,oi} + s_s\big)(1 - o_{s2}) - \big(\Delta y_{s,oi} + s_s\big)o_{s2}\big] \\ + (1 - o_{s1})\big[\big(\Delta x_{s,oi}\big)(1 - o_{s2}) - \big(\Delta x_{s,oi}\big)o_{s2}\big] \end{aligned} \tag{69}$$

$$\begin{aligned} y_{s,oi} = y_s + o_{s1}\big[\big(\Delta x_{s,oi}\big)(1 - o_{s2}) - \big(\Delta x_{s,oi}\big)o_{s2}\big] \\ + (1 - o_{s1})\big[-\big(\Delta y_{s,oi} + s_s\big)(1 - o_{s2}) + \big(\Delta y_{s,oi} + s_s\big)o_{s2}\big] \end{aligned} \tag{70}$$

where *ds* is a binary switch defined by the user to change between using the double spacing interaction formulation and a single spacing formulation and s_s is the spacing from the physical boundary to the I/O point. The horizontal and vertical lengths, l_o and d_o respectively, of the objects, based on their binary orientation variables and its x- and y-direction dimensions when not rotated (i.e., α_o and β_o), are found to be:

$$l_o = \alpha_o(1 - o_{o1}) + \beta_o o_{o1} \quad \forall\, o = 1, \dots, N \tag{71}$$

$$d_o = \alpha_o + \beta_o - l_o \quad \forall\, o = 1, \dots, N \tag{72}$$

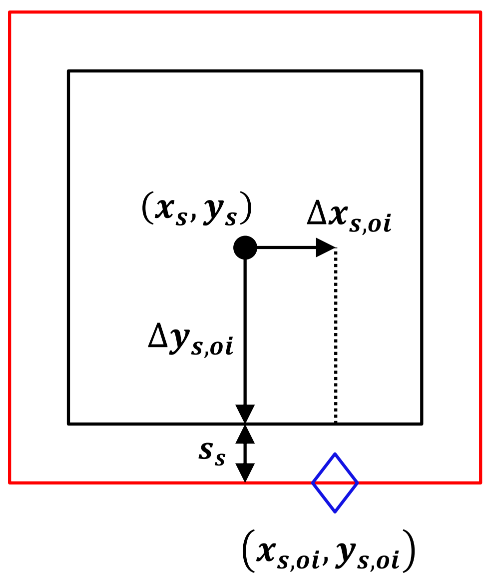


Figure 79 – Geometrical center coordinates and I/O point offsets (positive convention shown) of station *s*

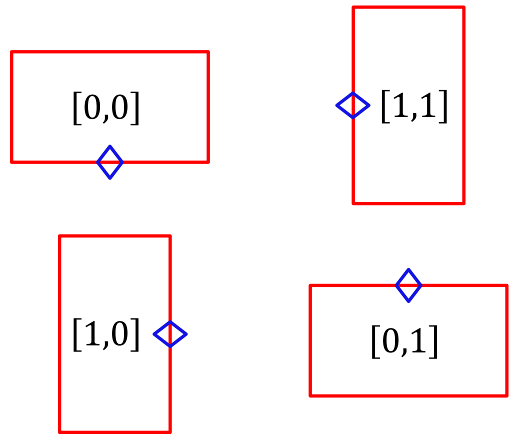


Figure 80 – Binary orientation variables [o_{o1}, o_{o2}] versus physical orientations

APPENDIX E

CONSTRAINT CONSIDERATIONS

There are five hard constraint groups considered in this dissertation. The first three presented: avoidance of overlapping objects in the space, I/O point accessibility, and the avoidance of objects creating closed loops are always hard in nature and thus always take the forms presented here. The other two constraints presented: object confinement to the building OML boundaries and budgetary restriction on the evolution of the layout design can be soft in nature. Therefore, only under certain circumstances will the constraint forms presented here be applicable for these. These circumstances, and the resulting form of these constraints when soft, are encapsulated in the implementation section of this document. Such circumstances and forms of these two constraints are outlined in Section 4.4.2.1. The five hard constraint forms of the constraint groups are presented as they were noted above starting with the overlap avoidance constraint group.

E.1 Overlap Avoidance Constraints

The object overlap avoidance constraint group addresses the notion of two objects overlapping one another. This can include two stations or a station and an infeasible region overlapping one another. The avoidance of overlap can be identified as the activation of any one of the below inequalities. In this constraint group, the maintenance spacings are applied given that they are the absolute physical constraint margin between the physical boundaries of the objects. These inequalities are performed for every s, o such that $o > 1$ (i.e., $\forall\ s = 1, \dots, N_s, o = s + 1, \dots, N$).

$$x_s - x_o \geq \frac{l_s + l_o}{2} + ms_s + ms_o \tag{73}$$

$$x_s - x_o \geq \frac{l_s + l_o}{2} + ms_s + ms_o \tag{74}$$

$$y_s - y_o \geq \frac{d_s + d_o}{2} + ms_s + ms_o \tag{75}$$

$$y_s - y_o \geq \frac{d_s + d_o}{2} + ms_s + ms_o \tag{76}$$

These inequalities form the non-overlapping disjunctive conditions where, if any single one is active, overlap is avoided for the object combination s ,o. When all combinations contain at least one active condition, no objects in the space are overlapping and the layout design is deemed feasible per this constraint group.

E.2 I/O Point Accessibility Constraints

To ensure that the I/O points remain accessible (i.e., outside of all stations and infeasible regions and within all boundaries) additional constraints must be included. The first series of constraints ensure that the I/O points fall within the outer boundaries. The four constraints in Eq. (77) provide this assurance. It may be observed that there is no factor of two on the walking spacing term in the below constraints. This is because the walking spacing associated with station s is built into the I/O point coordinates, $x_{s,oi}$ and $y_{s,oi}$, as was established in Eq. (69) and Eq. (70).

$$
\begin{aligned}
x_{s,oi} &\geq s_b \\
y_{s,oi} &\geq s_b \\
x_{s,oi} + s_b &\leq x_{max} \\
y_{s,oi} + s_b &\leq y_{max}
\end{aligned}
\qquad \forall\ s = 1, \dots, N_s \tag{77}
$$

$$
s_b = (ds)ws + (1 - ds)ms_b \tag{78}
$$

where *ds* is the binary switch, which switches between a double and a single spacing interaction formulation, *ws* is the walking spacing, and ms_b the boundary maintenance spacing.

I/O point accessibility is also dependent on the points falling outside of all other stations and infeasible regions. In other words, objects, as is not the case presented in Figure 81. In other words, overlap avoidance constraints similar to those established in Equations (73) – (76) must be implemented. The four disjunctive inequality conditions below ensure that the I/O points fall outside all other stations and infeasible regions. These inequalities are evaluated for every *s, o* (i.e., $\forall\ s = 1, \dots, N_s, o = 1, \dots, N$).

$$
x_{s,oi} - x_o \geq \frac{l_o}{2} + ws \tag{79}
$$

$$
x_o - x_{s,oi} \geq \frac{l_o}{2} + ws \tag{80}
$$

$$
y_{s,oi} - y_o \geq \frac{d_o}{2} + ws \tag{81}
$$

$$y_o - y_{s,oi} \geq \frac{d_o}{2} + ws \tag{82}$$

where if any one of these four are active, the I/O points remain accessible for the s, o object combination. When all combinations of objects have at least one active constraint in this set of disjunctive conditions as well as abide by the boundary constraints presented before, the layout design is deemed feasible.

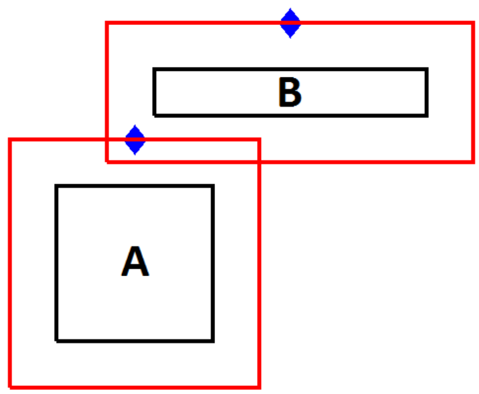


Figure 81 – I/O point overlapped by a station or infeasible region

Introduction of these additional constraints are a joint by-product of the multi-spacing and advanced flow distance formulations considered. These constraints also provide the added benefit of improving the performance of the formulation as excessive computations involving the path generation procedure may be avoided. For example, if an I/O point is inaccessible the path generation algorithm would proceed until the maximum limit number of branches is met as no feasible path could ever be achieved. The inclusion of these constraints in turn enables these unproductive computations to be avoided upfront. These act as an indicator, which triggers the use of the traditional rectilinear approach with a scaling factor to provide the objective function with a path distance value that is also reflective of the circumstances.

E.3 Closed Loop Avoidance – Preventing Inaccessible Regions

Using the I/O point accessibility assurance constraints, discussed above, as a way of identifying if an arrangement provides path feasibility is not adequate, however. Situations, such as that demonstrated in Figure 82, can occur where inaccessible regions are formed by a series of adjoined objects (i.e., stations or infeasible regions) and where these accessibility constraints remain unviolated.

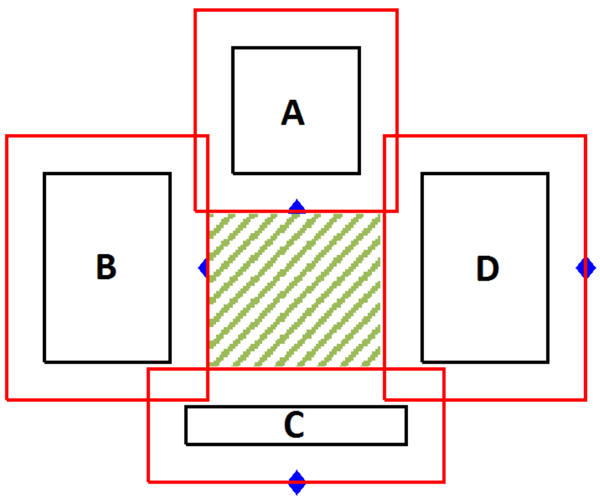


Figure 82 – Artificially created inaccessible regions

Objects can become adjoined and still avoid violating the overlap avoidance constraints defined in Equations (73) – (76) when only their walking boundaries (i.e., physical boundaries adjusted to encompass the walking spacing), and not their maintenance boundaries, overlap. In this situation, to get to the other side of the adjoined stations, one must go around the end, as cutting between the two stations is no longer feasible, which is denoted by the adjoined property of the two stations. If enough elements become adjoined, closed loops can form, as demonstrated in Figure 82 with station's A-D. These closed loops in turn can create *fenced* regions. This is emphasized by the rectangular green hashed area in Figure 82. If an I/O point falls within one of these regions, like that of station A or B, accessing it from outside becomes impossible. Furthermore, these situations do not violate the I/O point accessibility constraints defined in Equations (79) – (82) as they are not overlapped by an infeasible region or another station. Since the advanced flow distance method implemented in this dissertation is built on the premise that a feasible path is present, there is no point in executing such a computationally expensive procedure when no such path is achievable.

Determining if closed loops are present among a group of adjoined objects is not a trivial task. Doing so requires the use of advanced techniques such as graph theory. The problem of identifying the presence of closed loops is classified as the Hamiltonian cycle problem and in this application would be that for an undirected graph. This problem itself is NP-complete so solving it requires an exhaustive method [78]. Solving such a problem for every layout of a layout design, whose solution is a problem that is itself NP-hard, becomes computationally unmanageable and unrealistic.

To avoid this computational dilemma a less expensive approach was implemented that is based on a series of logical criteria. Note that outer boundaries are computationally modeled as infeasible regions. These boundary regions are only active if directly in contact with either a station or infeasible region. The criterion is based on a binary connection matrix for the adjoined objects, which establishes each object's direct connections (i.e., objects in which their walking boundaries overlap). Each object is identified in the matrix as being adjoined to itself for easier identification of some of the below criteria when implemented in a coding context. The following criteria identify adjoined configurations that do not produce inaccessible regions:

1) Groups of less than or equal to three objects

2) Groups of four objects where two are ends

3) Groups of four objects where all four are directly linked

4) Groups of *n* objects where at least *n-3* are ends (*n-3* rule)

5) Groups of *n* objects where *n-r* are directly connected to the same objects and *r* objects are ends

where *ends* are easily identified as those objects with only a single connection (i.e., a single non-zero row object in the matrix). Connections to the same object are identified by evaluating the columns of the matrix. Figure 83 provides a few example object configurations that fall under these criteria. It is understood that these criteria alone may not encompass all configurations that are acceptable, however, it should be noted that the first four criteria do completely cover all acceptable six object configurations.

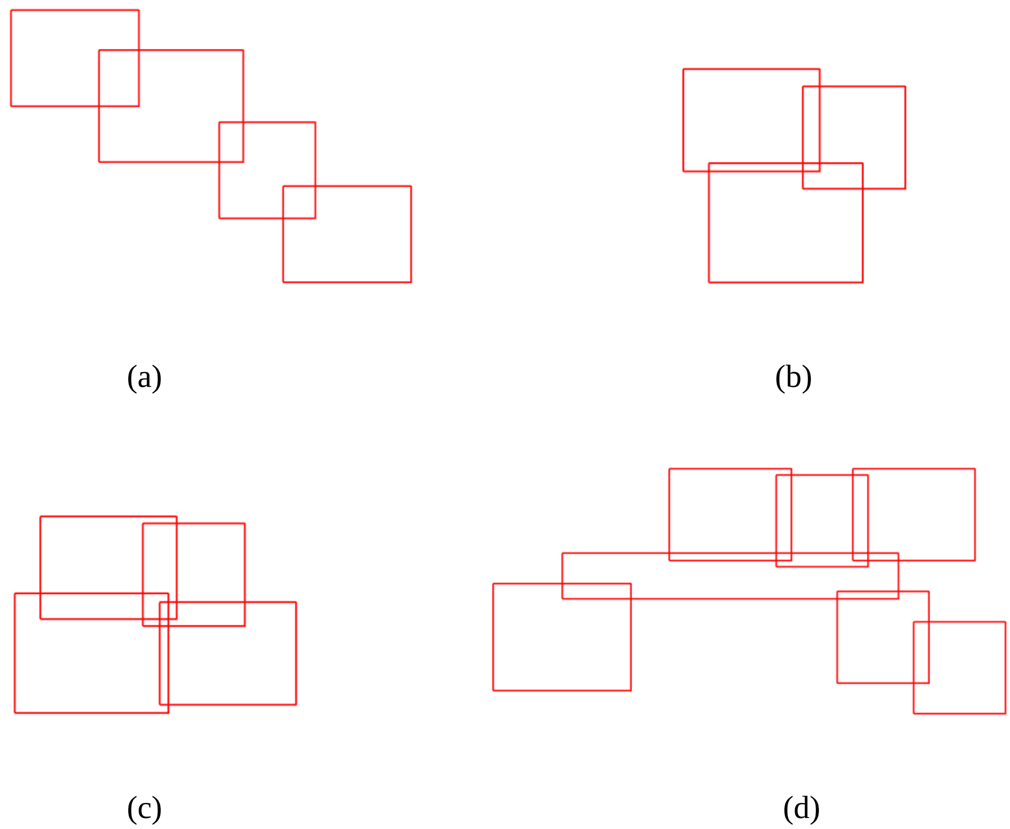


Figure 83 – Examples of acceptable configurations (a) criterion two active (b) criterion one active (c) criterion three active (d) criterion five active

In practice, grouping too many independent objects together may not be practical. Doing so could reduce the safety profile of the layout as emergency exit paths could become excessively long. Additionally, several objects grouped together could produce high temperatures and poor air flow, which would degrade the performance of both the machines and the workers. As a result, limiting the number of objects that can be grouped together can indirectly account for factors not being explicitly captured in this formulation. Therefore, the inclusion of a limiting group size is implemented such as to enable additional trades to be performed and user control to be provided.

Implementing these criteria and using them in combination with the I/O point accessibility constraints, Equations (73) – (76) and Equations (79) – (82), enables configurations for which path feasibility can be identified. If any of these constraints are violated or none of the above criteria apply, then the configuration is labeled as infeasible. This labeling acts as an indicator that, as mentioned earlier, triggers the use of the more traditional rectilinear approach to determine the path distances. A scaling factor is applied to said path distances to enable these infeasible cases to be identified. The inclusion of this approach to the formulation greatly improves its performance by ensuring that executions of the more computational expensive advanced flow distance method are avoided for cases where a feasible path will never be achievable.

The inclusion of the soft constraints, and more specifically the potential to allow violation of the boundaries under certain circumstances, required slight modification to this process of established the presence of closed loops. When violation of the boundaries is allowed, the boundary regions are set inactive, regardless if in contact with either a station or infeasible region. This again is a unique consequence of the implemented penalty function approach to defining the soft boundary constraints.

E.4 Outer Layout Boundary Constraints

Bounding the coordinate centroid position of each object is addressed by considering each object's relative position to the outer rectangular boundary (0, 0) and (x_{max}, y_{max}), or outer mold line (OML) of the space. These constraints are the first of the soft constraints to be discussed. The discussion to follow establishes the hard form of the boundary constraints, or in other words, when the designer/management deems such constraints as being absolute, whereby violation of them (i.e. placement of objects outside the OML of the layout) is not allowed. Bounding the objects was achieved by implementing the set of inequalities provided below, where the first two address the left and bottom boundaries respectively, whereas the last two address the right and top boundaries respectively.

$$\begin{aligned} x_s &\geq \frac{l_s}{2} + ms_s + ms_b \\ y_s &\geq \frac{d_s}{2} + ms_s + ms_b \\ x_s + \frac{l_s}{2} + ms_s + ms_b &\leq x_{max} \\ y_s + \frac{d_s}{2} + ms_s + ms_b &\leq y_{max} \end{aligned} \qquad \forall\, s = 1, \ldots, N_s \tag{83}$$

Here the maintenance spacing's are applied as the constituent of the absolute margin between the outer boundary and the objects. In other words, the objects cannot fall any closer to the boundary than the summation of the two maintenance spacings. Though the coordinate centroid position variables are bounded by ranges in the Stage Two algorithm, these ranges are only approximate and therefore are not enough alone to ensure all objects fall within the OML of the layout. This is a result of these ranges being dependent on the object's current orientation. Implementation of these constraints therefore ensures feasibility with respect to all objects falling within the OML of the layout regardless of orientation. Further, in Stage One the first two constraint equations are inherently guaranteed by the fundamental nature of the sequence-pair model employed. Instead, only the latter two equations for the top and right boundaries need be met to establish feasibility of the design.

E.5 Budget Constraints

The budget constraints are the second soft constraint type. Establishing financial feasibility when these constraints are rendered hard relies on observing the debt remaining after the prescribed budget is applied. The debt is determined as follows:

$$DEBT_t = \begin{cases} RCC_t + \phi_{boundary,t} - p_b \cdot Net\ Income_{t-1}, & Net\ Income_{t-1} > 0 \\ RCC_t + \phi_{budget,t}, & Net\ Income_{t-1} \leq 0 \end{cases} \tag{84}$$

This is the same as Equation (63) provided when the budget penalty function was presented in the implementation section of this document. If the budget completely covers the costs associated with rearranging the layout and the boundary penalty cost, then the resulting debt would be negative. When the following condition applies, the restructuring is feasible. When true for each period then the layout design is deemed feasible.

APPENDIX F

REPRESENTATIVE PROBLEM TEST SET

In this appendix, the *52 Problem Test Set* leveraged throughout much of the experimentation is defined. This test set was formed strategically to exercise the developed solution techniques and methods across a range of different problem characteristics, some generic, others unique to the formulation of this dissertation.

Table 54 provides the high-level problem setup characteristics for the 52 problems composing the *52 Problem Test Set*. The description column indicates first the boundary characteristics, U for unconstrained and C for constrained, and then the nature of the objects in the layout, U for all objects being movable and C for the presence of constrained objects in the problem. To simulate an unconstrained problem, the boundaries were set sufficiently large. Xmax and Ymax indicate the boundary unit distance lengths. Nm and Nf indicate the number of movable and fixed, i.e. constrained, objects in the problem respectively. The definition of the sizes of these objects was done randomly. They were assigned to have dimensional integer lengths between 2 and 7 unit lengths. The placement of the constrained objects was largely random, though some strategic placements were manually established to force the designs, once tested in Stage Two, to ideally break away from a tightly packed configuration. This was achieved by placing fixed objects at opposite ends or corners of the space and then defining them in the process flows as the start/end points of the processes. Np indicates the number of processes considered in the problem. Finally, WS and AR represent the problem's white space and aspect ratio. As one can observe, a variety of problem sizes varying both in number of objects and number of periods are considered in this test set. Additionally, varying white spaces and aspect ratios are considered to enable the performance of the solution techniques to be evaluated over such varying conditions. Lastly, different numbers of constrained objects are considered in this test set, which is a notably unique consideration of this test set. The definition of this test set was strategic. It enabled the developed FSPPM and the solution procedures to be tested over a range of varying characteristics.

Table 54 – 52 Problem Test Set

Problem	Description	Objects	Periods	Xmax	Ymax	Nm	Nf	Np	WS	AR
1	U + U	6	1	100	100	6	0	3	98.25	1
2	C + U	6	1	17.53	26.3	6	0	3	75.05	1.5
3	C + U	6	1	14.6	21.9	6	0	3	64.06	1.5
4	C + U	6	1	19.6	19.6	6	0	3	75.05	1
5	C + U	6	1	17.88	17.88	6	0	3	64.06	1
6	C + C	6	1	17.53	26.3	5	1	3	75.05	1.5
7	C + C	6	1	17.53	26.3	4	2	3	75.05	1.5
8	C + C	6	1	19.6	19.6	5	1	3	75.05	1
9	C + C	6	1	19.6	19.6	4	2	3	75.05	1
10	C + C	6	1	14.6	21.9	5	1	3	64.06	1.5
11	C + C	6	1	14.6	21.9	4	2	3	64.06	1.5
12	C + C	6	1	17.88	17.88	5	1	3	64.06	1
13	C + C	6	1	17.88	17.88	4	2	3	64.06	1
14	U + U	12	1	100	100	12	0	6	97.35	1
15	C + U	12	1	26.6	39.9	12	0	6	75.05	1.5
16	C + U	12	1	22.16	33.24	12	0	6	64.06	1.5
17	C + U	12	1	32.6	32.6	12	0	6	75.05	1
18	C + U	12	1	27.15	27.15	12	0	6	64.06	1
19	C + C	12	1	26.6	39.9	10	2	6	75.05	1.5
20	C + C	12	1	26.6	39.9	8	4	6	75.05	1.5
21	C + C	12	1	32.6	32.6	10	2	6	75.05	1
22	C + C	12	1	32.6	32.6	8	4	6	75.05	1
23	C + C	12	1	22.16	33.24	10	2	6	64.06	1.5
24	C + C	12	1	22.16	33.24	8	4	6	64.06	1.5
25	C + C	12	1	27.15	27.15	10	2	6	64.06	1
26	C + C	12	1	27.15	27.15	8	4	6	64.06	1
27	U + U	6	3	100	100	6	0	3	98.25	1

Table 54 (continued)

28	C + U	6	3	17.53	26.3	6	0	3	75.05	1.5
29	C + U	6	3	14.6	21.9	6	0	3	64.06	1.5
30	C + U	6	3	19.6	19.6	6	0	3	75.05	1
31	C + U	6	3	17.88	17.88	6	0	3	64.06	1
32	C + C	6	3	17.53	26.3	5	1	3	75.05	1.5
33	C + C	6	3	17.53	26.3	4	2	3	75.05	1.5
34	C + C	6	3	19.6	19.6	5	1	3	75.05	1
35	C + C	6	3	19.6	19.6	4	2	3	75.05	1
36	C + C	6	3	14.6	21.9	5	1	3	64.06	1.5
37	C + C	6	3	14.6	21.9	4	2	3	64.06	1.5
38	C + C	6	3	17.88	17.88	5	1	3	64.06	1
39	C + C	6	3	17.88	17.88	4	2	3	64.06	1
40	U + U	12	3	100	100	12	0	6	97.35	1
41	C + U	12	3	26.6	39.9	12	0	6	75.05	1.5
42	C + U	12	3	22.16	33.24	12	0	6	64.06	1.5
43	C + U	12	3	32.6	32.6	12	0	6	75.05	1
44	C + U	12	3	27.15	27.15	12	0	6	64.06	1
45	C + C	12	3	26.6	39.9	10	2	6	75.05	1.5
46	C + C	12	3	26.6	39.9	8	4	6	75.05	1.5
47	C + C	12	3	32.6	32.6	10	2	6	75.05	1
48	C + C	12	3	32.6	32.6	8	4	6	75.05	1
49	C + C	12	3	22.16	33.24	10	2	6	64.06	1.5
50	C + C	12	3	22.16	33.24	8	4	6	64.06	1.5
51	C + C	12	3	27.15	27.15	10	2	6	64.06	1
52	C + C	12	3	27.15	27.15	8	4	6	64.06	1

In addition to these problem characteristics several other assumptions were required so as to reduce the performance model to a more simplistic model that was more comparable to literature approaches that only consider material handling costs as their objective function. These assumptions are as follows:

1) Assumed that changes in assets from one period to the next did not need to be assessed, i.e. the same objects were to appear in all periods of the dynamic problems

2) With no additions of stations, capital costs are assumed to be zero across all periods

3) Assumed the labor cost adjustment factor was constant across the planning horizon and at a value of one

4) Assumed five work days in a week and eight-hour work days

5) Market values were assumed to be zero, effectively forcing the performance model to focus on costs

6) Direct consumable costs were assumed to be zero

7) Setup were assumed to be absent, i.e. set to zero

8) Worker labor rates (handlers and station workers) were set to zero and other handling costs to $1/unit distance travelled, which effectively reduces the performance model to just considering material handling costs like that of the literature

9) Assumed a production volume of 10 units/day in total and distributed evenly among the processes in the first evolution, then distributed randomly throughout the other evolutions (applicable for the dynamic problems)

10) Assumed no process changes (i.e. the same process flows were present across all periods and unit costs the same and as specified in assumption eight above)

11) Processes were generated mostly randomly, though start and end stations were sometimes strategically chosen as stated before

12) Processes varied in length, i.e. the number of objects visited

13) The capacity of the between station handlers was assumed to be sufficiently high such that the dynamic production rate would not alter the relative distributions of the production volumes across the processes, which would have the effect of skewing results and voiding the ability to make comparisons

14) Likewise, the capacities of stations were also assumed sufficiently high as to avoid the same occurrence as noted before

15) Number of handlers were set to one, though with their zero labor rates and high capacities they were rendered all but non-relevant; this still needed to be done such that handler utilization levels never exceeded 100% and thus the dynamic adjustment technique could be avoided

16) Costs of station uninstallation and installation were set to $0.10 (rotated, but unmoved)

17) Costs of station movement (i.e. rearrangement) were set to $1/unit distance

18) Assumed that the walking distances were zero, effectively matching literature approaches

19) Assumed the number of periods matched the number of horizon segments for simplicity

20) Assumed no regions, however fixed, i.e. constrained, stations create the same effect from a constraint perspective

21) Assumed a green facility design, whereby no initial rearrangement costs would be present in defining the first period design

22) Assumed the objective function would be that of direct retained earnings to focus on operational improvement from a material handling perspective provided the earlier assumptions implemented in defining the problems of the test set

23) Assumed boundary constraints were hard

24) Assumed that budget constraints were inactive to facilitate different designs being formed across the periods of the dynamic periods without restriction

Though many assumptions were made in in designing these test set problems, many of these were only made to simplify the model developed to a form more comparable to the literature. Moreover, since this *52 Problem Test Set* was primarily created to facilitate the testing of the developed methods and solution techniques, and not that of the performance model itself, this simplification was more than justifiable. The performance model is extensively tested in the final Experiment 6 of this dissertation.

APPENDIX G

EXPERIMENTAL RESULTS

G.1 Initial Optimization Observations

Initial testing of the solution algorithm revealed that using the retained earnings at the end of the horizon as the overall objective function produced poor optimization performance. It was believed that part of the reason for this was that the retained earnings, as it is defined in this formulation, encapsulated all direct and indirect costs. In the case where personnel are being underutilized, most of these costs become indirect (from idle labor, i.e. workers not contributing to value adding activities). Since indirect labor costs are purely a function of workable hours, it remains unchanged regardless of the layout design. As such, it skews the objective function in effect diminishing the impact that changes of the layout have on the overall objective function. The result is a suboptimal performing optimization algorithm. To correct this, a "direct" retained earnings figure, which neglected the production related indirect costs and indirect labor costs ($ILC_t + PRIC_t$ in Equation (36)) was implemented to avoid this problem and capture the impact more appropriately. The result, was a far better performing algorithm, generating improved result optimality.

G.2 Experiment I

Table 55 – Experiment 1.C. raw results

Problem	Case	Replication	GA Time	PopInt Time	Unique	Objective
6	1	1	94.612	20.152	1568	121000
6	1	2	101.080	19.857	1076	142000
6	1	3	120.360	18.864	1348	118000
6	1	4	125.550	22.095	1338	123000
6	1	5	130.070	21.198	1400	151000
6	2	1	88.073	3.109	1700	144000
6	2	2	105.960	2.932	1684	125000
6	2	3	111.300	3.080	916	136000
6	2	4	120.050	3.035	1145	136000
6	2	5	128.610	2.914	886	132000
6	3	1	88.374	3.053	1274	144000
6	3	2	106.410	3.059	1801	130000
6	3	3	114.570	2.808	1199	132000
6	3	4	120.670	2.933	1272	123000
6	3	5	126.480	3.183	1106	139000
6	4	1	88.157	2.972	1672	144000
6	4	2	105.880	2.600	1666	136000
6	4	3	114.960	3.289	1794	121000
6	4	4	120.150	2.935	1106	144000
6	4	5	131.610	3.216	1247	123000
6	5	1	90.074	3.210	1203	142000
6	5	2	504.620	3.009	1216	131000
6	5	3	113.540	3.138	1103	123000
6	5	4	119.430	3.181	857	141000
6	5	5	128.400	3.218	1028	139000
7	1	1	89.619	85.155	1144	155000
7	1	2	103.790	120.520	1152	146000
7	1	3	116.650	121.700	1245	146000
7	1	4	117.690	113.400	1223	146000
7	1	5	132.020	93.388	1100	146000
7	2	1	90.663	2.286	932	155000
7	2	2	103.320	2.221	1344	155000
7	2	3	111.390	2.331	836	155000
7	2	4	121.700	2.456	821	158000
7	2	5	133.060	2.250	1137	147000
7	3	1	89.374	2.374	1331	147000
7	3	2	103.000	2.283	1315	146000
7	3	3	113.280	2.479	903	155000

Table 55 (continued)

7	3	4	122.530	2.104	1111	155000
7	3	5	130.360	2.339	1170	147000
7	4	1	89.518	2.233	1130	155000
7	4	2	102.720	2.502	1118	147000
7	4	3	117.260	2.693	1080	155000
7	4	4	127.470	2.271	1155	155000
7	4	5	133.060	2.684	1559	137000
7	5	1	91.643	2.438	1444	154000
7	5	2	106.520	2.525	1377	146000
7	5	3	112.710	2.648	1022	155000
7	5	4	120.860	2.396	995	155000
7	5	5	128.770	2.886	1112	146000
8	1	1	93.198	32.732	1163	128000
8	1	2	110.720	32.909	1231	136000
8	1	3	121.770	32.441	1074	144000
8	1	4	125.360	32.006	772	149000
8	1	5	135.000	34.691	1036	143000
8	2	1	92.572	4.130	840	166000
8	2	2	105.570	4.535	931	148000
8	2	3	115.970	3.588	699	160000
8	2	4	123.490	3.971	936	141000
8	2	5	132.290	4.476	792	145000
8	3	1	93.529	4.708	958	136000
8	3	2	111.510	4.079	883	148000
8	3	3	121.360	4.107	1298	145000
8	3	4	120.980	4.075	738	160000
8	3	5	136.800	3.816	916	137000
8	4	1	95.138	4.049	1082	142000
8	4	2	109.570	4.623	858	155000
8	4	3	119.710	4.275	954	144000
8	4	4	125.790	3.922	1118	154000
8	4	5	134.580	3.789	1127	137000
8	5	1	94.814	4.353	976	143000
8	5	2	104.190	4.655	663	149000
8	5	3	114.710	4.576	720	161000
8	5	4	123.400	3.797	763	165000
8	5	5	136.020	4.792	1040	169000
9	1	1	98.175	181.830	649	157000
9	1	2	107.790	170.440	1018	159000
9	1	3	120.000	154.270	860	152000
9	1	4	123.670	154.180	817	143000
9	1	5	139.080	152.130	788	152000

Table 55 (continued)

9	2	1	92.650	2.751	892	159000
9	2	2	107.650	2.738	1059	164000
9	2	3	120.360	2.892	719	157000
9	2	4	127.730	2.701	835	168000
9	2	5	131.890	2.805	698	172000
9	3	1	93.825	2.986	721	156000
9	3	2	109.120	2.869	962	159000
9	3	3	118.720	2.949	974	159000
9	3	4	129.030	2.835	1020	168000
9	3	5	134.320	2.828	911	152000
9	4	1	94.879	3.074	831	159000
9	4	2	115.220	3.069	1088	157000
9	4	3	125.640	2.839	1076	159000
9	4	4	124.430	3.417	1042	159000
9	4	5	139.270	2.982	1149	159000
9	5	1	97.442	3.258	769	164000
9	5	2	108.080	3.321	710	160000
9	5	3	116.030	2.764	698	157000
9	5	4	122.200	3.141	799	164000
9	5	5	129.730	3.181	652	159000
10	1	1	94.332	7.490	2009	149000
10	1	2	106.190	7.584	1973	134000
10	1	3	117.060	7.294	2116	149000
10	1	4	127.310	7.259	1733	146000
10	1	5	130.290	7.110	3036	147000
10	2	1	95.701	1.562	2641	145000
10	2	2	104.330	1.545	1241	157000
10	2	3	116.410	1.506	1394	137000
10	2	4	123.880	1.631	2116	118000
10	2	5	126.880	1.573	2127	149000
10	3	1	92.336	1.494	1629	140000
10	3	2	106.370	1.626	2132	170000
10	3	3	113.030	1.581	1349	167000
10	3	4	121.640	1.618	1873	151000
10	3	5	127.840	1.587	1755	163000
10	4	1	95.065	1.463	2128	127000
10	4	2	105.110	1.464	1639	154000
10	4	3	113.720	1.534	2842	142000
10	4	4	124.620	1.562	2456	142000
10	4	5	132.270	1.686	1258	144000
10	5	1	93.157	1.605	1991	160000
10	5	2	104.440	1.591	1806	157000

Table 55 (continued)

10	5	3	114.910	1.566	1290	147000
10	5	4	124.250	1.591	2331	154000
10	5	5	130.800	1.616	2430	155000
11	1	1	95.850	32.054	1373	181000
11	1	2	108.690	39.422	1446	181000
11	1	3	117.830	32.339	1970	181000
11	1	4	125.110	38.964	1369	190000
11	1	5	130.850	32.127	2012	179000
11	2	1	96.193	1.460	2263	186000
11	2	2	107.050	1.451	1310	190000
11	2	3	113.740	1.452	1280	190000
11	2	4	123.640	1.477	1535	190000
11	2	5	129.750	1.494	1584	190000
11	3	1	94.176	1.534	1171	194000
11	3	2	105.040	1.463	1557	190000
11	3	3	116.370	1.451	1684	178000
11	3	4	129.080	1.480	2481	169000
11	3	5	128.900	1.517	1828	181000
11	4	1	94.932	1.457	1573	183000
11	4	2	108.110	1.527	1484	185000
11	4	3	114.370	1.403	1354	185000
11	4	4	124.690	1.471	1290	198000
11	4	5	131.310	1.401	2652	183000
11	5	1	95.834	1.470	1751	198000
11	5	2	109.030	1.480	1545	194000
11	5	3	117.290	1.441	1825	173000
11	5	4	125.930	1.501	1817	192000
11	5	5	130.510	1.472	1245	191000
12	1	1	98.038	6.876	2647	153000
12	1	2	107.390	7.246	1639	163000
12	1	3	116.100	6.555	2068	163000
12	1	4	127.750	7.148	1773	162000
12	1	5	131.050	7.199	1543	160000
12	2	1	98.940	1.470	2793	162000
12	2	2	111.090	1.487	2654	161000
12	2	3	116.540	1.433	2115	153000
12	2	4	126.830	1.483	2052	169000
12	2	5	133.160	1.504	2650	176000
12	3	1	95.308	1.523	1504	168000
12	3	2	107.820	1.475	1709	163000
12	3	3	119.690	1.516	2505	173000
12	3	4	127.230	1.542	2290	177000

Table 55 (continued)

12	3	5	130.310	1.418	2401	144000
12	4	1	96.765	1.491	1838	170000
12	4	2	110.270	1.448	2987	168000
12	4	3	117.920	1.544	2343	175000
12	4	4	126.950	1.451	1996	165000
12	4	5	134.500	1.464	2077	162000
12	5	1	100.520	1.509	2264	153000
12	5	2	108.530	1.562	2032	190000
12	5	3	115.810	1.444	2437	162000
12	5	4	124.760	1.483	2031	164000
12	5	5	133.270	1.502	2397	153000
13	1	1	98.310	32.156	1627	204000
13	1	2	109.240	23.552	1964	200000
13	1	3	118.920	27.132	1688	195000
13	1	4	129.720	32.659	2034	204000
13	1	5	133.800	28.118	1932	187000
13	2	1	97.783	1.274	2356	208000
13	2	2	109.370	1.310	2712	218000
13	2	3	115.410	1.258	2335	218000
13	2	4	129.340	1.286	2514	178000
13	2	5	130.330	1.324	1613	209000
13	3	1	96.751	1.359	1582	209000
13	3	2	109.600	1.331	1844	214000
13	3	3	116.660	1.335	2765	208000
13	3	4	126.160	1.284	2289	195000
13	3	5	136.020	1.312	2602	182000
13	4	1	97.002	1.334	1630	208000
13	4	2	108.160	1.334	1551	213000
13	4	3	116.120	1.319	1529	205000
13	4	4	127.040	1.303	2210	188000
13	4	5	132.460	1.360	1964	208000
13	5	1	99.001	1.321	2235	208000
13	5	2	112.490	1.303	2483	199000
13	5	3	117.140	1.288	1689	205000
13	5	4	125.850	1.345	2196	208000
13	5	5	131.620	1.328	2322	200000
19	1	1	155.760	361.310	872	745000
19	1	2	164.730	361.410	928	731000
19	1	3	153.520	361.290	783	766000
19	1	4	158.380	361.310	932	725000
19	1	5	167.420	361.320	1279	729000
19	2	1	132.130	14.735	1649	723000

Table 55 (continued)

19	2	2	140.110	14.946	1233	775000
19	2	3	134.850	14.430	995	730000
19	2	4	131.760	17.292	999	725000
19	2	5	137.300	14.293	912	848000
19	3	1	138.200	16.956	1489	755000
19	3	2	140.060	17.551	1044	750000
19	3	3	134.230	16.333	884	729000
19	3	4	143.040	15.883	1694	712000
19	3	5	138.060	17.959	1875	742000
19	4	1	129.570	15.865	1282	703000
19	4	2	157.260	16.776	1250	769000
19	4	3	137.350	18.330	950	767000
19	4	4	136.430	15.301	993	761000
19	4	5	138.620	14.539	968	787000
19	5	1	133.170	16.404	1024	790000
19	5	2	148.380	17.761	1299	738000
19	5	3	136.160	14.074	1454	699000
19	5	4	131.120	16.620	953	855000
19	5	5	141.710	16.967	876	774000
20	1	1	128.050	289.190	362	745000
20	1	2	147.880	333.870	689	731000
20	1	3	129.430	305.710	365	766000
20	1	4	131.890	360.420	377	725000
20	1	5	137.850	319.420	464	729000
20	2	1	128.050	289.190	362	844000
20	2	2	147.880	333.870	689	815000
20	2	3	129.430	305.710	365	876000
20	2	4	131.890	360.420	377	843000
20	2	5	137.850	319.420	464	855000
20	3	1	130.000	307.820	374	839000
20	3	2	149.100	322.280	499	829000
20	3	3	144.310	281.170	728	821000
20	3	4	133.560	360.410	430	838000
20	3	5	133.240	295.170	362	944000
20	4	1	136.100	334.800	387	919000
20	4	2	143.970	317.310	418	934000
20	4	3	141.030	332.200	380	964000
20	4	4	147.630	360.460	559	893000
20	4	5	151.070	310.930	444	803000
20	5	1	148.560	337.860	444	846000
20	5	2	161.340	351.480	547	746000
20	5	3	146.450	286.740	446	841000

Table 55 (continued)

20	5	4	152.590	360.440	590	805000
20	5	5	162.000	328.240	716	818000
21	1	1	169.100	361.390	1118	733000
21	1	2	159.130	361.370	1611	715000
21	1	3	160.520	361.620	1017	788000
21	1	4	154.170	361.290	2273	697000
21	1	5	154.730	361.290	1084	793000
21	2	1	133.860	9.569	1291	775000
21	2	2	135.330	9.063	1256	722000
21	2	3	132.090	9.236	1242	728000
21	2	4	148.370	11.518	1128	775000
21	2	5	160.560	11.979	1650	717000
21	3	1	129.880	9.912	1404	727000
21	3	2	127.350	11.852	1026	716000
21	3	3	131.710	9.661	1404	690000
21	3	4	156.210	11.153	1523	693000
21	3	5	150.300	11.410	1340	797000
21	4	1	137.870	12.344	1622	706000
21	4	2	146.400	10.822	1512	733000
21	4	3	128.720	10.609	1222	742000
21	4	4	135.670	10.455	1591	702000
21	4	5	123.340	11.871	869	692000
21	5	1	134.670	10.815	1581	707000
21	5	2	132.150	11.121	979	778000
21	5	3	129.040	12.275	1165	772000
21	5	4	134.000	14.475	1483	734000
21	5	5	135.220	11.375	1634	674000
22	1	1	444.190	361.430	378	890000
22	1	2	520.710	361.480	428	831000
22	1	3	444.190	361.430	378	860000
22	1	4	444.190	361.430	428	769000
22	1	5	520.710	361.480	428	841000
22	2	1	153.080	119.490	652	822000
22	2	2	140.580	122.100	647	805000
22	2	3	156.640	101.520	837	827000
22	2	4	158.650	123.010	819	767000
22	2	5	167.830	104.340	839	756000
22	3	1	157.570	120.780	982	735000
22	3	2	142.130	114.460	654	791000
22	3	3	139.730	121.420	536	842000
22	3	4	144.930	112.250	359	839000
22	3	5	164.680	98.093	521	811000

Table 55 (continued)

22	4	1	141.540	121.620	582	831000
22	4	2	140.500	98.328	447	837000
22	4	3	153.010	109.290	637	822000
22	4	4	159.030	113.440	610	765000
22	4	5	155.280	125.840	484	756000
22	5	1	137.940	138.080	441	880000
22	5	2	144.400	115.810	637	838000
22	5	3	147.400	122.770	534	747000
22	5	4	150.000	124.040	504	782000
22	5	5	175.190	126.980	659	781000
23	1	1	137.670	360.320	3865	805000
23	1	2	134.410	360.500	3505	801000
23	1	3	128.290	358.850	3509	863000
23	1	4	133.020	310.940	3900	878000
23	1	5	136.750	359.790	3535	836000
23	2	1	134.810	2.940	4385	884000
23	2	2	136.740	2.704	3211	840000
23	2	3	145.190	2.622	5601	875000
23	2	4	147.190	2.840	2727	924000
23	2	5	159.410	2.770	5093	890000
23	3	1	138.850	2.936	3652	874000
23	3	2	138.620	2.808	6122	820000
23	3	3	143.820	2.874	2770	861000
23	3	4	148.390	2.974	4347	888000
23	3	5	153.990	2.501	3257	915000
23	4	1	135.380	2.835	2483	855000
23	4	2	136.450	2.774	6187	866000
23	4	3	141.370	2.760	2164	870000
23	4	4	155.330	2.739	3396	882000
23	4	5	156.490	2.782	4363	868000
23	5	1	139.340	3.031	6800	828000
23	5	2	136.400	2.924	3695	848000
23	5	3	144.630	2.778	3894	840000
23	5	4	147.150	2.550	3579	857000
23	5	5	153.850	2.650	2875	869000
24	1	1	262.230	361.490	800	902000
24	1	2	421.850	361.480	896	933000
24	1	3	327.770	361.480	875	1080000
24	1	4	218.420	361.480	825	998000
24	1	5	249.380	361.430	1290	988000
24	2	1	154.980	17.209	1491	972000
24	2	2	144.100	16.293	1664	1010000

Table 55 (continued)

24	2	3	156.310	15.699	1081	982000
24	2	4	153.060	15.913	881	1030000
24	2	5	155.920	16.095	1029	1060000
24	3	1	153.100	17.820	1952	976000
24	3	2	144.920	14.214	1186	959000
24	3	3	165.050	14.617	1479	924000
24	3	4	156.460	14.475	836	964000
24	3	5	162.660	14.869	1433	1060000
24	4	1	143.210	16.762	1961	975000
24	4	2	150.410	14.299	2169	983000
24	4	3	169.130	14.217	2735	1020000
24	4	4	155.140	14.112	2120	984000
24	4	5	158.630	16.790	1113	1010000
24	5	1	143.600	15.599	1032	1070000
24	5	2	150.560	17.447	2629	931000
24	5	3	165.200	16.577	3335	909000
24	5	4	157.520	14.981	2409	1000000
24	5	5	159.430	16.232	1265	943000
25	1	1	129.640	201.840	3743	864000
25	1	2	136.570	215.470	4948	873000
25	1	3	133.540	214.320	5357	818000
25	1	4	134.960	210.390	3523	860000
25	1	5	132.040	215.840	3512	856000
25	2	1	132.780	2.152	3788	843000
25	2	2	131.260	1.978	4199	850000
25	2	3	137.950	2.074	3700	869000
25	2	4	146.600	2.002	4456	843000
25	2	5	153.470	2.044	5157	852000
25	3	1	135.130	1.983	5035	836000
25	3	2	134.110	1.860	3323	848000
25	3	3	138.520	2.002	3810	921000
25	3	4	151.850	2.053	6291	829000
25	3	5	151.800	2.048	4935	827000
25	4	1	137.970	2.219	4900	842000
25	4	2	135.610	1.928	5207	851000
25	4	3	139.360	1.872	4252	827000
25	4	4	152.880	2.011	5726	849000
25	4	5	149.870	2.040	4001	859000
25	5	1	137.700	2.046	5680	880000
25	5	2	134.050	1.857	3653	924000
25	5	3	143.090	1.922	5368	848000
25	5	4	149.000	2.060	6508	853000

Table 55 (continued)

25	5	5	152.910	1.982	4837	860000
26	1	1	285.170	361.500	2199	970000
26	1	2	458.790	361.490	859	1010000
26	1	3	227.050	361.410	1911	1060000
26	1	4	299.730	361.370	1303	1000000
26	1	5	254.210	361.430	1602	911000
26	2	1	150.030	22.208	2734	993000
26	2	2	140.650	24.937	1281	1030000
26	2	3	154.490	23.040	2274	931000
26	2	4	160.840	25.497	1858	1030000
26	2	5	167.260	18.980	2304	979000
26	3	1	146.420	23.595	1673	938000
26	3	2	143.640	23.557	1493	962000
26	3	3	148.630	19.866	1490	1000000
26	3	4	157.690	18.710	1436	1000000
26	3	5	163.090	20.105	2622	957000
26	4	1	156.870	18.715	2266	890000
26	4	2	151.010	18.898	2100	973000
26	4	3	153.460	16.933	2366	970000
26	4	4	162.200	19.545	1571	1020000
26	4	5	166.790	18.561	1502	996000
26	5	1	149.070	17.600	1301	1040000
26	5	2	146.260	17.910	1570	1010000
26	5	3	144.950	18.704	1016	1020000
26	5	4	165.810	18.075	3104	982000
26	5	5	163.440	19.651	1638	989000

G.3 Experiment II

Table 56 – Population mean and optimal solution objective function values for the various approaches

Problem	Optimal Solution			Population Mean Solutions		
	R	RA	A	R	RA	A
1	78751	98424	92673	294000	314200	290600
2	85895	132594	92056	315400	499000	315800
3	88070	120400	95301	358200	373600	361200
4	83630	109011	88043	292400	299000	305000
5	82041	104247	88411	319400	328000	315800
6	116200	133989	129800	340600	391600	332600
7	188600	183750	166000	374000	377600	352800
8	149000	164600	141600	350600	342200	334000
9	181800	182000	157800	346200	331000	322800
10	122800	121600	125600	366600	380000	358800
11	159800	142800	142000	378600	363400	367600
12	133400	138800	141400	354600	352600	355600
13	159000	162400	149600	376000	367000	359200
14	162600	186800	163800	416400	1438000	398000
15	161200	235500	171200	434200	1404000	406600
16	168200	238750	171000	468400	804200	471400
17	156800	170400	171200	415600	399000	400200
18	175600	217000	159600	459000	777400	418400
19	532200	455800	430800	701600	1219000	626400
20	596000	532800	509400	920600	1074400	802200
21	507200	445000	419400	666600	587600	600800
22	609400	533200	502400	908200	797000	811200
23	423200	392000	359200	616400	822600	558000
24	475400	435500	398000	926400	1382200	777400
25	410000	367250	333600	584800	1319000	523200
26	469600	428500	387800	791000	1278000	774800
27	339200	448200	330000	798800	834200	766400
28	330600	407200	324800	855600	820800	807200
29	307800	380200	322200	767000	712000	807800
30	298000	367200	307800	753400	750400	766400
31	305200	410600	333800	715200	676600	818600
32	457200	480200	437000	907400	849400	884800
33	692600	638800	557400	1024600	935600	911800
34	518400	585200	493200	951600	910400	860200
35	632600	591600	563600	927000	869800	927200
36	422400	476400	438800	831400	755600	945000

Table 56 (continued)

37	553800	528800	512400	928600	799200	949000
38	458800	529600	441800	910200	825200	864400
39	559000	557800	530800	961600	846200	943600
40	744200	798600	1020000	1430000		1593333
41	650200	1956200	952333	1286000		1810000
42	663400	742200	844000	1276000		1900000
43	682600	784200	907667	1268000		1596667
44	598400	605000	763333	1160000		1463333
45	1710000	2693333	1486667	2116000		1953333
46	1916000	1726000	1593333	2380000		2163333
47	1644000	1476000	1473333	2020000		1983333
48	1948000	1664000	1586667	2344000		2046667
49	1370000	1318000	1210000	1742000		1773333
50	1534000	1336667	1256667	2108000		1960000
51	1322000	1186000	1123333	1716000		1606667
52	1516000	1364000	1280000	2078000		1683333

Table 57 – Solution times and unique designs generated for the various approaches

Problem	Solution Times			Unique Solutions		
	R	RA	A	R	RA	A
1	115	na	807	4751	na	3970
2	120	na	851	3692	na	4104
3	121	na	738	2312	na	2514
4	121	na	778	3507	na	3355
5	120	na	883	2598	na	2581
6	119	na	537	3431	na	2844
7	119	na	612	2155	na	2278
8	119	na	860	2490	na	3014
9	118	na	603	2705	na	2353
10	122	na	754	2069	na	2355
11	119	na	623	1686	na	1731
12	121	na	735	2408	na	2083
13	119	na	550	2033	na	2006
14	133	na	10716	9731	na	8283
15	138	na	7603	5698	na	6797
16	139	na	5951	4917	na	4324
17	137	na	7749	8466	na	7049
18	141	na	5767	4545	na	5372
19	140	na	5992	8497	na	7195
20	143	na	2586	3723	na	4541

Table 57 (continued)

21	141	na	5636	8947	na	6158
22	143	na	2100	3225	na	3145
23	139	na	5627	5602	na	5272
24	141	na	2450	1857	na	2106
25	136	na	5852	7412	na	5323
26	147	na	2252	2645	na	2321
27	222	na	3306	13540	na	11201
28	224	na	3299	8839	na	9481
29	235	na	2607	7881	na	6667
30	228	na	3317	8987	na	9143
31	237	na	2245	7791	na	5736
32	231	na	2534	8151	na	9711
33	230	na	1692	10460	na	9980
34	225	na	2350	7040	na	9964
35	225	na	1682	9319	na	8039
36	231	na	1976	8692	na	5650
37	232	na	1720	6941	na	6120
38	242	na	2051	6502	na	8006
39	245	na	1923	6457	na	6431
40	308	na	12971	15625	na	3386
41	316	na	12742	12810	na	3526
42	323	na	12645	8448	na	2930
43	314	na	12882	13687	na	4019
44	320	na	12688	10713	na	6170
45	321	na	11932	14140	na	10957
46	348	na	8201	9859	na	8037
47	316	na	12741	15031	na	7776
48	344	na	5570	9591	na	7937
49	343	na	12678	12004	na	7715
50	343	na	7287	5046	na	4803
51	333	na	12762	13252	na	8631
52	340	na	9759	6088	na	6693

G.4 Experiment III

Table 58 – Solution times and optimal value for FSA on and off approaches

Problem	Optimal Solution		Solution Times	
	FSA On	FSA Off	FSA On	FSA Off
1	77596	78751	2259	115
2	75141	85895	2266	120
3	78166	88070	2220	121
4	73081	83630	2202	121
5	71569	82041	2205	120
6	107600	116200	2250	119
7	185600	188600	2294	119
8	122000	149000	2281	119
9	176400	181800	2325	118
10	109600	122800	2279	122
11	151200	159800	2321	119
12	110400	133400	2275	121
13	161000	159000	2329	119
14	150400	162600	3454	133
15	171800	161200	3451	138
16	143800	168200	3434	139
17	152000	156800	3401	137
18	156600	175600	3319	141
19	541800	532200	3388	140
20	595200	596000	3366	143
21	511200	507200	3634	141
22	601600	609400	3599	143
23	433400	423200	3593	139
24	462000	475400	3574	141
25	403800	410000	3551	136
26	473600	469600	3520	147
27	322600	339200	12655	222
28	289200	330600	12512	224
29	284600	307800	12060	235
30	301800	298000	12610	228
31	304200	305200	12661	237
32	470000	457200	12677	231
33	688600	692600	12659	230
34	501200	518400	12688	225
35	642400	632600	12741	225
36	395800	422400	12213	231
37	582400	553800	12709	232

Table 58 (continued)

38	454600	458800	12654	242
39	574200	559000	12670	245
40	715000	744200	12782	308
41	578400	650200	12675	316
42	563400	663400	12692	323
43	568200	682600	12733	314
44	581600	598400	12764	320
45	1688000	1710000	12706	321
46	1890000	1916000	12872	348
47	1618000	1644000	12776	316
48	1918000	1948000	12961	344
49	1350000	1370000	12705	343
50	1480000	1535000	13105	343
51	1294000	1322000	12680	333
52	1468000	1516000	12838	340

Table 59 – Time to optimality and time per generations for FSA approaches

Problem	Time to Optimal (sec)		Time/Generation (sec)	
	FSA On	FSA Off	FSA On	FSA Off
1	1360	80	11.29	0.58
2	1312	85	11.33	0.61
3	1507	76	11.10	0.62
4	1232	88	11.01	0.61
5	1804	74	11.03	0.61
6	1579	97	11.25	0.61
7	1552	70	11.47	0.61
8	1251	68	11.40	0.60
9	1664	68	11.63	0.60
10	1418	69	11.40	0.62
11	1273	63	11.61	0.61
12	1391	72	11.38	0.61
13	1299	65	11.65	0.60
14	3082	126	17.27	0.68
15	2954	129	17.26	0.70
16	2775	134	17.17	0.71
17	2706	130	17.01	0.70
18	2786	126	16.60	0.72
19	2497	126	16.94	0.71
20	2509	107	16.83	0.75
21	2952	121	18.17	0.72

Table 59 (continued)

22	2485	119	17.99	0.75
23	2549	107	17.96	0.71
24	2351	117	17.87	0.79
25	2893	110	17.76	0.69
26	2690	117	17.60	0.79
27	11667	204	98.53	1.13
28	9816	193	64.27	1.14
29	9475	214	60.30	1.20
30	9614	211	69.34	1.16
31	9995	211	65.00	1.21
32	10014	203	79.64	1.18
33	10754	189	98.93	1.17
34	10275	171	87.39	1.14
35	11333	198	103.77	1.15
36	8106	213	61.07	1.18
37	9261	171	82.25	1.18
38	8737	180	72.93	1.23
39	10434	206	88.12	1.25
40	12131	305	163.02	1.56
41	12001	295	101.43	1.62
42	12159	311	96.64	1.66
43	12227	305	99.07	1.60
44	12059	299	107.76	1.63
45	11268	294	109.12	1.64
46	12355	350	106.46	1.87
47	11825	296	115.12	1.61
48	12296	339	108.37	1.84
49	11574	298	107.13	1.76
50	11648	381	102.26	2.14
51	12173	303	107.16	1.70
52	10477	344	97.67	1.96

Table 60 – Time to FSA off optimality for the FSA on approach

Problem	Time (sec)
1	1627
2	1257
3	1346
4	1244
5	1426
6	1435
7	1627
8	506
9	1552
10	1362
11	1213
12	407
13	1804
14	2501
15	3266
16	1770
17	3000
18	2368
19	2965
20	2569
21	3162
22	2236
23	3293
24	2252
25	2423
26	2602
27	10608
28	7974
29	8074
30	10029
31	8270
32	11290
33	10864
34	9443
35	11053
36	6472
37	10075
38	9666
39	11562
40	10652
41	9847

Table 60 (continued)

42	7903
43	8739
44	11342
45	10692
46	11058
47	10620
48	9428
49	9397
50	7652
51	9051
52	5998

G.5 Experiment IV

Table 61 - L18 orthogonal array leveraged for parameter screening in Stage One

	Control Factors						
Exp. No.	**A**	**B**	**C**	**D**	**E**	**F**	**G**
1	1	1	1	1	2	2	1
2	1	2	2	2	1	1	1
3	1	1	2	2	1	1	2
4	1	2	1	1	2	2	2
5	1	1	2	1	1	2	3
6	1	2	1	2	2	1	3
7	2	1	1	2	2	1	1
8	2	2	2	1	1	2	1
9	2	1	1	2	1	2	2
10	2	2	2	1	2	1	2
11	2	1	2	1	2	1	3
12	2	2	1	2	1	2	3
13	3	1	2	2	2	2	1
14	3	2	1	1	1	1	1
15	3	1	1	1	1	1	2
16	3	2	2	2	2	2	2
17	3	1	1	1	1	1	3
18	3	2	2	2	2	2	3

Table 62 – RDP and S/N ratios for the optimal objective value Stage One results across the 26 SLPs

Exp No.	**Witness**							**S/N Ratio**
	RPD 1	RPD 2	RPD 3	…	RPD 24	RPD 25	RPD 26	
1	0.404	0.151	0.388	**…**	0.111	0.071	0.100	9.250
2	0.327	0.798	0.573	**…**	0.142	0.088	0.050	8.669
3	0.437	0.383	0.647	**…**	0.099	0.076	0.066	8.819
4	0.241	0.609	0.796	**…**	0.078	0.153	0.076	9.991

Table 62 (continued)

5	0.492	0.507	0.394	…	0.049	0.155	0.230	9.546
6	0.046	0.432	0.881	…	0.047	0.009	0.075	10.570
7	0.582	0.129	0.300	…	0.059	0.052	0.036	10.955
8	0.364	0.568	0.390	…	0.123	0.130	0.125	10.850
9	0.567	0.568	0.343	…	0.045	0.073	0.050	11.400
10	0.315	0.317	0.572	…	0.113	0.091	0.063	11.134
11	0.224	0.269	0.209	…	0.079	0.092	0.077	13.325
12	0.510	0.201	0.846	…	0.086	0.055	0.053	10.611
13	0.280	0.279	0.343	…	0.055	0.068	0.063	12.023
14	0.316	0.352	0.424	…	0.076	0.027	0.042	11.612
15	0.087	0.148	0.293	…	0.042	0.047	0.068	15.846
16	0.326	0.337	0.318	…	0.033	0.010	0.062	12.914
17	0.249	0.093	0.333	…	0.037	0.054	0.052	13.101
18	0.252	0.291	0.246	…	0.048	0.049	0.021	16.753

Table 63 – RDP and S/N ratios for the solution time Stage One results across the 26 SLPs

Exp No.	Witness							S/N Ratio
	RPD 1	RPD 2	RPD 3	…	RPD 24	RPD 25	RPD 26	
1	0.637	0.092	0.187	…	0.276	0.272	0.335	10.126
2	0.211	0.065	0.166	…	0.243	0.287	0.342	9.086
3	0.841	0.705	0.843	…	0.922	0.949	1.027	-0.569
4	0.834	0.732	0.896	…	0.986	1.031	1.082	-0.780
5	1.835	1.484	1.615	…	1.577	1.776	1.777	-5.434
6	1.927	1.552	1.705	…	1.649	1.794	1.825	-5.698
7	1.373	1.077	1.107	…	0.768	0.745	0.788	0.025
8	1.360	1.050	1.096	…	0.721	0.739	0.723	0.350
9	2.864	2.417	2.397	…	1.572	1.678	1.649	-6.320
10	2.879	2.507	2.377	…	1.542	1.653	1.614	-6.272
11	4.589	3.980	3.809	…	2.405	2.643	2.574	-10.221
12	4.615	4.051	3.776	…	2.297	2.617	2.540	-10.128
13	3.733	3.184	2.980	…	1.563	1.454	1.633	-6.863
14	3.692	3.151	2.965	…	1.407	1.435	1.472	-6.609
15	6.529	5.618	5.424	…	2.675	2.697	2.677	-11.819
16	6.617	5.679	5.569	…	2.764	2.744	2.842	-11.963
17	9.688	8.551	7.835	…	4.029	4.056	4.039	-15.319
18	10.021	8.733	8.096	…	4.234	4.313	4.363	-15.715

Table 64 – RDP and S/N ratios for the unique solution Stage One results across the 26 SLPs

Exp No.	Witness							S/N Ratio
	RPD 1	RPD 2	RPD 3	…	RPD 24	RPD 25	RPD 26	
1	0.376	1.305	0.148	…	0.262	0.553	3.442	-15.751
2	0.706	0.158	0.434	…	1.797	0.930	5.391	-8.543
3	0.707	1.614	0.266	…	2.839	1.726	4.615	-11.349
4	0.956	1.698	0.603	…	1.173	1.341	6.666	-6.655
5	0.006	0.657	0.029	…	0.157	0.058	0.647	-34.617
6	1.864	1.574	1.179	…	2.561	2.770	7.538	3.336
7	3.946	4.313	3.522	…	8.183	5.539	23.121	11.627
8	0.905	1.510	1.573	…	1.065	2.551	4.825	3.206
9	2.957	3.014	2.077	…	3.165	3.938	26.100	9.595
10	2.740	5.445	2.421	…	1.280	3.230	10.919	9.218
11	3.455	7.034	4.985	…	3.632	4.266	6.925	11.448
12	3.940	4.670	2.362	…	5.406	5.349	23.394	11.265
13	8.351	14.844	9.190	…	5.904	12.789	37.075	19.012
14	4.885	6.881	5.829	…	5.239	7.959	15.958	13.378
15	5.794	8.476	6.242	…	11.509	10.567	14.477	14.904
16	13.482	15.424	8.497	…	17.606	18.072	60.440	21.187
17	4.197	9.362	4.281	…	18.986	8.051	18.675	14.520
18	8.506	18.454	12.509	…	14.239	20.652	43.053	22.008

Table 65 – RDP and S/N ratios for the optimal objective value Stage One results across the 26 DLPs

Exp No.	Witness							S/N Ratio
	RPD 27	RPD 28	RPD 29	…	RPD 50	RPD 51	RPD 52	
1	0.659	0.359	0.669	…	0.110	0.070	0.138	8.836
2	0.844	0.668	0.730	…	0.075	0.070	0.116	7.965
3	0.134	0.446	0.529	…	0.075	0.102	0.079	10.365
4	0.270	0.511	0.195	…	0.086	0.084	0.146	11.686
5	0.241	0.597	0.440	…	0.073	0.012	0.089	11.356
6	0.046	0.591	0.473	…	0.056	0.059	0.052	11.297
7	0.538	0.806	0.575	…	0.119	0.144	0.118	8.085
8	0.314	0.517	0.460	…	0.081	0.084	0.118	10.686
9	0.362	0.469	0.418	…	0.071	0.064	0.167	11.235
10	0.544	0.426	0.186	…	0.066	0.016	0.051	12.512
11	0.241	0.288	0.345	…	0.045	0.050	0.033	13.948
12	0.347	0.357	0.131	…	0.048	0.054	0.072	12.644
13	0.877	0.441	0.673	…	0.107	0.067	0.094	8.355
14	0.334	0.480	0.452	…	0.086	0.048	0.078	11.282

Table 65 (continued)

15	0.217	0.120	0.637	…	0.036	0.016	0.108	13.387
16	0.196	0.639	0.370	…	0.066	0.064	0.091	11.028
17	0.096	0.182	0.336	…	0.053	0.029	0.061	15.859
18	0.200	0.164	0.133	…	0.018	0.050	0.038	15.243

Table 66 – RDP and S/N ratios for the solution time Stage One results across the 26 DLPs

Exp No.	**Witness**							**S/N Ratio**
	RPD 27	RPD 28	RPD 29	…	RPD 50	RPD 51	RPD 52	
1	0.285	0.858	0.788	…	0.554	0.536	0.396	1.880
2	0.249	0.714	0.874	…	0.623	0.578	0.429	4.989
3	0.880	1.871	1.787	…	1.362	0.990	1.120	-2.171
4	0.919	1.723	1.684	…	1.270	1.019	1.026	-2.528
5	1.556	2.401	2.625	…	1.971	1.851	1.629	-5.893
6	1.595	2.527	2.650	…	2.041	1.935	1.702	-6.233
7	0.688	1.732	2.046	…	1.657	1.246	1.317	-3.579
8	0.691	1.661	1.934	…	1.469	1.150	1.182	-2.799
9	1.499	3.270	3.713	…	2.716	2.231	2.232	-8.599
10	1.493	3.244	3.410	…	2.420	2.227	2.016	-8.356
11	2.340	4.687	4.792	…	3.470	3.276	3.107	-11.590
12	2.389	4.724	5.023	…	3.676	3.467	3.056	-11.842
13	1.487	3.746	4.702	…	3.852	2.800	3.324	-10.672
14	1.433	3.535	4.194	…	3.013	2.534	2.641	-9.892
15	2.667	5.948	6.628	…	4.597	4.219	3.989	-14.272
16	2.705	6.458	7.138	…	5.518	4.560	4.903	-14.941
17	3.991	8.404	9.105	…	6.431	6.007	5.597	-17.362
18	4.217	8.077	9.665	…	7.579	6.604	5.929	-17.915

Table 67 – RDP and S/N ratios for the unique solution Stage One results across the 26 DLPs

Exp No.	**Witness**							**S/N Ratio**
	RPD 27	RPD 28	RPD 29	…	RPD 50	RPD 51	RPD 52	
1	0.034	0.020	0.363	…	0.195	0.111	0.113	-38.268
2	0.268	0.342	0.427	…	1.110	0.464	0.554	-11.853
3	0.571	0.414	0.234	…	3.946	0.759	0.846	-12.260
4	0.538	0.297	0.316	…	2.021	0.599	0.244	-12.797
5	0.134	0.088	0.213	…	2.300	1.528	0.481	-14.139
6	1.352	1.372	0.741	…	2.611	1.573	0.569	1.526

Table 67 (continued)

7	1.534	1.699	2.037	…	2.943	1.700	1.259	5.118
8	1.295	1.357	1.367	…	4.507	1.469	1.184	2.955
9	2.554	2.729	2.839	…	3.615	3.089	2.345	8.983
10	2.558	2.404	1.831	…	2.808	2.332	2.731	7.174
11	2.652	3.840	1.106	…	6.100	4.255	3.319	9.106
12	3.859	3.760	1.340	…	13.157	4.095	4.657	10.492
13	4.167	4.522	6.834	…	13.392	4.987	7.832	14.590
14	3.468	2.971	2.785	…	3.659	3.545	4.677	10.897
15	5.404	4.079	3.540	…	5.503	4.842	3.367	13.214
16	6.629	7.147	6.424	…	13.627	7.934	8.167	17.725
17	5.831	3.869	2.318	…	12.990	6.799	4.083	13.733
18	9.219	7.891	9.815	…	23.258	10.598	15.938	20.178

Table 68 – S/N ratios for the Stage One metrics of interest

Exp. No.	**S/N Ratios (26 SLPs)**			**S/N Ratios (26 DLPs)**		
	Objective	Time	Unique	Objective	Time	Unique
1	9.250	10.126	-15.751	8.836	1.880	-38.268
2	8.669	9.086	-8.543	7.965	4.989	-11.853
3	8.819	-0.569	-11.349	10.365	-2.171	-12.260
4	9.991	-0.780	-6.655	11.686	-2.528	-12.797
5	9.546	-5.434	-34.617	11.356	-5.893	-14.139
6	10.570	-5.698	3.336	11.297	-6.233	1.526
7	10.955	0.025	11.627	8.085	-3.579	5.118
8	10.850	0.350	3.206	10.686	-2.799	2.955
9	11.400	-6.320	9.595	11.235	-8.599	8.983
10	11.134	-6.272	9.218	12.512	-8.356	7.174
11	13.325	-10.221	11.448	13.948	-11.590	9.106
12	10.611	-10.128	11.265	12.644	-11.842	10.492
13	12.023	-6.863	19.012	8.355	-10.672	14.590
14	11.612	-6.609	13.378	11.282	-9.892	10.897
15	15.846	-11.819	14.904	13.387	-14.272	13.214

Table 68 (continued)

16	12.914	-11.963	21.187	11.028	-14.941	17.725
17	13.101	-15.319	14.520	15.859	-17.362	13.733
18	16.753	-15.715	22.008	15.243	-17.915	20.178

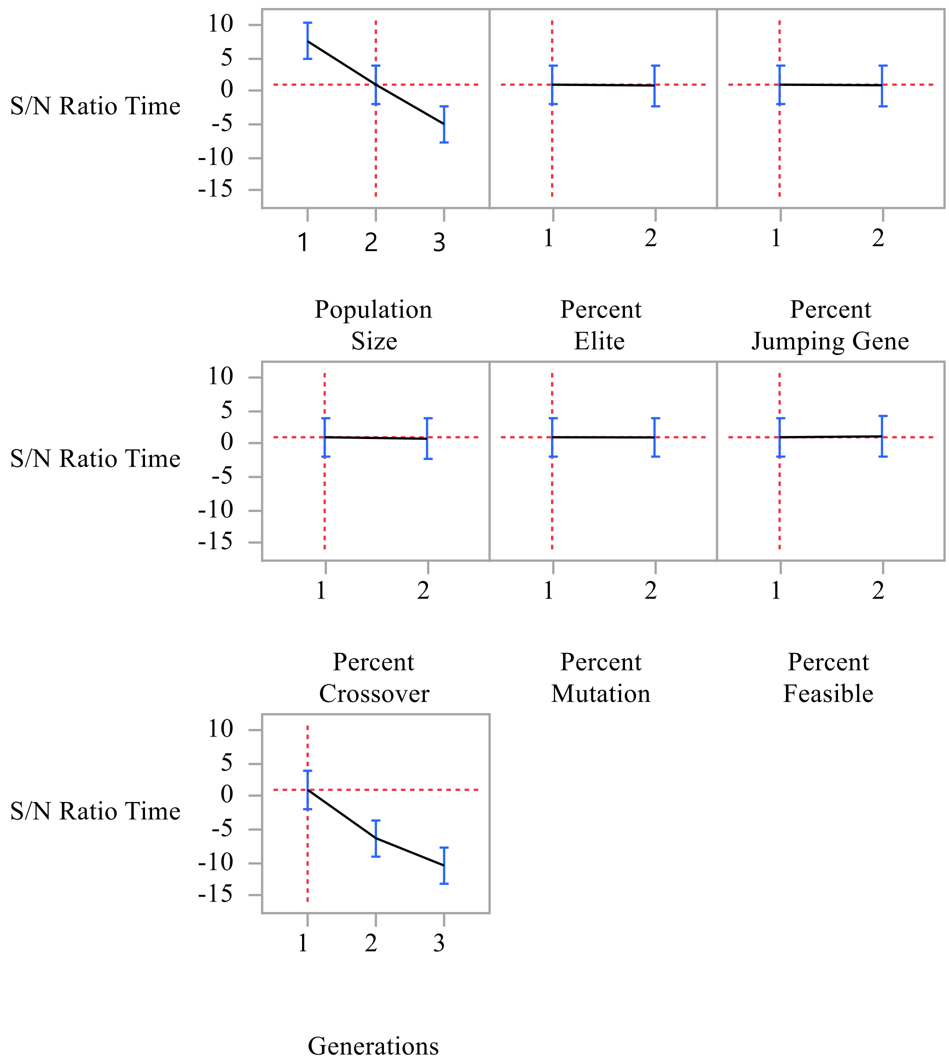


Figure 84 – Mean solution time S/N ratio for each level of the control factors for the 26 SLPs in Stage One

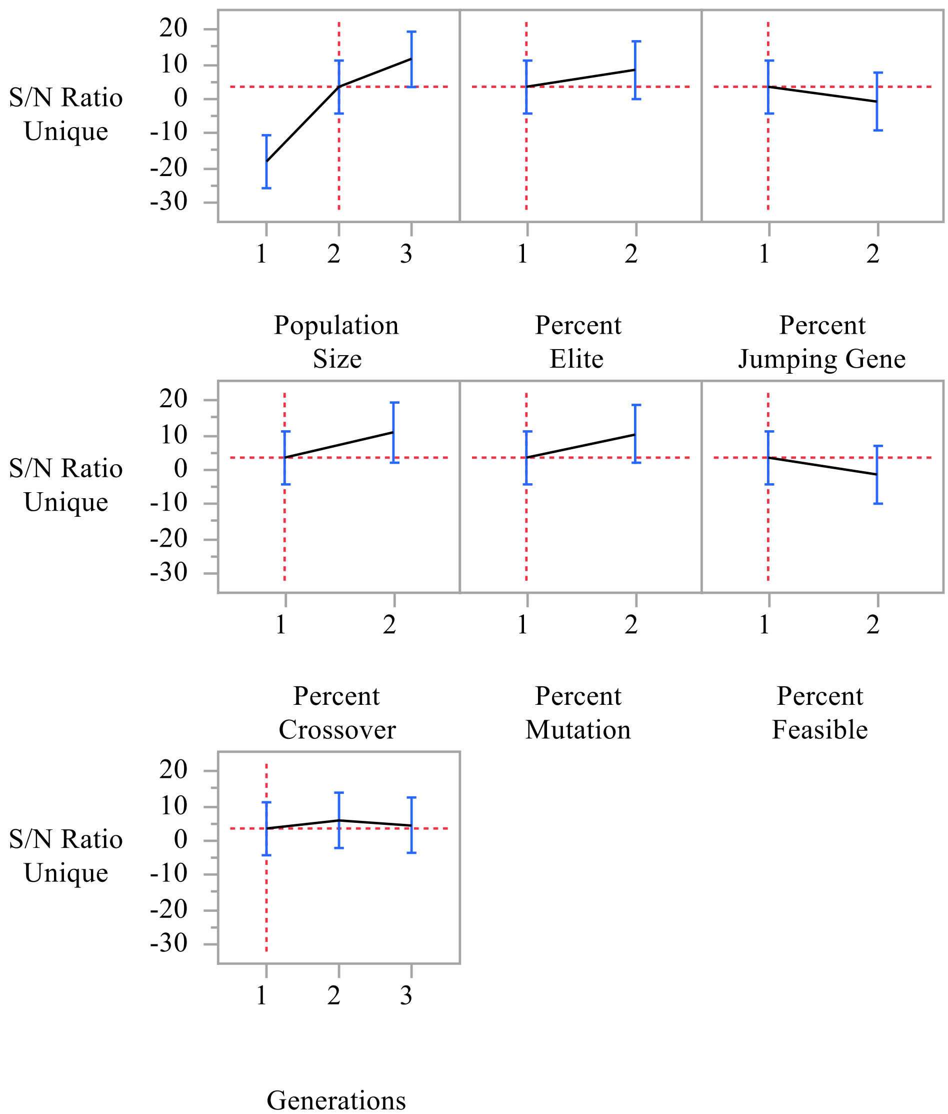


Figure 85 – Mean unique solutions S/N ratio for each level of the control factors for the 26 SLPs in Stage One

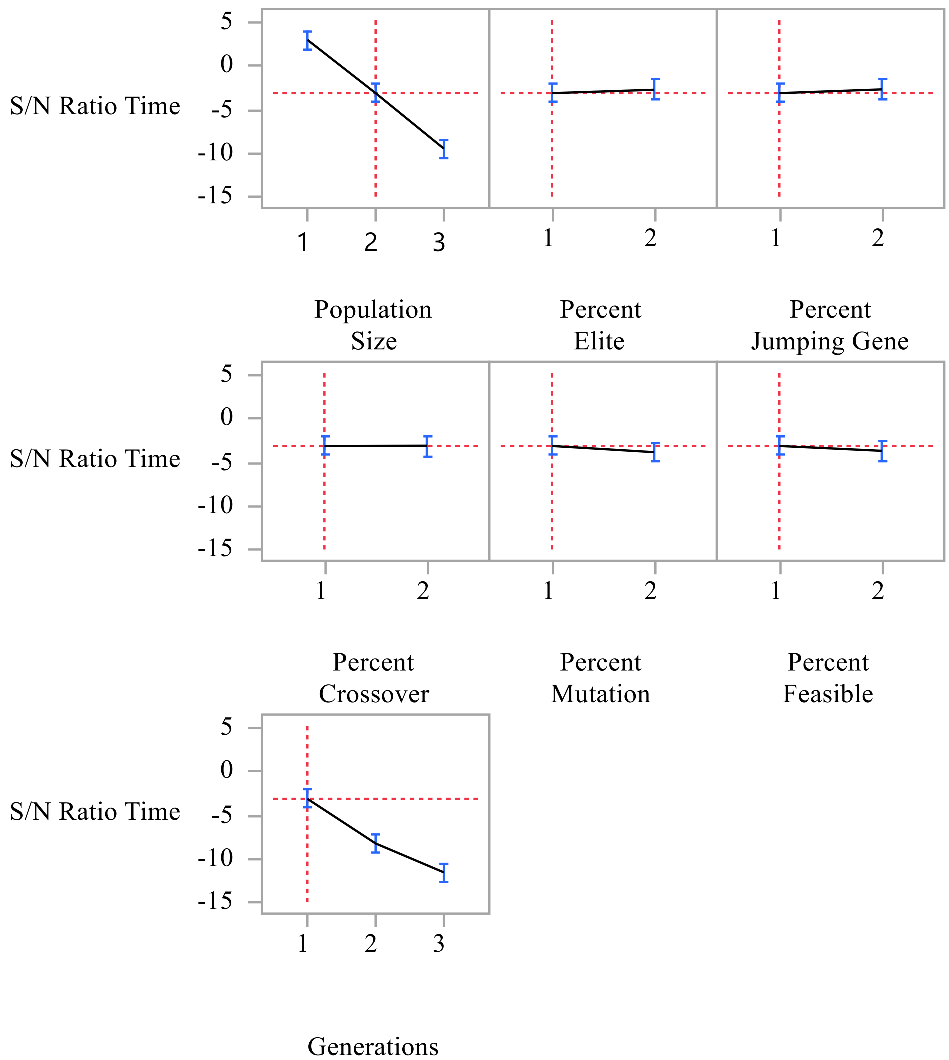


Figure 86 – Mean solution time S/N ratio for each level of the control factors for the 26 DLPs in Stage One

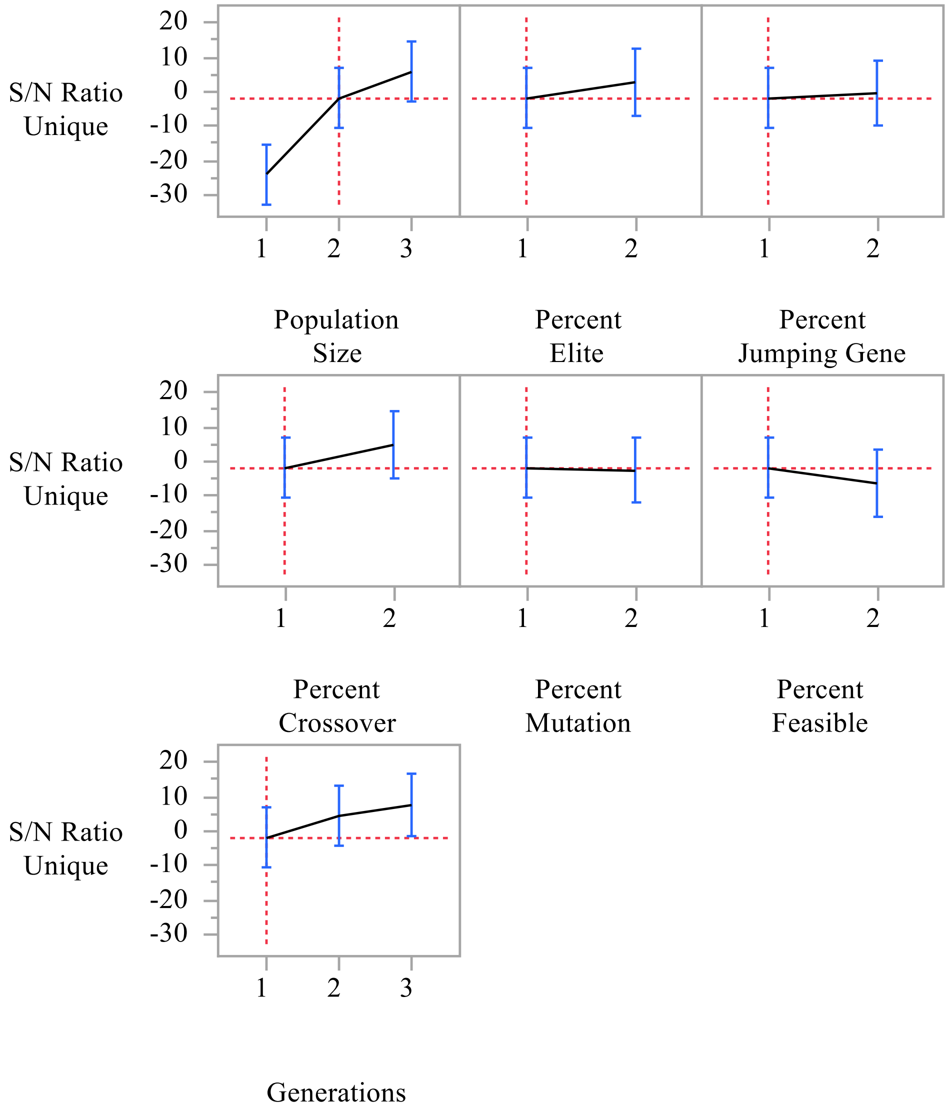


Figure 87 – Mean unique solutions S/N ratio for each level of the control factors for the 26 DLPs in Stage One

Table 69 – ANOVA of the optimal objective S/N ratio for the SLPs in Stage One

Factor		ANOVA Statistics				
Symbol	Description	DOF	SS	F-ratio	P-value	Pct. Contr.
A	Population Size	2	53.960	12.601	0.003	64.436
B	Percent Elite	1	0.136	0.064	0.807	0.163
C	Percent Jumping Gene	1	0.013	0.006	0.940	0.015
D	Percent Crossover	1	0.328	0.153	0.706	0.391
E	Percent Mutation	1	2.604	1.216	0.302	3.109
F	Percent Feasible Int. Pop.	1	0.061	0.029	0.870	0.073
G	Number of Generations	2	9.512	2.221	0.171	11.359
Error		8	17.128			20.454
Sum		17	83.742			100

Table 70 – ANOVA of the optimal objective S/N ratio for the DLPs in Stage One

Factor		ANOVA Statistics				
Symbol	Description	DOF	SS	F-ratio	P-value	Pct. Contr.
A	Population Size	2	15.592	8.188	0.012	17.729
B	Percent Elite	1	1.376	1.446	0.264	1.565
C	Percent Jumping Gene	1	0.108	0.113	0.745	0.123
D	Percent Crossover	1	9.334	9.804	0.014	10.614
E	Percent Mutation	1	0.327	0.344	0.574	0.372
F	Percent Feasible Int. Pop.	1	0.282	0.296	0.601	0.321
G	Number of Generations	2	53.310	27.996	0.000	60.616
Error		8	7.617			8.661
Sum		17	87.947			100

G.6 Experiment V

Table 71 - L18 orthogonal array leveraged for parameter screening in Stage Two

	Control Factors									
Exp. No.	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**
1	1	1	1	1	1	2	2	2	1	1
2	1	1	2	2	2	1	1	1	1	1
3	1	2	1	2	2	1	1	2	3	3
4	1	2	2	1	1	2	2	1	3	3
5	1	3	1	2	1	1	2	1	2	2
6	1	3	2	1	2	2	1	2	2	2
7	2	1	1	1	2	2	1	1	3	2
8	2	1	2	2	1	1	2	2	3	2
9	2	2	1	1	2	1	2	2	2	1
10	2	2	2	2	1	2	1	1	2	1
11	2	3	1	2	1	2	1	2	1	3
12	2	3	2	1	2	1	2	1	1	3
13	3	1	1	2	2	2	2	1	2	3
14	3	1	2	1	1	1	1	2	2	3
15	3	2	1	1	1	1	1	1	1	2
16	3	2	2	2	2	2	2	2	1	2
17	3	3	1	1	1	1	1	1	3	1
18	3	3	2	2	2	2	2	2	3	1

Table 72 – RDP and S/N ratios for the optimal objective value Stage Two results across the 26 SLPs

Exp. No.	**Witness**							**S/N Ratio**
	RPD 1	RPD 2	RPD 3	…	RPD 24	RPD 25	RPD 26	
1	0.227	0.142	0.139	**…**	0.029	0.031	0.018	21.410
2	0.262	0.111	0.139	**…**	0.030	0.035	0.021	17.853
3	0.238	0.142	0.139	**…**	0.016	0.034	0.010	18.033
4	0.194	0.129	0.139	**…**	0.023	0.034	0.025	18.851

Table 72 (continued)

5	0.223	0.139	0.111	…	0.030	0.027	0.026	18.686
6	0.171	0.140	0.139	…	0.017	0.022	0.014	18.960
7	0.191	0.124	0.139	…	0.020	0.020	0.021	19.128
8	0.231	0.140	0.139	…	0.016	0.032	0.016	19.033
9	0.219	0.128	0.139	…	0.018	0.028	0.024	19.591
10	0.225	0.123	0.139	…	0.016	0.028	0.020	19.443
11	0.244	0.103	0.139	…	0.021	0.031	0.015	19.277
12	0.210	0.142	0.132	…	0.022	0.020	0.024	19.299
13	0.216	0.125	0.139	…	0.012	0.023	0.014	18.975
14	0.200	0.091	0.139	…	0.016	0.020	0.016	19.547
15	0.181	0.124	0.139	…	0.016	0.018	0.022	19.860
16	0.209	0.096	0.139	…	0.016	0.024	0.023	19.788
17	0.190	0.135	0.139	…	0.006	0.031	0.006	20.232
18	0.211	0.117	0.119	…	0.011	0.024	0.014	20.433

Table 73 – RDP and S/N ratios for the optimal objective value Stage Two results across the 26 DLPs

Exp. No.	**Witness**							**S/N Ratio**
	RPD 27	RPD 28	RPD 29	…	RPD 50	RPD 51	RPD 52	
1	0.045	0.063	0.181	…	0.014	0.140	0.072	20.752
2	0.045	0.063	0.189	…	0.019	0.152	0.117	20.040
3	0.045	0.063	0.085	…	0.001	0.109	0.110	22.369
4	0.045	0.063	0.130	…	0.003	0.099	0.057	21.762
5	0.045	0.056	0.197	…	0.006	0.118	0.077	20.863
6	0.045	0.059	0.132	…	0.012	0.124	0.077	22.171
7	0.045	0.063	0.138	…	0.010	0.113	0.098	21.465
8	0.045	0.050	0.098	…	0.007	0.126	0.063	22.691
9	0.043	0.063	0.048	…	0.006	0.123	0.057	23.097
10	0.045	0.063	0.140	…	0.008	0.119	0.104	21.604
11	0.045	0.062	0.026	…	0.004	0.085	0.087	23.276
12	0.045	0.063	0.118	…	0.005	0.131	0.089	22.694
13	0.045	0.063	0.153	…	0.006	0.119	0.079	21.896
14	0.036	0.063	0.023	…	0.006	0.136	0.072	23.954
15	0.045	0.063	0.159	…	0.010	0.119	0.052	21.836
16	0.045	0.063	0.065	…	0.003	0.099	0.077	23.879
17	0.045	0.063	0.050	…	0.002	0.125	0.050	23.176
18	0.045	0.063	0.021	…	0.002	0.090	0.037	24.941

Table 74 – S/N ratios optimal objective metric in Stage Two

Exp. No.	S/N Ratios (26 SLPs)	S/N Ratios (13 DLPs)
	Objective	Objective
1	21.410	20.752
2	17.853	20.040
3	18.033	22.369
4	18.851	21.762
5	18.686	20.863
6	18.960	22.171
7	19.128	21.465
8	19.033	22.691
9	19.591	23.097
10	19.443	21.604
11	19.277	23.276
12	19.299	22.694
13	18.975	21.896
14	19.547	23.954
15	19.860	21.836
16	19.788	23.879
17	20.232	23.176
18	20.433	24.941

Table 75 – ANOVA of the optimal objective S/N ratio for the SLPs in Stage Two

Factor		ANOVA Statistics				
Symbol	Description	DOF	SS	F-ratio	P-value	Pct. Contr.
A	Population Size (1 | 3)	2	2.151	3.577	0.161	17.449
B	Population Size (2 | M)	2	0.154	0.256	0.789	1.250
C	Pct. Jmp. Gene (1 | 2 | 3 | M)	1	0.265	0.881	0.417	2.149
D	Pct. Crossover (1 | 2 | 3)	1	1.931	6.422	0.085	15.662
E	Pct. Crossover (M)	1	1.232	4.097	0.136	9.992

Table 75 (continued)

F	Pct. Mutation (1 | 2 | 3)	1	1.149	3.822	0.146	9.321
G	Pct. Mutation (M)	1	0.938	3.121	0.176	7.611
H	Iso. Generations (1 | 2 | 3)	1	0.944	3.141	0.175	7.660
I	Merged Generations (M)	2	0.480	0.798	0.527	3.892
J	Migration Rate	2	2.182	3.628	0.158	17.696
Error		3	0.902			7.316
Sum		17	12.329			100

Table 76 – ANOVA of the optimal objective S/N ratio for the DLPs in Stage Two

Factor		**ANOVA Statistics**				
Symbol	Description	DOF	SS	F-ratio	P-value	Pct. Contr.
A	Population Size (1 | 3)	2	11.571	45.483	0.006	45.625
B	Population Size (2 | M)	2	3.370	13.246	0.032	13.288
C	Pct. Jmp. Gene (1 | 2 | 3 | M)	1	0.735	5.781	0.096	2.900
D	Pct. Crossover (1 | 2 | 3)	1	0.074	0.581	0.501	0.291
E	Pct. Crossover (M)	1	0.059	0.467	0.544	0.234
F	Pct. Mutation (1 | 2 | 3)	1	0.031	0.241	0.657	0.121
G	Pct. Mutation (M)	1	0.065	0.512	0.526	0.257
H	Iso. Generations (1 | 2 | 3)	1	6.861	53.934	0.005	27.051
I	Merged Generations (M)	2	1.366	5.370	0.102	5.387
J	Migration Rate	2	0.847	3.331	0.173	3.341
Error		3	0.382			1.505
Sum		17	25.361			100

G.7 Experiment VI

G.7.1 Initial Solution Observations

After an initial testing of the concepts had been performed, a few important observations, which hadn't been observed before, were made regarding the performance of the developed FSPPM. Though the method performed well under the conditions it was tested for in the *52 Problem Test Set*, its limitations became more noticeable while tackling the complex case layouts of the case study, which had more diverse layout characteristics. With some constrained objects in the space (e.g. pillars) having significantly smaller footprints relative to the other objects in the space and moreover a sparsely populated, high aspect ratio, high white space layout considered, led to the method performing less than ideally when compared to before. Despite this, it remained more effective than the baseline random method of the literature, which as demonstrated before, would not be capable of solving such a complex and sized problem to begin with.

After testing the solution of the scenario problems in Stage One of the solution approach and observing clearly non ideal layouts being generated, such as those demonstrated below in Figure 88 for Concept 2A, it was concluded that a future extension of this work should focus on further developing the FSPPM. Such efforts should attempt to account for such sparse, high white space layout characteristics, constrained objects with small areas relative to the other objects and the space itself, along with the general positioning of the constrained objects relative not only to the diagonal line, but also their vicinity to the boundaries. As can be observed, due to the current nature of the developed FSPPM, designs generated, such as those presented in

Figure 88, often placed even fixed stations incorrectly (e.g. design on the left where the receiving station was actually placed above the first pillar in the space). Moreover, due to the placement of the pillars and largely sparse space, stations were often placed well away from the input/output doors of the space, which is less than ideal from a handling perspective. This result is due to the station objects needing to be placed between or after these constrained pillar positions in the sequence-pairs. This is why in both designs provided, both an inspection and shipping station are placed above the upper most pillar. This pillar, based on its diagonal distance, required placement in the sequence two from the end thereby requiring two such stations to then be placed after it and therefore above it in the physical space.

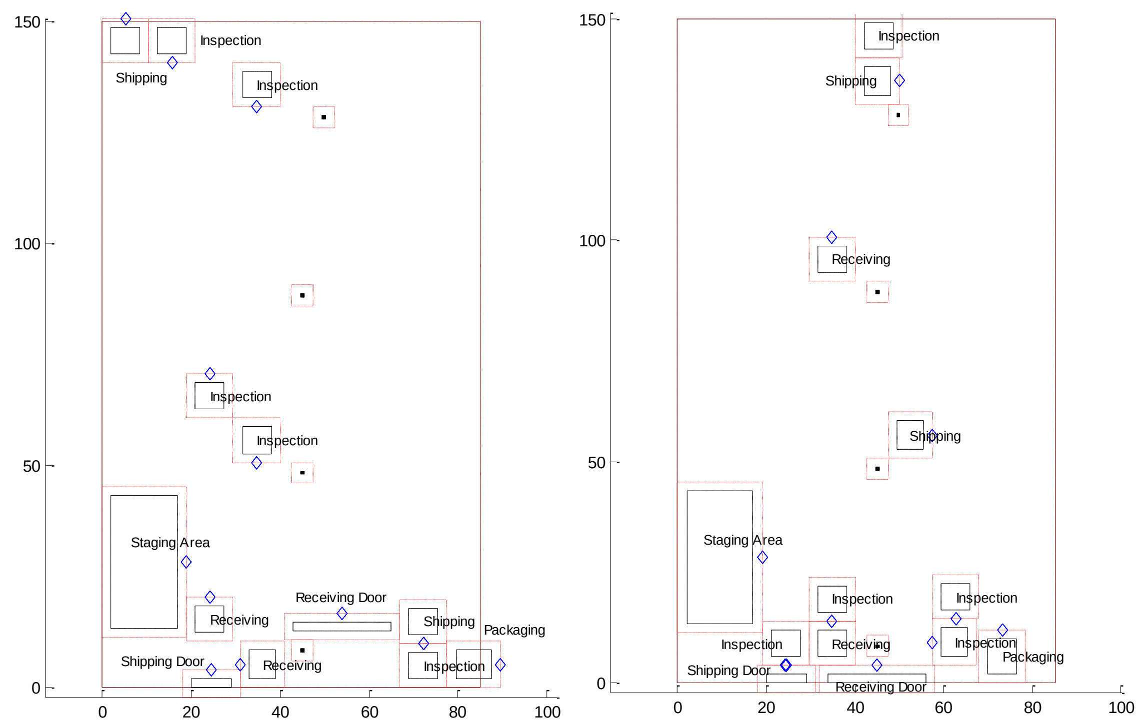


Figure 88 – Inferior layout designs generated for Concept 2A

In light of these observations, it is also believed that the method for defining the expected position of placement in the sequences could additionally be made a function of the characteristics noted before (white space, aspect ratio, relative size, position relative to the boundaries); that is, in addition to its position relative to the diagonal bisecting lines as it is currently defined in the FSPPM deployed. It is also believed that a better approach to defining the sigma value would be to define it as a function of such characteristics as well and moreover, have it defined individually for each constrained object in the layout.

The following examples of dependency were formed following a testing of how the positions of the constrained and unconstrained objects in Concept 2A and 2B were placed for different combinations of mean and sigma values defined on an individual constrained object-basis. First, constrained objects falling in the middle of the space (i.e. near the bisecting diagonals) should in general have a higher sigma value and moreover, such objects having small relative sizes should have yet a higher sigma property. This is because as small as their footprint is, they can easily be inserted into small openings between objects. As such, they need be less rigidly defined within the sequence-pairs. In fact, it could be beneficial to place such small objects, such as that of the pillars in this example, randomly. This could be achieved by having the sigma value be a function of its relative size, whereby it could become so large that it effectively has equal probability of placement across a larger range of positions in the sequence.

Second, objects falling near the boundaries of the space should also have lower sigma values and additionally, their placements should be pushed towards the end of the sequences. In retrospect, this makes sense. It is a result of how the step lines are formed, which map a physical layout to a sequence-pair. In the provided Concept 2B example in Figure 89, the receiving PO door, though falling near the negative bisecting diagonal line, was actually found, due to its vicinity to the lower boundary, to fall within roughly the fourth position in the negative sequence, coming after that of the rack door, receiving SO door, and shipping SO door (in that order). As such, it is believed that if the placement algorithm was modified to establish the position of the object relative to the ratio between its normal distance to the bisecting line (like before) to that of the normal distance to the closest boundary (rather than the maximum corner distance), better placement could then be achieved. In Figure 89, this would be from the negative bisecting diagonal line to the lower boundary point whose normal line passes through the centroid of the receiving PO door. Continuing with the negative sequence and the provided example in Figure 89, these objects falling on the left and bottom boundaries should then be ordered accordingly. Objects on the left boundary and at the top should be placed at the very beginning, then working inward in the negative sequence the order should be those then on the left boundary, then those in the bottom-left corner, then finally those along the bottom edge before finally that of the receiving PO door near the right end of the bottom boundary. This ordering is demonstrated by the arrow direction in Figure 89, whereby the source of the arrow coincides with the start of the negative sequence. This ordering aligns with the fundamental construct of the SP model and further aligns with observations made while investigating the placement in each of the various concept setups. Moreover, it was observed that this ordering always produced a design where these constrained door stations fell appropriately when mapped to the physical layout. By extension, the same logic can be applied for the other end of the negative sequence as well as the ends of the positive sequence.

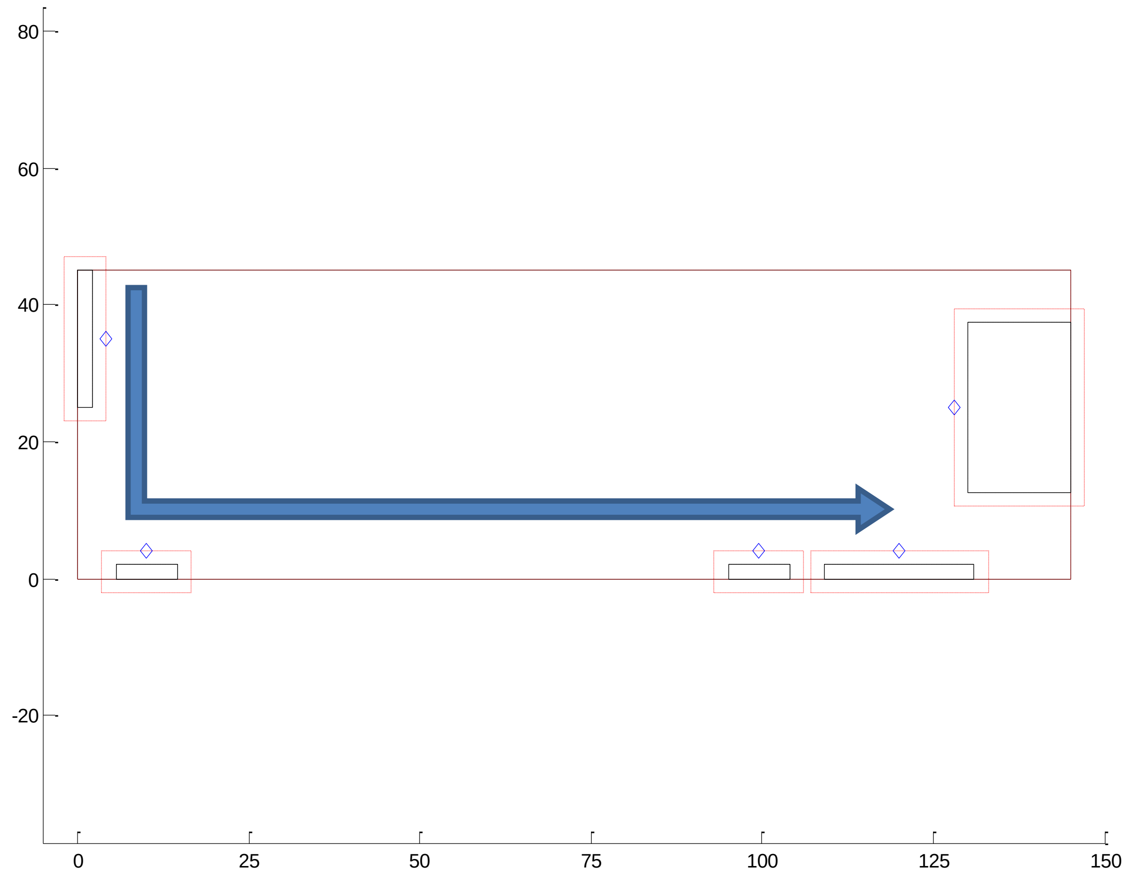


Figure 89 – Concept 2B placement ordering in the negative sequence

Third, when handling a layout with a high white space, those constrained objects falling in the upper-right of the space should be pushed outward towards the end of the negative sequence in order to facilitate the placement of the other objects between it and the bottom-left corner of the space. The absence of such a dependency in the current approach is what led to the poor layout configurations observed in Figure 88 earlier. In those examples, the furthest top pillar, being not that far from the negative bisecting diagonal line, was placed by the FSPPM well from the end of the negative sequence. This effectively forces some of the movable objects to then be placed right or above it, thereby leading to an expanded layout configuration, as demonstrated in Figure 88. Implementing a white space dependency, coupled with the relative size of the movable objects and moreover, the position of the constrained object (e.g. being in the top right), would effectively push this constrained pillar towards the end of the negative sequence. This would then allow those objects falling to the right and top to then fall to left and below it, which would produce a more compact and ideal design configuration. The following forward-looking hypothesis is made in light of these observations:

> *If the FSPPM is further developed to encapsulate an algorithm defining the expected position and sigma values of the constrained objects on an individual-basis and moreover, encapsulating the above acknowledged dependencies, then better placement performance by the FSPPM would be observed under such layout characteristics as experienced in this layout problem.*

As a result of this performance, it was decided that the pillar objects in the space would be neglected. Given their size relative to the other objects, this is a reasonable simplification. Moreover, after examining the placements of the constrained objects in each concept, sets of sigma and mean values appropriate for each of the constrained objects in each of the concepts were further defined and enforced in the FSPPM. Their definitions are provided in Table 77, Table 78, and Table 79. N in the tables represents the total number of objects in the space. This action was taken to facilitate a better performing algorithm in the absence of the forward looking more sophisticated FSPPM proposed.

Table 77 – Concept 1 constrained station sigma and mean sequence-position placements

`	Property	Enforced Values			
		Staging	Receiving	Shipping	Rack
Negative	Sigma	0.1	0.1	0.1	0.1
	Mean	2	4	3	1
Positive	Sigma	0.2	0.5	0.7	1.0
	Mean	1	N-1	N-4	8

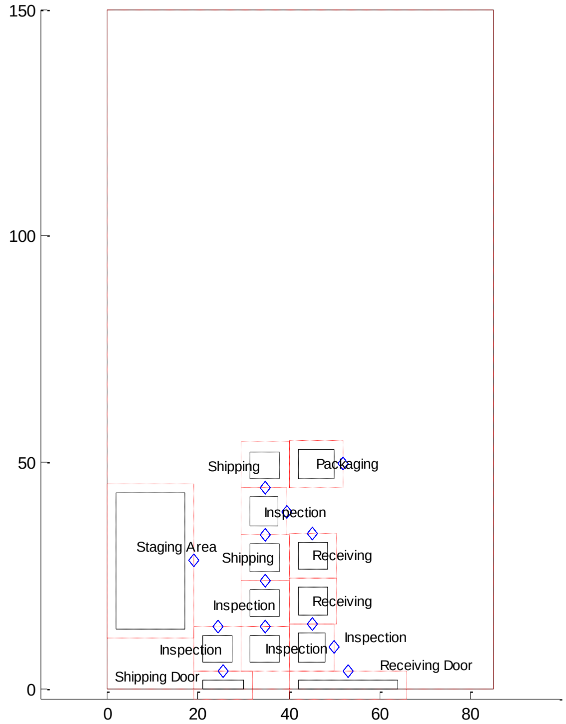
Table 78 – Concept 2A constrained station sigma and mean sequence-position placements

Sequence	Property	Enforced Values		
		Staging	Receiving	Shipping
Negative	Sigma	0.1	0.1	0.1
	Mean	1	3	2
Positive	Sigma	0.1	0.2	0.2
	Mean	1	N	N-1

Table 79 – Concept 2B constrained station sigma and mean sequence-position placements

Sequence	Property	Enforced Values				
		Staging	PO Rec.	Shipping	Rack	SO Rec.
Negative	Sigma	0.1	0.2	0.2	0.2	0.1
	Mean	N	4	3	2	1
Positive	Sigma	0.3	0.1	0.5	0.2	0.2
	Mean	1	N-1	N-4	8	8

After enforcing the sigma and mean sequence-position placements provided in the above tables, the FSPPM performed significantly better. This is demonstrated by the far more optimally placed stations and layout designs then yielded by the algorithm and illustrated in Figure 90 below. Moreover, the FSPPM generated designs with the constrained stations nearly always placed appropriately in the space.



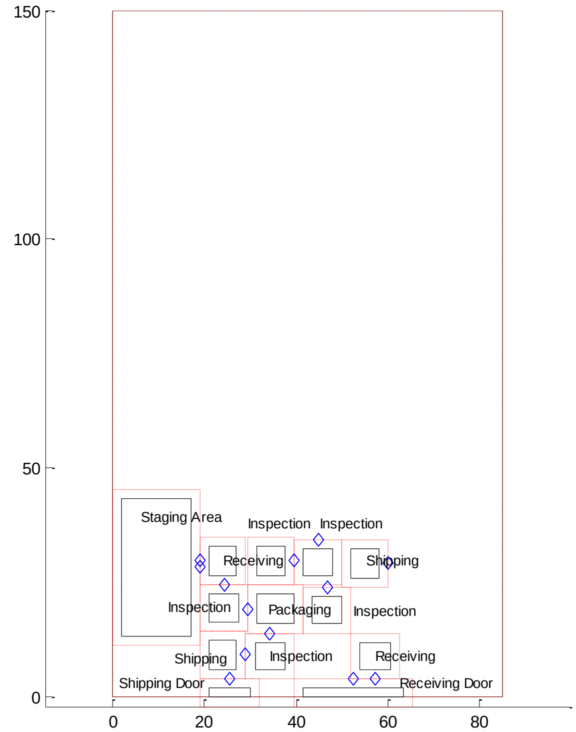


Figure 90 – More effectively placed stations by the FSPPM

G.7.2 Assumptions and Input Condition Definitions for the Scenarios

1) Walking spacing about all objects was assumed to be 2ft to enable forklifts to have sufficient space at 4ft combined to pass between stations safely

2) Parts per day (PPD) is assumed to be composed of 95% human handled and 5% forklift handled

3) ROII parts constitute a constant 1.5% of PPD, ROIS 1.5%, and ROO the remaining 2% of the allotted 5% not attributed to the PO and SO processes

4) Indirect production costs were assumed to be 30% of the direct costs

5) Direct consumable costs (DCC) were assumed to be $0.75/part human handled and $1.25/part when forklift handled

6) Setups are neglected (reflected in the setup rates and capacities defined below as zero)

7) Number of handlers per part is assumed to be one

8) Budget was set to 100% of the previous period's net income

9) Capital costs were considered to be zero provided the system had all the necessary stations at its disposal already

10) Assumed cost to install receiving, inspection, shipping, and packaging stations were $7/hr while the staging areas and fixed stations don't matter so were set to $0/hr

11) Assumed cost move receiving, inspection, shipping, and packaging stations was $2.5/ft while the staging areas and fixed stations don't matter so were set to $0/ft

12) Assumed cost to reroute supporting items for receiving, inspection, shipping, and packaging stations was $10/ft to account for network cables and electrical

conduit required to run the computers at the stations while the others were assumed to be $0/ft

13) Staging area capacities were set to a large value so as to avoid such stations from becoming the bottleneck of the system when they only act as a holding place for parts

Table 80 – Case study station input data

Stations:	Type:	Width (ft)	Height (ft)	Depth - 3D (ft)	Spacing (ft)	Manning
Receiving Station	WORKSTATION	6.5	6	3	0	1
Inspection Station	WORKSTATION	6.5	6	3	0	1
Shipping Station	WORKSTATION	6.5	6	3	0	1
Packaging Station	WORKSTATION	6.5	8	3	0	1
Scan Station	WORKSTATION	3	2.5	3	0	1
Staging Area PO	STAGING	30	15	3	0	0
PO Receiving Door	DOOR	22	2	3	0	0
SO Receiving Door	DOOR	9	2	3	0	0
Shipping Door	DOOR	9	2	3	0	0
Building A Door	DOOR	20	2	3	0	0
Building C Door	DOOR	20	2	3	0	0
Staging Area SO	STAGING	25	15	3	0	0

Stations:	Type:	I/O Xoffset (ft)	I/O Yoffset (ft)	Install Time (hr)	Uninstall Time (hr)	Move Rate (ft/hr)
Receiving Station	WORKSTATION	0	3	0.5	0.2	5280
Inspection Station	WORKSTATION	0	3	0.5	0.2	5280
Shipping Station	WORKSTATION	0	3	0.5	0.2	5280
Packaging Station	WORKSTATION	0	4	0.5	0.2	5280
Scan Station	WORKSTATION	0	1.25	0.5	0.2	5280
Staging Area PO	STAGING	0	7.5	0.1	0.1	16368
PO Receiving Door	DOOR	0	1	0	0	0
SO Receiving Door	DOOR	0	1	0	0	0
Shipping Door	DOOR	0	1	0	0	0
Building A Door	DOOR	0	1	0	0	0
Building C Door	DOOR	0	1	0	0	0
Staging Area SO	STAGING	0	7.5	0.1	0.1	16368

Table 81 – Case study region input data

Regions:	Width (ft)	Height (ft)	Depth - 3D (ft)	Spacing (ft)
Racks	60	215	10	0
Office Space (Side)	40	160	10	0
Office Space (Front)	55	40	10	0
Building Pillar	0.75	0.75	10	0

Table 82 – Case study personnel data

Personnel ID	Labor Rate ($/hr)	Unit
Receiver	$16.75	Receiving Station
Shipper	$16.75	Shipping Station
Inspector	$24.00	Inspection Station
Packager	$16.75	Packaging Station
Handler	$14.00	Handling

Table 83 – Case study horizon-based discrete condition inputs

Condition	Value	Unit	Notes
Labor Factor	1	[]	Constant across the planning horizon
Work Day	5	days/week	Constant across the planning horizon
Work Hours	8	hours/day	Constant across the planning horizon

Table 84 – Case study horizon-product-based linear condition inputs

Condition	Value	Units
Setup Rate	0	units / setup
Market Value	75	$ / unit
Estimated Manufacturing Cost	10	$ / unit

Table 85 – Case study process-based condition inputs

Condition	Value	Units
Setup Capacity	0	setups / hour
Handler Flow-Rate Capacity	2	unit mph
Forklift Flow-Rate Capacity	2.8	unit mph
Handler Labor Rate	14	$ / hour
Other Handling Costs (Handlers)	0	$ / ft unit
Other Handling Costs (Forklift)	0.1	$ / ft unit

Table 86 – Case study station capacities

Station Capacity	Value	Units
Inspection	5	units/hr
Receiving	1.5	units/hr
Shipping	6	units/hr
Packaging	20	units/hr
Staging	6000	units/hr

Table 87 – Restructuring schedule options

Option	Periods	Restructuring Schedule
1	1	M0
2	2	M0, M12
3	2	M0, M18
4	3	M0, M12, M18
5	3	M0, M12, M24

Table 88 – Manned station decomposition option definition for restructuring option one

Option	**Period 1**		
	Inspection	Receiving	Shipping
1	10 (5,5)	4 (2,2)	4 (2,2)
2	12 (5,7)	4 (2,2)	4 (2,2)
3	10 (4,6)	4 (2,2)	4 (2,2)
4	12 (7,5)	4 (2,2)	4 (2,2)

Table 89 – Manned station decomposition option definition for restructuring options two and three

Option	**Period 1**			**Period 2**		
	Inspection	Receiving	Shipping	Inspection	Receiving	Shipping
1	8 (4,4)	2 (1,1)	2 (1,1)	12 (6,6)	4 (2,2)	4 (2,2)
2	10 (5,5)	2 (1,1)	2 (1,1)	12 (6,6)	4 (2,2)	4 (2,2)
3	8 (4,4)	2 (1,1)	2 (1,1)	12 (4,8)	4 (1,3)	4 (1,3)
4	10 (5,5)	2 (1,1)	2 (1,1)	12 (5,7)	4 (2,2)	4 (2,2)

Table 90 – Manned station decomposition option definition for restructuring options four and five

Option	Period 1			Period 2			Period 3		
	Inspection	Receiving	Shipping	Inspection	Receiving	Shipping	Inspection	Receiving	Shipping
1	8 (4,4)	2 (1,1)	2 (1,1)	10 (5,5)	2 (1,1)	2 (1,1)	12 (6,6)	4 (2,2)	4 (2,2)
2	8 (4,4)	2 (1,1)	2 (1,1)	12 (6,6)	4 (2,2)	4 (2,2)	12 (6,6)	4 (2,2)	4 (2,2)
3	8 (4,4)	2 (1,1)	2 (1,1)	10 (4,6)	2 (1,1)	2 (1,1)	12 (4,8)	4 (1,3)	4 (1,3)
4	8 (4,4)	2 (1,1)	2 (1,1)	11 (5,6)	4 (2,2)	4 (2,2)	12 (5,7)	4 (2,2)	4 (2,2)

In the above tables, the 10 (5,5) form indicates the total number of stations then the assignment of these across the two process groups as follows (PO/ROIS stations, SO/ROO/ROII stations). Some of these manned station decomposition options considered, start with the current operational structure (i.e. number of active stations / workers) while others consider hiring at the start. As can be observed, several assignment distributions are considered and for the most part across the different restructuring forms, these distributions are maintained for the option levels.

Table 91 – Handler options considered

Option	Period Structure		
	Single	Double	Triple
1	(0.25,)	(0.25, 0.25)	(0.25, 0.25, 0.5)
2	(0.5,)	(0.25, 0.5)	(0.25, 0.375, 0.5)

These handler quantities, being less than one, indicate a portion of a handler's time. After studying the current configuration, it was identified that, based on the standard 2mph human movement rate that was assumed, a full handlers time each day would be excessive regardless of what PPD option was considered. Since it was desired for handler constrained cases to be observed on occasion for demonstration purposes, these were adjusted to below a single worker to effectively force such situations to arise on occasion. Note also that for Concept 2, these handler ratios are evenly split across the two concept layouts provided that the processes were segregated in this considered concept.

Table 92 – PPD distribution options considered

Option	**Process**	**Planning Horizon**			
		M0	M12	M24	M36
1	PO	60%	60%	60%	60%
	SO	35%	35%	35%	35%
2	PO	60%	55%	50%	45%
	SO	35%	40%	45%	50%
3	PO	60%	45%	25%	25%
	SO	35%	50%	70%	70%

G.7.3 Case Study Supplementary Results

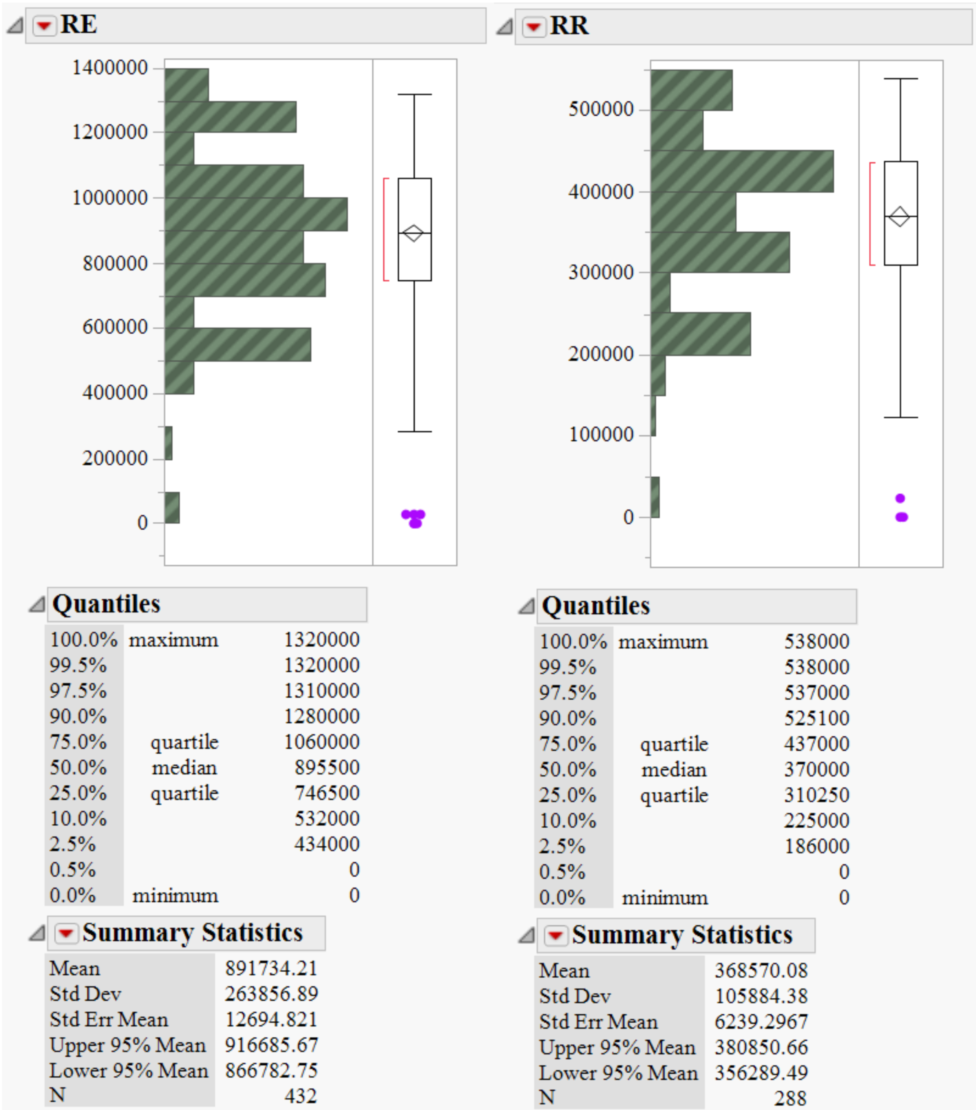


Figure 91 – Distribution of RE and RR for Concept 0

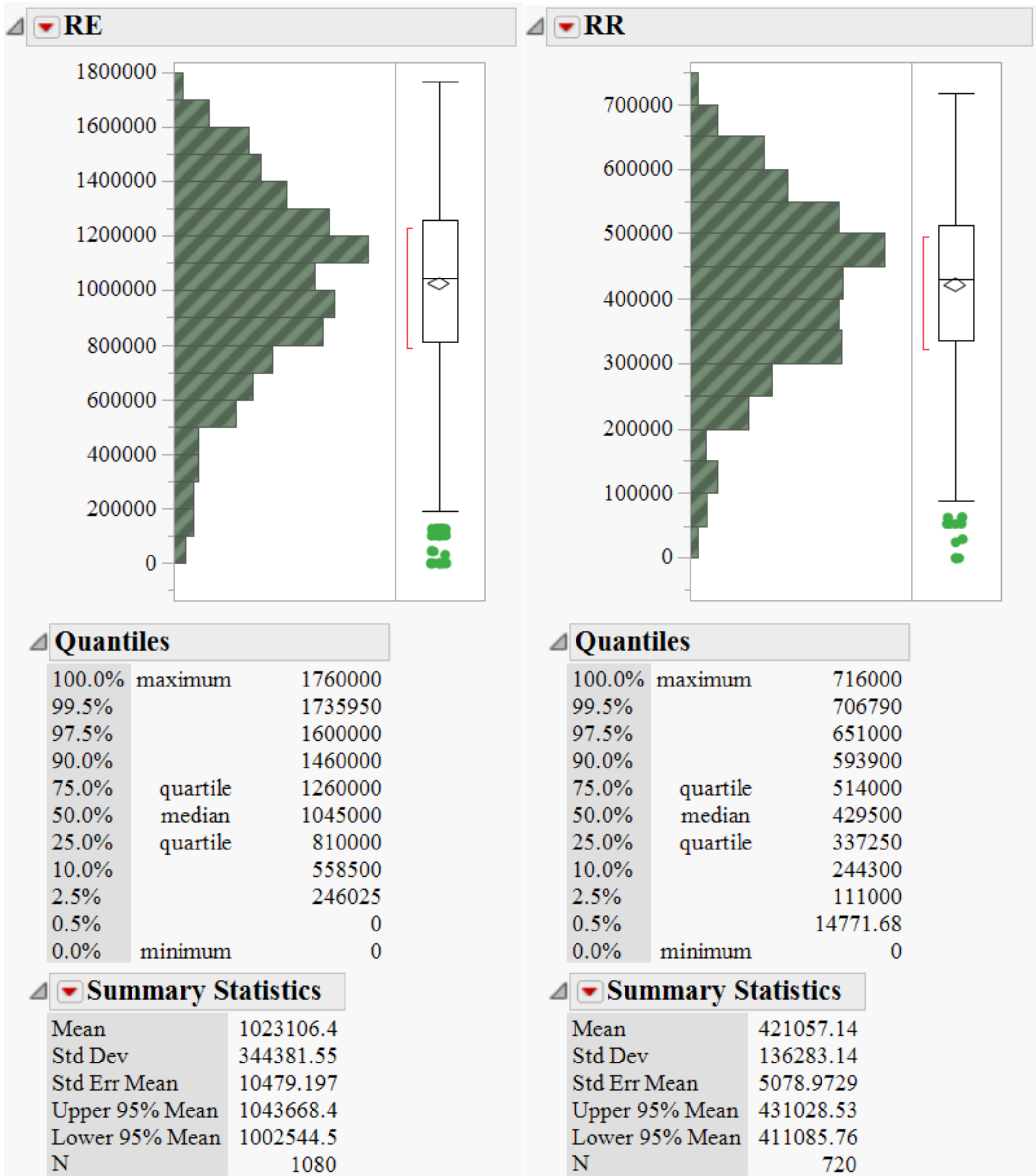
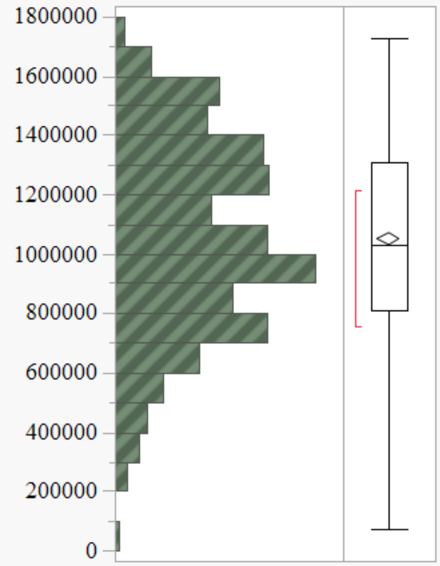


Figure 92 – Distribution of RE and RR for Concept 1

RE



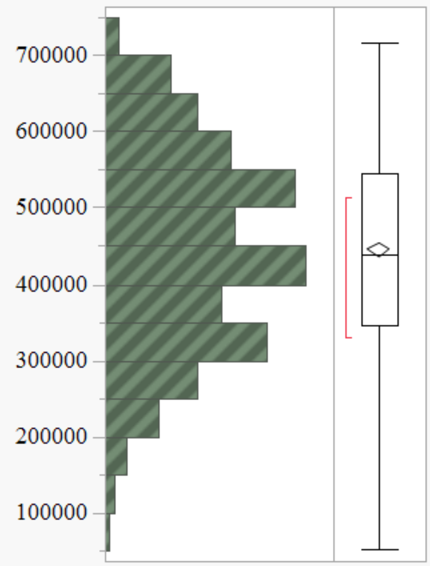
Quantiles

100.0%	maximum	1728000
99.5%		1711595
97.5%		1620950
90.0%		1501000
75.0%	quartile	1306000
50.0%	median	1031500
25.0%	quartile	807500
10.0%		645200
2.5%		376719.475
0.5%		208376.225
0.0%	minimum	72839

Summary Statistics

Mean	1051676.8
Std Dev	326330.93
Std Err Mean	9929.934
Upper 95% Mean	1071161
Lower 95% Mean	1032192.7
N	1080

RR



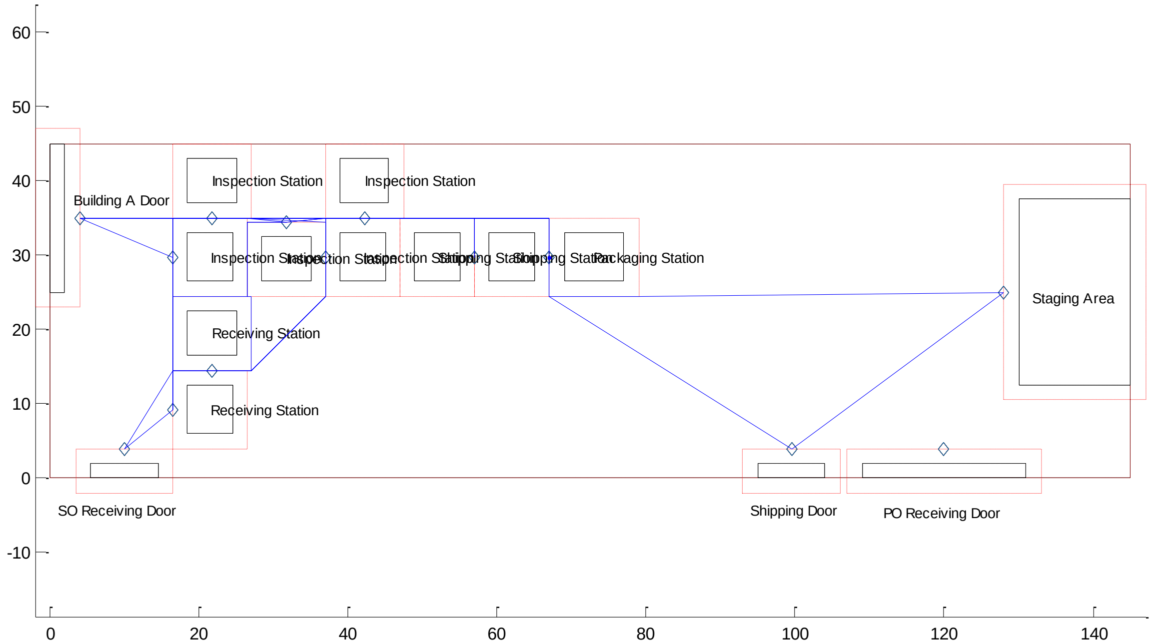
Quantiles

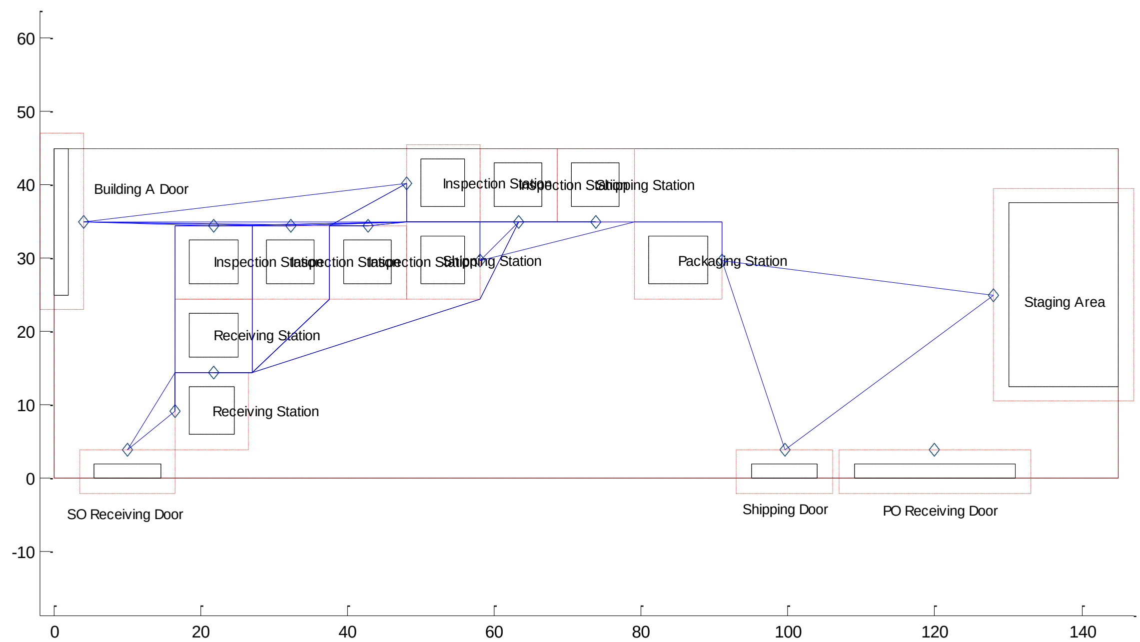
100.0%	maximum	715000
99.5%		709000
97.5%		671975
90.0%		624900
75.0%	quartile	546000
50.0%	median	438218.5
25.0%	quartile	347014.5
10.0%		282044.2
2.5%		174631
0.5%		106752.68
0.0%	minimum	53625.4

Summary Statistics

Mean	444745.82
Std Dev	130785.73
Std Err Mean	4874.0964
Upper 95% Mean	454314.99
Lower 95% Mean	435176.66
N	720

Figure 93 – Distribution of RE and RR for Concept 2


60
50
40
30
20
10
0
-10
0
20
40
60
80
100
120
140
Building A Door
Inspection Station
Inspection Station
Packaging Station
Staging Area
Receiving Station
Receiving Station
SO Receiving Door
Shipping Door
PO Receiving Door


60
50
40
30
20
10
0
-10
0
20
40
60
80
100
120
140
Building A Door
Shipping Station
Packaging Station
Staging Area
Receiving Station
Receiving Station
SO Receiving Door
Shipping Door
PO Receiving Door

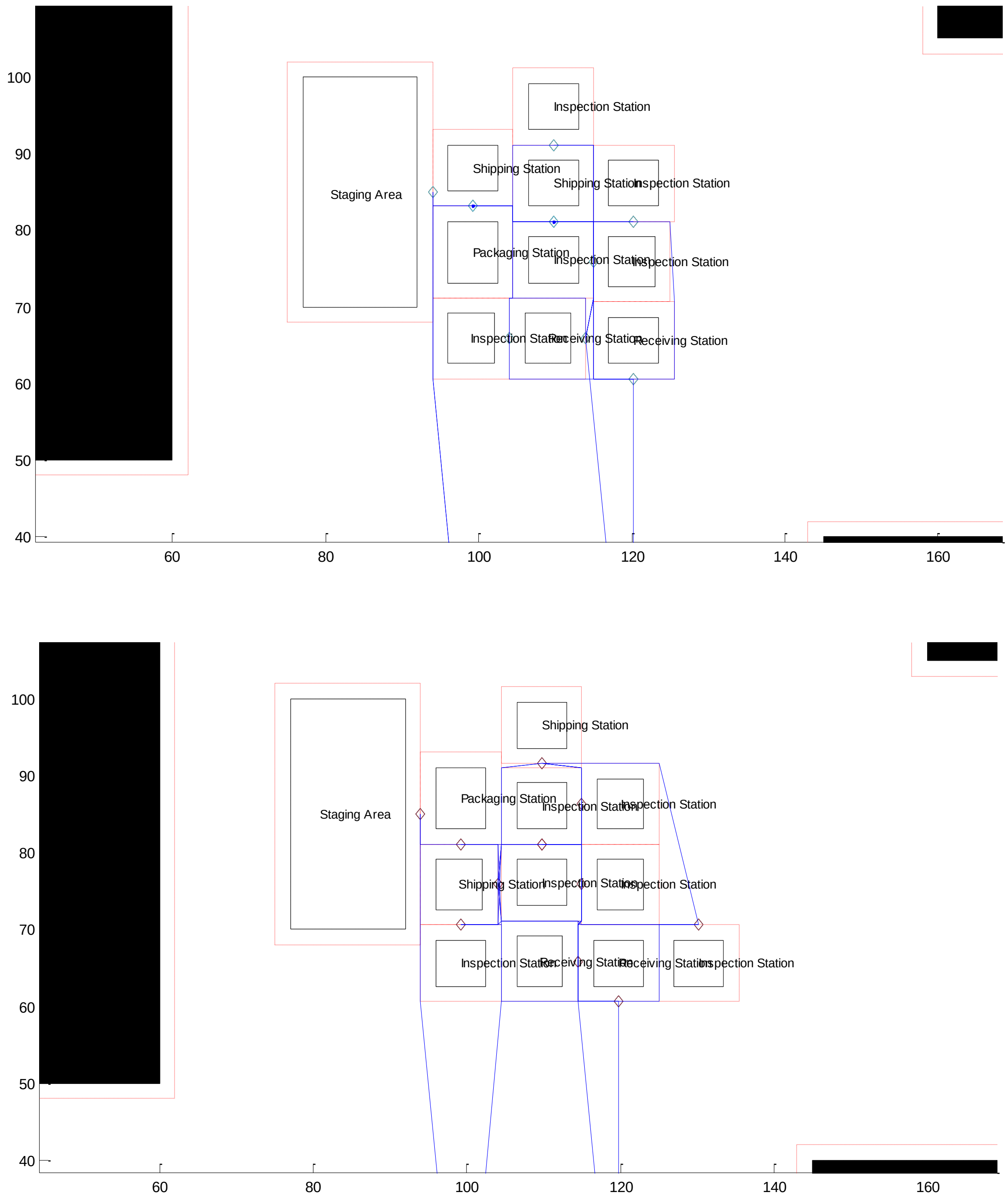


Figure 94 – Alternative layout designs

REFERENCES

[1] "Lean Manufacturing and the Environment." https://nepis.epa.gov/Exe/ZyPURL.cgi?Dockey=P1001XP2.TXT: Environmental Protection Agency.

[2] "Lean Thinking and Methods | Lean Manufacturing and the Environment | Us Epa." http://www.epa.gov/lean/environment/methods/index.htm: Environmental Protection Agency, 09 June 2014.

[3] "Price Cost Analytics." *PRICE Systems*. Web.

[4] "Welcome to Kapes.Com." *Welcome to KAPES.com*. Web.

[5] Abdi, Hervé. "Coefficient of Variation." *Encyclopedia of Research Design*. Ed. Salkind, N. J. Thousand Oaks, CA: SAGE Publications Inc., 2010. 169-71.

[6] Afentakis, Panos, Robert A. Millen, and Marius M. Solomon. "Dynamic Layout Strategies for Flexible Manufacturing Systems." *International Journal of Production Research* 28.2 (1990).

[7] Albert, Shon. "Solving Mixed Integer Linear Programs Using Branch and Cut Algorithm." Dissertation. North Carolina State University, 2006.

[8] Allen, Arnold A. *Probability, Statistics, and Queueing Theory: With Computer Science Applications*. Gulf Professional Publishing, 1990. Print.

[9] Armour, Gordon C., and Elwood S. Buffa. "A Heuristic Algorithm and Simulation Approach to Relative Location of Facilities." *Management Science* 9.2 (1963).

[10] Ashby, Michael F., et al. "Chapter 3.1: Processes and Process Selection." *Engineering Materials and Processes Desk Reference*. 1st ed: Butterworth-Heinemann, 2009. 89-91. Print.

[11] Azimi, Parham, and Hamid Reza Charmchi. "A New Optimization Via Simulation Approach for Dynamic Facility Layout Problem with Budget Constraints." *Modelling and Simulation in Engineering* (2012).

[12] Balakrishnan, Jaydeep, and Chun Hung Cheng. "Dynamic Layout Algorithms: A State-of-the-Art Survey." *Omega* 26.4 (1998).

[13] ---. "Genetic Search and the Dynamic Layout Problem." *Computers & Operations Research* 27.6 (2000).

[14] Balakrishnan, Jaydeep, Chun Hung Cheng, and Daniel G. Conway. "An Improved Pair-Wise Exchange Heuristic for the Dynamic Plant Layout Problem." *International Journal of Production Research* 38.13 (2000).

[15] Balakrishnan, Jaydeep, et al. "A Hybrid Genetic Algorithm for the Dynamic Plant Layout Problem." *International Journal of Production Economics* 86.2 (2003).

[16] Balakrishnan, Jaydeep, and Cun Hung Cheng. "The Dynamic Plant Layout Problem: Incorporating Rolling Horizons and Forecast Uncertainty." *Omega* 37.1 (2009).

[17] Balakrishnan, Jaydeep, Robert F. Jacobs, and Munirpallam A. Venkataramanan. "Solutions for the Constrained Dynamic Facility Layout Problem." *European Journal of Operational Research* 57.2 (1992).

[18] Balasubramani, Kamalam, and Karnan Marcus. "A Study on Flower Pollination Algorithm and Its Applications." *International Journal of Application or Innovation in Engineering & Management* 3 (2014).

[19] Ballou, Ronald H. "Dynamic Warehouse Location Analysis." *Journal of Marketing Research* 5.3 (1968).

[20] Barbosa-Póvoa, Ana Paula, Ricardo Mateus, and Augusto Q. Novais. "Optimal Facility Layout - a Generic Approach to Process Plants." *Proceedings of the Manufacturing and Production & Control and Logistics* (2000).

[21] ---. "Optimal Two-Dimensional Layout of Industrial Facilities." *International Journal of Production Research* 39.12 (2001).

[22] Baykasoglu, Adil, Turkay Dereli, and Ibrahim Sabuncu. "An Ant Colony Algorithm for Solving Budget Constrained and Unconstrained Dynamic Facility Layout Problems." *Omega* 34.4 (2006).

[23] Baykasoglu, Adil, and Nabil N.Z. Gindy. "A Simulated Annealing Algorithm for Dynamic Layout Problem." *Computers & Operations Research* 28.14 (2001).

[24] Becker, Jörg, and Martin Prischmann. "Konstruktionsbegleitendekalkulation Mit Neuronalen Netzen." *Kostenrechnungspraxis* 38.3 (1994).

[25] ---. *Veskonn: Prototypische Umsetzung Eines Modularen Konzepts Zur Konstruktionsunterstützung Mit Konnektionistischen Methoden*. Inst. für Wirtschaftsinformatik, 1993. Print.

[26] Benjaafar, Saif, Sunderesh S. Heragu, and Shahrukh A. Irani. "Next Generation Factory Layouts: Research Challenges and Recent Progress." *Interfaces* 32.

[27] Benjaafar, Saifallah. "Modeling and Analysis of Congestion in the Design of Facility Layouts." *Management Science* 48.5 (2002).

[28] Bennage, W. A., and A. K. Dhingra. "Single and Multiobjective Structural Optimization in Discrete-Continuous Variables Using Simulated Annealing." *International Journal for Numerical Methods in Engineering* 38.16 (1995).

[29] Bertsimas, Dimitris, and John Tsitsiklis. "Simulated Annealing." *Statistical Science* 8.1 (1993).

[30] Bhowmik, Rekha. "Building Design Optimization Using Sequential Linear Programming." *JCP Journal of Computers* 3.4 (2008).

[31] Bixby, Robert E. "A Brief History of Linear and Mixed-Integer Programming Computation Bixby." *Documenta Mathematica* (2012).

[32] ---. "Solving Real-World Linear Programs: A Decade and More of Progress." *Operations Research* 50.1 (2002).

[33] Bode, Jürgen. "Decision Support with Neural Networks in the Management of Research and Development: Concepts and Application to Cost Estimation." *Information & Management* 34.1 (1998).

[34] ---. "Neural Networks for Cost Estimation: Simulations and Pilot Application." *International Journal of Production Research* 38.6 (2000).

[35] Boehm, Barry, Chris Abts, and Sunita Chulani. "Software Development Cost Estimation Approaches — a Survey." *Annals of Software Engineering* 10.1 (2000).

[36] Braglia, Marcello, Simone Zanoni, and Lucio Zavanella. "Layout Design in Dynamic Environments: Strategies and Quantitative Indices." *International Journal of Production Research* 41.5 (2003).

[37] Brook, Anthony, David Kendrick, and Alexander Merraus. "Gams: A User's Guide." *ACM SIGNUM Newsletter 23* (Dec. 1988).

[38] Brusco, Michael J., and Stephanie Stahl. "Branch-and-Bound Applications in Combinatorial Data Analysis." 4 (2006).

[39] Bullington, S.F., and D.B. Webster. "Evaluating the Flexibility of Facilities Layouts Using Estimated Relayout Costs." *Proceedings of the IXth International Conference on Production Research*. 1987.

[40] Bureau, of Economic Analysis. "Top 20 Facts About Manufacturing." http://www.nam.org/Newsroom/Top-20-Facts-About-Manufacturing/: NAM 13, Nov. 2016. NAM calculations using IMPLAN.

[41] Burkard, Rainer E., et al. "The Quadratic Assignment Problem." SFB F003 - Optimierung Und Kontrolle. Graz: Universitat Graz/Technische Universitat Graz, 1998.

[42] Bussieck, Michael R., and Armin Pruessner. "Mixed-Integer Nonlinear Programming " *SIAG/OPT Newsletter: Views & News* 14.1 (2003).

[43] Case study interviews by EPA with Goodrich Aerostructures Group representatives on October 3, 2002, and “Aerospace Industry Mimics Toyota,” Financial Post, Canada (March 10, 1999).

[44] Cerny, V. "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm." *Journal of Optimization Theory and Applications* 45.1 (1985).

[45] Chan, K. C., and H. Tansri. "A Study of Genetic Crossover Operations on the Facilities Layout Problem." *Computers & Industrial Engineering* 26.3 (1994).

[46] Chang, Pei-Chann, and Shih-Hsin Chen. "The Development of a Sub-Population Genetic Algorithm Ii (Spga Ii) for Multi-Objective Combinatorial Problems." *Applied Soft Computing* 9.1 (2009).

[47] Chen, Tung-Chieh, and Yao-Wen Chang. "Modern Floorplanning Based on B*-Tree and Fast Simulated Annealing." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 25.4 (2006).

[48] Conway, Daniel G., and Munirpallam A. Venkataramanan. "Genetic Search and the Dynamic Facility Layout Problem " *Computers & Operations Research* 21.8 (1994).

[49] Cove, Identity. "Data Is Knowledge and Knowledge Is Power." A Medium Corporation, 27 Nov. 2017, medium.com/coveidentity/data-is-knowledge-and-knowledge-is-power-9895093b1090. Web.

[50] Dan, Galorath. SEER for Hardware, Electronics and Systems. Computer software. Galorath, Inc.

[51] ---. "Cost Estimation Software for Planning & Tracking Complex Projects – Seer by Galorath." Galorath, Inc. Web.

[52] Darwin, Charles. "On the Origin of Species by Means of Natural Selection or the Preservation of Favored Races in the Struggle for Life." Murray, London 1859.

[53] Dawar, Niraj, and Tony Frost. "Competing with Giants: Survival Strategies for Local Companies in Emerging Markets." Harvard Business Review, 31 July 2014. 05 Nov. 2016. <https://hbr.org/1999/03/competing-with-giants-survival-strategies-for-local-companies-in-emerging-markets>. March-April 1999 Issue.

[54] De Castro, Leandro Nunes, and Fernando J. Von Zuben. "Artificial Immune Systems: Part I–Basic Theory and Applications." *Universidade Estadual de Campinas*. Technical Report 210 (1999).

[55] ---. "The Clonal Selection Algorithm with Engineering Applications." *Proceedings of GECCO* (2000).

[56] Diaz-Gomez, Pedro A., and Dean F. Hougen. "Empirical Study: Initial Population Diversity and Genetic Algorithm Performance." *Artificial Intelligence and Pattern Recognition 2007* (2007).

[57] Dong, Ming, Chang Wu, and Forest Hou. "Shortest Path Based Simulated Annealing Algorithm for Dynamic Facility Layout Problem under Dynamic Business Environment." *Expert Systems with Applications* 36.8 (2009).

[58] Dreo, J., et al. *Evolutionary Algorithms*. Metaheuristics for Hard Optimization Simulated Annealing, Tabu Search, Evolutionary and Genetic Algorithms, Ant Colonies,...: Methods and Case Studies. Berlin: Springer, 2006. Print.

[59] ---. *Simulated Annealing*. Metaheuristics for Hard Optimization Simulated Annealing, Tabu Search, Evolutionary and Genetic Algorithms, Ant Colonies,...: Methods and Case Studies. Berlin: Springer, 2006. Print.

[60] Drud, Arne Stolbjerg. "Conopt - a Large-Scale Grg Code." *ORSA Journal on Computing* 6.2 (1994).

[61] Dunker, Thomas, Gunter Radons, and Engelbert Westkamper. "A Coevolutionary Algorithm for a Facility Layout Problem." *International Journal of Production Research* 41.15 (2003).

[62] ---. "Combining Evolutionary Computation and Dynamic Programming for Solving a Dynamic Facility Layout Problem." *European Journal of Operational Research* 165.1 (2005).

[63] Easton, Peter Douglas, et al. "Financial & Managerial Accounting for Mbas." 4th ed: Cambridge Business Publishers, 2015. 6-18. Print.

[64] ---. "Financial & Managerial Accounting for Mbas." 4th ed: Cambridge Business Publishers, 2015. 1-14. Print.

[65] ---. "Financial & Managerial Accounting for Mbas." 4th ed: Cambridge Business Publishers, 2015. 2-12. Print.

[66] Engin, Orphin, and Doyen Alpher. "Artificial Immune Systems and Applications in Industrial Problems." *Gazi University Journal of Science* 17.1 (2004).

[67] Erel, E., J. B. Ghosh, and J. T. Simon. "New Heuristic for the Dynamic Layout Problem." *Journal of the Operational Research Society* 54.12 (2003).

[68] Eversheim, Walter, Joern Neuhausen, and Marc Sesterhenn. "Design-to-Cost for Production Systems." *CIRP Annals-Manufacturing Technology* 47.1 (1998).

[69] Forghani, Kamran, Mohammad Mohammadi, and Vahidreza Ghezavati. "Designing Robust Layout in Cellular Manufacturing Systems with Uncertain Demands." *International Journal of Industrial Engineering Computations* 4.2 (2013).

[70] Foulds, L. R. *Graph Theory Applications*. New York: Springer-Verlag, 1992. Print.

[71] Francis, R. L., L. F. McGinnis, and J. A. White. *Facility Layout and Location: An Analytical Approach*. 2nd ed. Englewood Cliffs, NJ: Prentice-Hal, 1992. Print.

[72] Francis, R. L., and John A. White. *Facility Layout and Location: An Analytical Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1974. Print.

[73] Frech, Joachim Th. *Ein Verfahren Zur Integrierten, Prozessbegleitenden Vorkalkulation Für Die Kostengerechte Konstruktion*. Springer, 1998. Print.

[74] Fu, Michael C., and Bharat K. Kaku. "Minimizing Work-in-Process and Material Handling in the Facilities Layout Problem." *IIE Transactions* 29.1 (1997).

[75] ---. *On Queueing Network Models and the Quadratic Assignment Problem*. College Park, MD: College of Business and Management, University of Maryland, 1997.

[76] Galorath, Dan. "Product Brief: Seer for Hardware Electronics & Systems." Galorath Inc. Web. September 6, 2017.

[77] ---. "Product Brief: Seer for Manufacturing." Galorath Inc. Web. September 6, 2017.

[78] Garey, M. R., and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of Np-Completeness*. New York: W. H. Freeman & Company, 1979. Print.

[79] Goetschalckx, Marc. "An Interactive Layout Heuristic Based on Hexagonal Adjacency Graphs." *European Journal of Operational Research* 63.2 (1992).

[80] Goldberg, David E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989. Print.

[81] Grefenstette, J.J. "Optimization of Control Parameters for Genetic Algorithms." *IEEE Transactions on Systems, Man, and Cybernetics* 16.1 (1986).

[82] Gröner, Lothar. "Entwicklungsbegleitende Vorkalkulation." Heidelberg: Springer-Verlag Berlin, 1991.

[83] Grundmann, Mechtildis. *Entwicklung Eines Kalkulationsinstrumentariums Für Spritzgiessbetriebe Auf Der Basis Von Ähnlichkeitsbetrachtungen*. Verlag der Augustinus-Buchh, 1994. Print.

[84] Gupta, R.M. "Flexibility in Layouts: A Simulation Approach." *Material Flow* 3 (1986).

[85] Gutin, Gregory, and Abraham P. Punnen. "The Traveling Salesman Problem: Applications, Formulations and Variations." *The Traveling Salesman Problem and Its Variations*. Vol. 12. Boston, MA: Springer US, 2007. Web.

[86] Hassan, Mohsen MD, and Gary L. Hogg. "A Review of Graph Theory Application to the Facilities Layout Problem." *Omega* 15.4 (1987).

[87] Hicks, Christian. "A Genetic Algorithm Tool for Optimising Cellular or Functional Layouts in the Capital Goods Industry." *International Journal of Production Economics* 104.2 (2006).

[88] Hosseini-Nasab, Hasan, and Leila Emami. "A Hybrid Particle Swarm Optimisation for Dynamic Facility Layout Problem." *International Journal of Production Research* 51.14 (2013).

[89] INCONTROL. *Enterprise Dynamics 8.1*. Vers. 8.1. Computer software.

[90] Kennedy, J., and R. C. Eberhart. "Particle Swarm Optimization." *Procedings of IEEE International Conference on Neural Networks* 4 (1995).

[91] Kim, J-G, and Y-D Kim. "A Branch and Bound Algorithm for Locating Input and Output Points of Departments on the Block Layout." *Journal of the Operational Research Society* 50.5 (1999).

[92] Kiritsis, D., K-P. Neuendorf, and P. Xirouchakis. "Petri Net Techniques for Process Planning Cost Estimation." *Advances in Engineering Software* 30.6 (1999).

[93] Kiritsis, D., and P. Xirouchakis. "Deviplan: A Bidpreparation System for Mechanical Sme." *Proceedings of DECT/CIE'00, ASME* (2000).

[94] Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi. "Optimization by Simulated Annealing." *Science* 220.4598 (1983) 671-80. Web. JSTOR. 06 Nov. 2016.

[95] Konak, Abdullah, et al. "A New Mixed Integer Programming Formulation for Facility Layout Design Using Flexible Bays." *Operations Research Letters* 34.6 (2006).

[96] König, Thomas. *Konstruktionsbegleitende Kalkulation Auf Der Basis Von Ähnlichkeitsvergleichen*. Eul, 1995. Print.

[97] Koopmans, Tjalling C., and Martin Beckmann. "Assignment Problems and the Location of Economic Activities." *Econometrica* 25.1 (1957).

[98] Koste, Lori L., and Manoj K. Malhotra. "A Theoretical Framework for Analyzing the Dimensions of Manufacturing Flexibility." *Journal of Operations Management* 18 (1999).

[99] Kouvelis, Panagiotis, Abbas A. Kurawarwala, and Genaro J. Gutierrez. "Algorithms for Robust Single and Multiple Period Layout Planning for Manufacturing Systems." *European Journal of Operational Research* 63.2 (1992).

[100] Kulturel-Konak, Sadan. "Approaches to Uncertainties in Facility Layout Problems: Perspectives at the Beginning of the 21st Century." *Journal of Intelligent Manufacturing* 18.2 (2007).

[101] Kumar, S. Anil. *Operations Management*. New Age International, 2009. Print.

[102] Kümper, Ralf. *Ein Kostenmodell Zur Verursachungsgerechten Vorkalkulation*. Aachen: Shaker, 1996. Print.

[103] Kusiak, Andrew, and Sunderesh S. Heragu. "The Facility Layout Problem." *European Journal of Operational Research* 29.3 (1987).

[104] Lacksonen, T. A. "Static and Dynamic Layout Problems with Varying Areas." *The Journal of the Operational Research Society* 45.1 (1994).

[105] Lacksonen, T. A. "Preprocessing for Static and Dynamic Facility Layout Problems." *International Journal of Production Research* 35.4 (1997).

[106] Lacksonen, T. A., and E. Emory Enscore. "Quadratic Assignment Algorithms for the Dynamic Layout Problem." *International Journal of Production Research* 31.3 (1993).

[107] Layer, Alexander, et al. "Recent and Future Trends in Cost Estimation." *International Journal of Computer Integrated Manufacturing* 15.6 (2002).

[108] Lee, Kyu-Yeul, Myung-Il Roh, and Hyuk-Su Jeong. "An Improved Genetic Algorithm for Multi-Floor Facility Layout Problems Having Inner Structure Walls and Passages." *Computers & Operations Research* 32.4 (2005).

[109] Leon-Garcia, Alberto. *Probability, Statistics, and Random Processes for Electrical Engineering*. 3rd ed: Prentice Hall, 2008. Print.

[110] Lew, Art, and Holger Mauch. "Introduction to Dynamic Programming." *Dynamic Programming: A Computational Tool*. Berlin: Springer, 2007. 5.

[111] Liebers, Arthur. "An Architecture for Cost Control in Manufacturing: The Use of Cost Information in Order-Related Decisions." Universiteit Twente, The Netherlands, 1998.

[112] Lin, Lie Chien, and Gunter P. Sharp. "Quantitative and Qualitative Indices for the Plant Layout Evaluation Problem." *European Journal of Operational Research* 116 (1999).

[113] LINDO Systems Inc. *Lingo 8.0*. Vers. 8.0. Computer software.

[114] Little, John D.C. "A Proof for the Queuing Formula: L = Λw." *Operations Research* 9.3 (1961).

[115] Liu, Qi, and Russell D. Meller. "A Sequence-Pair Representation and Mip-Model-Based Heuristic for the Facility Layout Problem with Rectangular Departments." *IIE Transactions* 39.4 (2007).

[116] Malakooti, B., and G.I. D'souza. "Multiple Objective Programming for the Quadratic Assignment Problem." *International Journal of Production Research* 25.2 (1987).

[117] Man, K. F., et al. "Jumping-Genes in Evolutionary Computing." *Proceedings of the 30th Annual Conference of IEEE Industrial Electronics Society* (2004).

[118] Mazinani, Mostafa, Mostafa Abedzadeh, and Navid Mohebali. "Dynamic Facility Layout Problem Based on Flexible Bay Structure and Solving by Genetic Algorithm." *The International Journal of Advanced Manufacturing Technology* 65 (2013).

[119] McClintock, Barbara. "The Origin and Behavior of Mutable Loci in Maize." *Proceedings of the National Academy of Sciences* 36.6 (1950).

[120] McKendall, Alan R. Jr, and Artak Hakobyan. "Heuristics for the Dynamic Facility Layout Problem with Unequal-Area Departments." *European Journal of Operational Research* 201.1 (2010).

[121] McKendall, Alan R. Jr., Jin Shang, and Saravanan Kuppusamy. "Simulated Annealing Heuristics for the Dynamic Facility Layout Problem." *Computers & Operations Research* 33.8 (2006).

[122] Metropolis, Nicholas, et al. "Equation of State Calculations by Fast Computing Machines." *The Journal of Chemical Physics* 21.6 (1953).

[123] Minier, Ethan. "Optimization and Assessment of Variable Production Manufacturing Environments." Unpublished report: Georgia Institute of Technology, 2014.

[124] Montgomery, Douglas C., and George C. Runger. "Probability- Random Variables." *Applied Statistics and Probability for Engineers*. 5th ed. Hoboken, NJ: Wiley, 2011. 58. Print.

[125] Montreuil, Benoit. "A Modelling Framework for Integrating Layout Design and Flow Network Design." *Progress in Material Handling and Logistics Material Handling ‘90* (1991).

[126] Montreuil, Benoit, and Andree Laforge. "Dynamic Layout Design Given a Scenario Tree of Probable Futures." *European Journal of Operational Research* 63.2 (1992).

[127] Moslemipour, Ghorbanali, Tian Soon Lee, and Dirk Rilling. "A Review of Intelligent Approaches for Designing Dynamic and Robust Layouts in Flexible Manufacturing Systems." *The International Journal of Advanced Manufacturing Technology* 60.1 (2011).

[128] Murata, H., et al. "Rectangle-Packing-Based Module Placement." *Proceedings of IEEE International Conference on Computer Aided Design (ICCAD)* (1995).

[129] ---. "Vlsi Module Placement Based on Rectangle-Packing by the Sequence-Pair." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 15.12 (1996).

[130] Nicol, M. L., and R. H. Hollier. "Plant Layout in Practice." *Material Flow* 1.3 (1983).

[131] Nocedal, Jorge, and Stephen J. Wright. "Linear Programming: The Simplex Method." *Numerical Optimization*. 2nd ed. New York: Springer Science & Business Media, 2006. 355-91. Print.

[132] Norman, Bryan A., and Alice E. Smith. "Considering Production Uncertainty in Block Layout Design." *European Journal of Operational Research* (1999).

[133] Ott, H., and V. Hubka. "Vorausberechnung Der Herstellkosten Von Maschinenteilen Beim Entwerfen." *Schweizer Maschinenmarkt* 83.31 (1983).

[134] Ou-Yang, C., and T. S. Lin. "Developing an Integrated Framework for Feature-Based Early Manufacturing Cost Estimation." *The international journal of advanced manufacturing technology* 13.9 (1997).

[135] Parkera, Rodney P., and Andrew Wirthb. "Manufacturing Flexibility: Measures and Relationships." *European Journal of Operational Research* 118.3 (1999).

[136] Peters, Brett A., and Taho Yang. "A Combined Adjacency Graph/Integer Program Formulations for Facility Layout Problems in Flexible Manufacturing Systems." *IIE Transactions, In Review*. *Anan Mungwattana Bibliography* 112 (1995).

[137] Pham, D. T., and Dervis Karaboga. "Genetic Algorithms." *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*. London: Springer, 2000. Print.

[138] Pickel, H. *Kostenmodelle Als Hilfsmittel Zum Kostengunstigen Konstruieren*. Muchen: Hanser, 1989. Print.

[139] Pillai, Madhusudanan V., Irappa Basappa Hunagund, and Krishna K. Krishnan. "Design of Robust Layout for Dynamic Plant Layout Problems." *Computers & Industrial Engineering* 61.3 (2011).

[140] Pourvaziri, Hani, and B. Naderi. "A Hybrid Multi-Population Genetic Algorithm for the Dynamic Facility Layout Problem." *Applied Soft Computing* 24 (2014).

[141] Raman, Dhamodharan, and Sev V. Nagalingam. "Productive Area Utilisation: Towards Measuring the Effectiveness of Facilities Layout." *Proceedings of the 4th International Conference on Manufacturing Research*. 2006.

[142] Raman, Dhamodharan, Sev V. Nagalingam, and Grier C.I. Lin. "Towards Measuring the Effectiveness of a Facilities Layout." *Robotics and Computer-Integrated Manufacturing* 25.1 (2009).

[143] Ripon, Kazi Shah Nawaz, et al. "Genetic Algorithm Using a Modified Backward Pass Heuristic for the Dynamic Facility Layout Problem." *Paladyn, Journal of Behavioral Robotics* 2.3 (2011).

[144] Rokach, Lior, and Oded Maimon. *Data Mining with Decision Trees Theory and Applications (Series in Machine Perception and Artificial Intelligence)*. Singapore: World Scientific Publishing Company, 2007. Print.

[145] Romanow, Peter. *Konstruktionsbegleitende Kalkulation Von Werkzeugmaschinen*. Springer-Verlag, 1995. Print.

[146] Rosenblatt, Meir J. "The Dynamics of Plant Layout." *Management Science* 32.1 (1986).

[147] Rosenblatt, Meir J., and Hau L. Lee. "A Robustness Approach to Facilities Design." *International Journal of Production Research* 25.4 (1987).

[148] Sahin, Ramazan, Kadir Ertogral, and Orhan Turkbey. "A Simulated Annealing Heuristic for the Dynamic Layout Problem with Budget Constraint." *Computers & Industrial Engineering* 59.2 (2010).

[149] Sahni, Sartaj, and Teofilo Gonzalez. "P-Complete Approximation Problems." *Journal of the ACM* 23.3 (1976).

[150] Schrage, L. "Lindo: Linear Interactive Discrete Optimizer." Redwood City, CA: Scientific, 1982.

[151] Sechen, Carl, and Alberto Sangiovanni-Vincentelli. "The Timberwolf Placement and Routing Package." *IEEE Journal of Solid-State Circuits* 20.2 (1985).

[152] Sethi, Andreakrasa, and Sureshpal Sethi. "Flexibility in Manufacturing: A Survey." *International Journal of Flexible Manufacturing Systems* 2.4 (1990).

[153] Silver, Edward A., and Rein Peterson. *Decision Systems for Inventory Management and Production Planning*. 2nd ed. New York: Wiley, 1985. Print.

[154] Singh, S. P., and R. R. K. Sharma. "A Review of Different Approaches to the Facility Layout Problems." *The International Journal of Advanced Manufacturing Technology* 30.5 (2006).

[155] Sutanthavibul, Suphachai, Eugene Shragowitz, and J. B. Rosen. "An Analytical Approach to Floorplan Design and Optimization." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 10.6 (1991).

[156] Takahashi, Toshihiko. "An Algorithm for Finding a Maximum Weight Decreasing Sequence in a Permutation, Motivated by Rectangle Packing Problem." *Technical Report IEICE 96* (1996).

[157] Tang, Kit Sang, et al. "A Theoretical Development and Analysis of Jumping Gene Genetic Algorithm." *IEEE Transactions on Industrial Informatics* 7.3 (2011).

[158] Tang, Wallace S., Sam W. Kwong, and Kim Man. "A Jumping Genes Paradigm: Theory, Verification and Applications." *IEEE Circuits and Systems Management* 8.4 (2008).

[159] Tang, Xiaoping. "Fundamental Algorithms for Physical Design Planning of Vlsi." Thesis. The University of Texas at Austin, 2002.

[160] Tang, Xiaoping, Ruiqi Tian, and D.F. Wong. "Fast Evaluation of Sequence Pair in Block Placement by Longest Common Subsequence Computation." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 20.12 (2001).

[161] Tang, Xiaoping, and D.F. Wong. "Fast-Sp: A Fast Algorithm for Block Placement Based on Sequence Pair." *Proceedings of the ASP-DAC 2001 Asia and South Pacific Design Automation Conference 2001 (Cat. No.01EX455)* (2001).

[162] Toledo, Claudio Fabiano Motta, Renato Resende Ribeiro De Oliveira, and Paulo Morelato Franca. "A Hybrid Multi-Population Genetic Algorithm Applied to Solve the Multi-Level Capacitated Lot Sizing Problem with Backlogging." *Computers & Operations Research* 40.4 (2013).

[163] Tompkins, James A., et al. *Facilities Planning*. Hoboken, NJ: J. Wiley, 2003. Print.

[164] Trender, Lutz. "Entwicklungsintegrierte Kalkulation Von Produktlebenszykluskosten Auf Basis Der Ressourcenorientierten Prozesskostenrechnung." PhD Thesis. University of Karlsruhe, 2000.

[165] Tsourveloudis, N.C., and Y.A. Phillis. "Manufacturing Flexibility Measurement: A Fuzzy Logic Framework." *IEEE Transactions on Robotics and Automation* 14.4 (1998).

[166] U.S. Census Bureau, Statistics of U.S. Businesses. "Top 20 Facts About Manufacturing." NAM 13 Nov. 2016. Web.

[167] Ulutas, Berna Haktanirlar, and A. Attila Islier. "A Clonal Selection Algorithm for Dynamic Facility Layout Problems." *Journal of Manufacturing Systems* 28.4 (2009).

[168] Urban, Timothy L. "A Heuristic for the Dynamic Facility Layout Problem." *IIE Transactions* 25.4 (1993).

[169] Vanderplaats, Garret N. *Multidiscipline Design Optimization*. 1st ed. Monterey, CA: Vanderplaats Research & Development, 2007. Print.

[170] Wang, Laung-Terng, Yao-Wen Chang, and Kwang-Ting Cheng. "Floorplanning." *Electronic Design Automation: Synthesis, Verification, and Test*. Amsterdam: Morgan Kaufmann/Elsevier, 2009. 575-634. Print.

[171] Webster, Dennis B., and Michael B. Tyberghein. "Measuring flexibility of Job-Shop Layout." *International Journal of Production Research* 18.1 (1980).

[172] Womack, James P., Daniel T. Jones, and Daniel Roos. *The Machine That Changed the World: The Story of Lean Production--Toyota's Secret Weapon in the Global Car Wars That Is Revolutionizing World Industry*. Toronto 2007. Print.

[173] Yang, Taho, and Brett A. Peters. "Flexible Machine Layout Design for Dynamic and Uncertain Production Environments." *European Journal of Operational Research* 108.1 (1998).

[174] Yang, Taho, Brett A. Peters, and Mingan Tu. "Layout Design for Flexible Manufacturing Systems Considering Single-Loop Directional Flow Patterns." *European Journal of Operational Research* 164.2 (2005).

[175] Yang, Xin-She. "Flower Pollination Algorithm for Global Optimization." *International Conference on Unconventional Computing and Natural Computation*. Springer Berlin Heidelberg 2012. Print.

[176] Yang, Xin-She, Mehmet Karamanoglu, and Xingshi He. "Multi-Objective Flower Algorithm for Optimization." *Procedia Computer Science* 18 (2013).

[177] Zhan, Yong, Yan Feng, and Sachin S. Sapatnekar. "A Fixed-Die Floorplanning Algorithm Using an Analytical Approach." *Asia and South Pacific Conference on Design Automation* (2006).