An Intelligent, Knowledge-based Multiple Criteria Decision Making Advisor for Systems Design

A Thesis
Presented to
The Academic Faculty

by

Yongchang Li

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Aerospace Engineering
Georgia Institute of Technology
May 2007

An Intelligent, Knowledge-based Multiple Criteria Decision Making Advisor for Systems Design

Approved by:

Dr. Dimitri N. Mavris
Professor, Faculty Advisor
Georgia Institute of Technology

Dr. James M. McMichael
Director, Aerospace, Transportation and Advanced Systems Laboratory
Georgia Tech Research Institute

Dr. Daniel P. Schrage
Professor
Georgia Institute of Technology

Mr. Frank T. Ferrese
Lead Engineer
Office of Naval Research

Dr. Neil R. Weston
Research Engineer II
Georgia Institute of Technology

Date Approved: January 09, 2007

To my lovely wife,

Hong Yang,

for her endless love and inspiration.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Dimitri Mavris, for his encouragement and support these years. I gratefully acknowledge the invaluable guidance and advisement he has provided to me throughout this process. I really appreciate the opportunities he has given me and cannot say enough about my gratitude to him.

I would also like to thank Dr. Neil Weston whose valuable feedbacks and suggestions help me shape this dissertation in many ways. Many thanks go to Dr. James McMichael who was kind enough to meet with me several times to discuss this work and offer thoughtful advice. I also want to thank Dr. Daniel Schrage for his comments and remarks which are valuable in improving this dissertation. I would also like to thank Mr. Frank Ferrese of ONR for participating in my thesis defense and always promptly responding to my questions associated with this work.

My fellow graduate students have also been enormously helpful throughout the course of my studies. Specifically, I would like to thank Henry Won, Tori Thomas, Samson Lim, Santiago Balestrini, and Stephen Dufresne.

Finally, I would like to thank my family and loved ones. Thank my parents for their unending love and support along the path of my academic pursuits. I have been a long way from home for a long time and could not make step forward without their understanding and support. I would like to express my deepest gratitude to my lovely wife Hong, without her sincere encouragement, great patience and mostly unconditional love, I would not have been able to get through this long journey.

Thank you all!

Yongchang Li

Atlanta,, GA

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

AHP	Analytical Hierarchy Process
AI	Artificial Intelligence
CR(s)	Customer Requirement(s)
CTOL	Conventional Take-Off and Landing
CW-RSAD	Chilled Water Reduced Scale Advanced Demonstrator
DDM	Distributed Decision Making
DDMUU	Dynamic Decision Making Under Uncertainty
DDSS	Distributed Decision Support System
DM(s)	Decision Maker(s)
DSS	Decision Support System
EC(s)	Engineering Characteristic(s)
EDF	Empirical Distribution Function
ES	Expert System
EUT	Expected Utility Theory
FPI	Fast Probability Integration
GP	Goal Programming
IEP	Integrated Engineering Plant
IPPD	Integrated Product Process Development
IRIS	Integrated Reconfigurable Intelligent System
IS	Information System
JPDF	Joint Probability Density Function
JPDM	Joint Probability Decision Making

JPM	Joint Probability Model
KDD	Knowledge Discovery in Databases
MADM	Multiple Attribute Decision Making
MCDM	Multiple Criteria Decision Making
MIDAS	Multiple-Criteria Interactive Decision-Making Advisor and Syntheses process
MCS	Monte Carlo Simulation
MDP	Markov Decision Process
MODM	Multiple Objective Decision Making
OEC	Overall Evaluation Criteria
ONR	Office of Naval Research
PAV(s)	Personal Air Vehicle(s)
POS	Probability of Success
QFD	Quality Function Deployment
RSAD	Reduced Scale Advanced Demonstrator
RSE(s)	Response Surface Equation(s)
SE	Systems Engineering
TIES	Technology Identification, Evaluation and Selection
TIF	Technology Impact Forecasting
TOPSIS	Technique for Ordered Preference by Similarity to the Ideal Solution
TRIZ	Theory of Inventive Problem Solving
UAV	Unmanned Aerial Vehicle
UTE	Unified Tradeoff Environment

VTOL	Vertical Take-Off and Landing

SUMMARY

In systems engineering, design and operation of systems are two main problems which always attract researcher's attentions. The accomplishment of activities in these problems often requires proper decisions to be made so that the desired goal can be achieved, thus, decision making needs to be carefully fulfilled in the design and operation of systems.

Design is a decision making process which permeates through out the design process, and is at the core of all design activities. In modern aircraft design, more and more attention is paid to the conceptual and preliminary design phases so as to increase the odds of choosing a design that will ultimately be successful at the completion of the design process, therefore, decisions made during these early design stages play a critical role in determining the success of a design. Since aerospace systems are complex systems with interacting disciplines and technologies, the Decision Makers (DMs) dealing with such design problems are involved in balancing the multiple, potentially conflicting attributes/criteria, transforming a large amount of customer supplied guidelines into a solidly defined set of requirement definitions. Thus, one could state with confidence that modern aerospace system design is a Multiple Criteria Decision Making (MCDM) process.

A variety of existing decision making methods are available to deal with this type of decision problems. The selection of the most appropriate decision making method is of particular importance since inappropriate decision methods are likely causes of

misleading engineering design decisions. With no sufficient knowledge about each of the methods, it is usually difficult for the DMs to find an appropriate analytical model capable of solving their problems. In addition, with the complexity of the decision problem and the demand for more capable methods increasing, new decision making methods are emerging with time. These various methods exacerbate the difficulty of the selection of an appropriate decision making method. Furthermore, some DMs may be exclusively using one or two specific methods which they are familiar with or trust and not realizing that they may be inappropriate to handle certain classes of the problems, thus yielding erroneous results. These issues reveal that in order to ensure a good decision a suitable decision method should be chosen before the decision making process proceeds.

The first part of this dissertation proposes an MCDM process supported by an intelligent, knowledge-based advisor system referred to as Multi-Criteria Interactive Decision-Making Advisor and Synthesis process (MIDAS), which is able to facilitate the selection of the most appropriate decision making method and which provides insight to the user for fulfilling different preferences. This advisor consists of an MCDM library storing the typical decision making methods widely used in dealing with the decision making problems and a knowledge base providing the information required in the method selection process.

The most suitable method is selected through an intelligent reasoning process utilizing the information in the knowledge base. This method selection is based on the concept that the characteristics of the method should “best” satisfy the applicable problem related criteria. Once the most appropriate method is selected for the given problem, the advisor is also able to aid the DM to reach the final decision by following the rigorous problem solving procedure of the selected method. The advisor is also able to provide guidance as to the requirements needed to be fulfilled by a potentially new method for cases where no suitable method is available in the library. In addition, the advisor is capable of validating the decision made using one specific method and aid the DM to arrive at a better decision if the decision made is not appropriate.

In many other domains, such as complex system operation, proper decision making is required to keep the system working functionally and effectively. This type of decision making often occurs in a dynamic environment with rapidly changing situations, and is completed based on the assessment of uncertain or incomplete information due to the data availability and variation of the operational environment. Therefore, an advanced decision making strategy is needed not only to capture the system's dynamic characteristics and environmental uncertainty but also to meet the operational objectives. Particularly, in naval ship operation, more emphasis has been placed on increasing the mission effectiveness and ship survivability, and reducing cost and manning workload. To satisfy these requirements right decisions should be made to determine the most suitable actions taken in different system states, as a result, the best course of action needs to be identified.

The second part of this dissertation presents an autonomous decision making advisor which is capable of dealing with ever-evolving real time information and making autonomous decisions under uncertain conditions. The advisor encompasses a Markov Decision Process (MDP) formulation which takes uncertainty into account when determines the best action for each system state. The execution of the actions consumes resources, which results in a resource allocation problem. Thus, the resource allocation problem can be achieved by finding the optimal policy which specifies the best action to take for each of the states. As a result, the limited resources are reallocated to different agents under various scenarios to maximize the total rewards obtained from executing the actions. The successful resource allocation leads to a reconfiguration of the system which is the most suitable to handle the situation at hand.

CHAPTER I

INTRODUCTION

Engineering provides a variety of tools and approaches to develop solutions to diverse problems such as design, production and operation of products or processes. Systems Engineering (SE) is an interdisciplinary engineering management process which integrates multiple engineering tools, approaches and disciplines to realize and deploy successful systems satisfying customer requirements [Defense Acquisition University Press, 2001]. This indicates SE forms a structured approach which is able to facilitate the activities in design, production and operation of systems. To accomplish these activities, proper decisions require to be made to determine what actions need to be performed and how they are carried out so that the desired goal can be achieved. As a result, decision making becomes an essential part of the problem solving procedure.

Design is about using available information to make intelligent decisions leading to optimal solutions which satisfy the customer's requirements. During the design phases, decision making permeates through the entire design process, and is at the core of all design activities. Problem definition, for example, involves deciding what the customer requirements are, and how to define the constraints and targets. Other design activities such as alternative concepts generation, technology infusion, and concept selection heavily rely on or are pure decision-making processes. In addition, the selection of the design parameter, the basic element of the design process, represents the decision. Therefore, one can state with confidence that design is a decision making process.

In modern aircraft design, progressively more and more emphasis has been given to the conceptual and preliminary design phases so as to increase the capability of choosing an optimal or a robust design. Decisions made during these stages play a central role in determining the success of the design. This new paradigm in aerospace system design must deal with the increased desire for reducing costs, increasing profit, increased performance, environmental friendliness and quality. The DMs are involved in balancing the multiple, potentially conflicting attributes/criteria, and transforming a large number of customer supplied guidelines into a solidly defined set of requirement definitions. As a result, many criteria have to be all simultaneously taken into account, and a compromise becomes an essential part of the decision making process. Therefore, decision making in the conceptual and preliminary system design stages apparently has multi-level, multi-criteria with uncertain and sometimes incomplete information in nature.

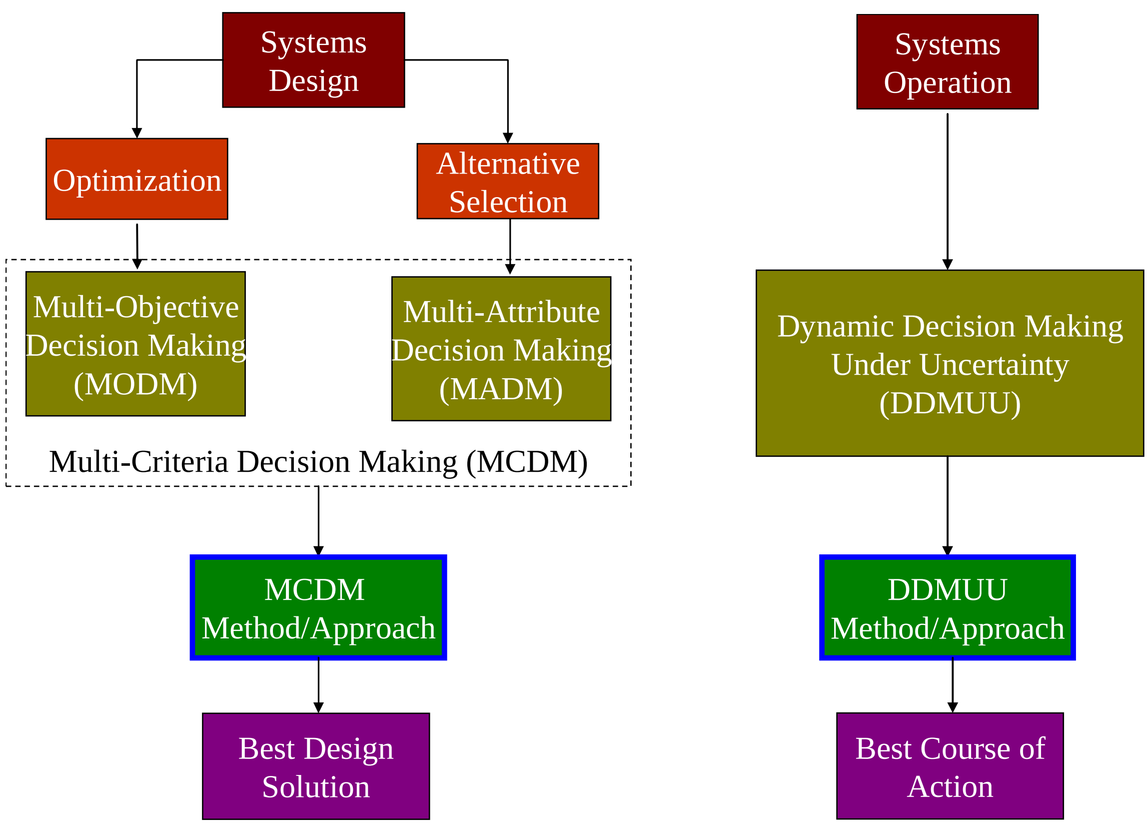
To handle this type of Multiple Criteria Decision Making (MCDM) problem in the early design stage, various methods have been developed. Currently, over 70 decision making methods have been proposed with the intention of facilitating the decision making process, and have already been applied to deal with different decision problems. With the complexity of the decision problem and the demand for more capable methods increasing, new methods keep emerging. Paradoxically, these numerous methods don't ease the decision problem as they are expected to do, but complicate the problem because one has to determine which method is appropriate before he/she can proceed, considering the fact that the use of inappropriate method may create misleading solutions to the decision making problem. However, figuring out the appropriate decision making method may be viewed as a difficult problem for the DMs since this selection itself is a complicated MCDM problem. One part of this dissertation attempts to formulate a process which explores the appropriateness of the decision making methods and selects the one that is the most appropriate to solve the problem under consideration.

In the case of complex systems operation, proper decisions need to be made to keep the system functioning properly and effectively. Since systems operation often occurs in an environment with rapidly changing situations and uncertain conditions, the data gathering, processing and evaluation must be fast enough to support the decision making which is able to capture the dynamic characteristics and uncertainty existing in the problem. During the period of operation, the right action should be determined at each decision epoch based on the state of mission, operational environment and system status. Thus, the primary goal of the systems operation is to identify and perform the best action in each system state to maximize the system effectiveness and minimize cost. After the action is executed, the system randomly transits to next state. However, choosing the best action requires thinking about more than just the immediate effects of the actions because the action results in maximum immediate reward may cause side-effect in the future. Therefore, tradeoff should be done between the immediate rewards and the future gains to yield the best possible solution. This fact indicates that in complex systems operation, sequential decisions should be made in a dynamic environment to identify the best course of action for a stochastic process. This type of decision making is hard to be successfully accomplished by an individual DM or even a group consisting of wise DMs since it is always a source of difficulty for DMs to make dynamic decisions and take the future effect of the decisions into account. The other part of this dissertation proposes a dynamic decision making formulation which is able to find the best course of action for systems operation problem under uncertain conditions.

1.1 Motivation

Decision making ubiquitously occurs in many areas, including systems design, manufacturing and systems operation. Traditionally, a decision is made by an individual or a group of DMs based on their intuition, values and preferences. Decisions made using this approach often highly depend on the DMs' experience and preference, therefore the quality of the decision made varies a lot with different DMs. In addition, for a specific DM, decisions made for the same problem at different times may be different because his/her preference is not always consistent.

In systems design, decision making permeate the design steps, such as defining requirements and targets, creating the optimal solutions, and making the final decision. The decisions in those steps should be carefully made in order to obtain an optimum and/or robust design, however, the traditional approach is known to be incompetent to make such wise decision. Thus, advanced decision making method should be used. Design process often starts from a set of customer requirements expressed in term of objectives, goes through several steps iteratively and then creates desired alternatives which meet the specified objectives. In this design alternative generation step, Multi-Objective Decision Making (MODM) methods are often utilized to facilitate this design process. Typically, MDO methods handles the problems which "involve the design of alternatives which optimize or 'best satisfy' the objectives of the decision maker" [Hwang and Masud, 1979]. Once the optimal alternatives are generated, the final selection will be made to determine the one that "best" meets the customer requirements and DM preference. This leads to an alternative selection problem which is often solved using Multi-Attribute Decision Making (MADM) methods. It can be seen that MODM problems are optimization problems while MADM problem are alternative selection problems, and the methods for solving both problems is classified as Multi-Criteria Decision Making methods [Bandte, 2000]. To obtain the best solution to the design problem, an appropriate MCDM method needs to be used since the use of inappropriate method may lead to misleading solution. Thus, it is critical to select the most appropriate decision making method for the problem under consideration, which is illustrated in Figure 1(a) and will be detailed described in Section 1.1.1.



(a) Selection of the most appropriate decision making method in systems design

(b) Dynamic Decision making under uncertainty in systems operation

Figure 1: Decision Making in Systems Design and Operation

In the case of complex systems operation, decision making is required to be fast enough to handle a large amount of information which is changing over time. In addition, the operation of the system usually occurs in an environment where uncertain conditions are always involved, thus uncertainty needs to be taken into account. Furthermore, decision made at a certain decision epoch has effect on future system state, which further complicates the decision making process. Apparently, these complexities make it difficult for DM to make decisions by employing the traditional approach. In order to improve the quality of decision making and identify the best course of action for the system, a more advanced decision making approach requires to be developed to be able to capture the essence of the systems operation problem. This need is shown in Figure 1 (b) and will be explained in Section 1.1.2.

1.1.1 Selection of the Most Appropriate Decision Making Method

More and more emphasis has been given to conceptual and preliminary design stages in modern aerospace system design in order to increase the probability of success of a design at the completion of the design process. To achieve the success in these phases one is expected to bring as much knowledge as possible forward and maintain efficient freedom in these early stages to avoid locking in the cost [Mavris et al., 1998; Mavris and DeLaurentis, 2000a].

The essence of this new design paradigm is to increase the knowledge in early design stages so that wise decisions can be made. It is clear that decisions made during the conceptual and preliminary design phases have a considerable impact on the final design solution. Thus, decision making, which is at the core of the design process, needs to be carefully formulated and carried out. To reach a good design decision, the problem identification, including the investigation of the requirements, may be one of the two most important parts of the whole process. If the identification does not capture the essence of the problem, it is most likely that the final solution is misleading since the design decision is based on the wrong structure of the problem. This issue has attracted significant attention and been handled in several ways [Neufville, 1990; Kirby, 2001; Garcia, 2002; Hollingsworth, 2004]. The other most valuable part of the decision making process is to determine the most appropriate decision making method for the problem under consideration before the decision making proceeds. The importance of the selection of the most appropriate method results from the fact that the use of an inappropriate method could lead to an unjustified decision though a well defined problem is achieved. It has been recognized that the systematic analysis model can highly improve the effectiveness of the decision making, thus this fact stimulates many research works concentrating on developing MCDM methods. As a result, numerous methods were proposed and available to handle different decision making problems. These available methods certainly ease the decision making process by giving DMs various options in solving their problems, however, on the other hand, they complicate the decision making process from the beginning since DMs have to select the most appropriate method among the existing methods for their specific decision problems. It is obvious that the selection of the most appropriate method has critical impact on the decision making process since the use of an inappropriate method may result in an undesired solution, however, it is an area that has not been given adequate consideration.

1.1.1.1 *Existence of Various Decision Making Methods*

Many efforts have been made to facilitate the MCDM process so that various methods and techniques have been developed, such as Simple Additive Weighting (SAW), Technique for Ordered Preference by Similarity to the Ideal Solution (TOPSIS) [Hwang and Yoon, 1981] and Analytical Hierarchy Process (AHP) [Saaty, 1980]. Up to now, over 70 MCDM methods [Roman et al., 2004] have been proposed, and each method has a different analysis model intending to solve some class of problem. Furthermore, new methods are continuously emerging aiming at handling more complicated decision making problems.

The existence of the various decision making methods implies that different methods have their own advantages and disadvantages and there is not a general, universal method capable of handling all types of problems. This fact indicates that in order to obtain a desired solution for the problem under consideration a suitable method should be utilized since the existing methods have different degrees of appropriateness in handling a given problem. This statement can be further supported by the fact that for a given problem significantly different conclusions may be obtained from the application of the various methods.

For example, as shown in Table 1, when the DMs of an airline consider purchasing one aircraft among three competing aircraft designs based on the attributes of interest, they will make their decision based on the solution obtained by using a specified decision making method of choice. Study shows that, with the same preference information (i.e. all attributes have same weight), aircraft C is recommended as the “best” design by AHP method [Hazelrigg, 2003] while aircraft A is suggested as the one to buy by TOPSIS.

However, SAW will select either aircraft A or aircraft C dependent on the attribute values of the baseline.

Table 1: An Example of MCDM Problem

Attribute	Airplane A	Airplane B	Airplane C
Range	1500	2000	3000
Speed	550	450	600
Payload	30000	25000	50000
Cost	15 M	20 M	10 M
Reliability	0.97	0.98	0.999
Safety	0.99999	0.99999	0

One can easily see that aircraft C has no safety at all and obviously nobody is going to take it, hence, it is not a design that any airline will spend money on. This fact indicates that AHP and SAW, which recommend the undesired solution, are not the appropriate methods for the problem under consideration. On the other hand, TOPSIS is a better choice for this problem. However, it is not prudent to conclude that TOPSIS excels the other two methods in solving decision making problem because TOPSIS' appropriateness over AHP and SAW is only valid under some conditions. That is, it is justified to state that TOPSIS is a more suitable method than the other two methods when handing the decision problem described in Table 1 but this statement does not hold for any other decision problem.

From this example, two observations can be formulated:

Observation 1: Various decision making methods have been proposed to deal with the decision problem. The methods have their own advantages and disadvantages.

Observation 2: Different decision making methods may finally produce diverse solutions to the same problem, and undesirable solutions can be obtained by the utilization of some inappropriate methods.

Therefore, an appropriate method is necessary to be selected ahead of the decision making process in order to get the desired solution for the problem under consideration. Unfortunately, it is not always an easy task for the DMs to select the most appropriate decision making method among a large number of available ones without knowing their characteristics, that is, basically, their advantages and disadvantages. This issue is always a source of frustration for the DMs.

1.1.1.2 Method Preference and Knowledge Limitation

When it is required to perform analysis and selection of alternatives, some DMs may always use the methods or techniques which they are familiar or feel comfortable with for any problem under consideration. Typically, they trust these methods because they believe these methods can generate the "best" and/or robust solutions for almost all types of problems. This method preference indicates that some DMs do not recognize or even often ignore the importance of selecting the most suitable method for a specific problem. This usually stops them exploring other more appropriate method and techniques to solve the given problem. However, as discussed before, no universal method can solve all types of problems and the use of an inappropriate method will result in a misleading solution, thus, the method preference often misdirects the DM's judgment. This fact leads the observation below:

For instance, some DMs think TOPSIS is a great technique to deal with most of the decision problems, so they tend to use it to solve any problems involving decision making. Consider the example presented in §1.2.1 with two changes: 1) the airline has a requirement for safety, which is the safety must be greater than 0.8; 2) instead of totally unsafe, aircraft C has a safety of 0.2. Study shows that with these two modifications TOPSIS selects aircraft C as the "best" design evaluated by the six criteria listed in Table 1. However, obviously aircraft C is not a feasible design because it violates the safety requirement and no airline will buy it to risk their business. The reason that TOPSIS selected aircraft C is that this aircraft dominates in every attribute except safety, and has the highest average goodness. TOPSIS' decision rule determines the alternative with the highest average goodness will be selected as the best solution, therefore aircraft C is chosen as the one. This inconsistency indicates that TOPSIS may suggest a design as the "best" solution even it is an infeasible design, which makes TOPSIS alone not an appropriate method to solve this specific problem.

The other reason why people tend to use the methods that they are familiar with is that they have limited knowledge on the other methods. Since there are numerous decision making methods available and new methods are emerging with time, the difficulty of finding the "best" method for the given problem is increasing. Each of the methods has its own characteristics, so to understand all these methods is time consuming and tedious. In addition, it is not appropriate to require a DM to know all the decision methods because it is not practical and not necessary.

Based on the discussion in this section, an observation can be formulated:

Observation 3: Due to method preference and knowledge limitation, some DMs employ one or two methods to solve any given problem which is often not appropriate for the problem under consideration.

Obviously, a approach is needed to help the DMs select the most suitable method before decision making is performed, and then provide guidance to aid the DMs reach the final decision by following the decision making procedure of the selected method.

1.1.1.3 New Method Generation

With the evolution of the requirements and technologies, the complexity of the decision problem is increasing, so existing methods may be incapable of dealing with these types of problems. This phenomena leads to the following observation:

Observation 4: In some cases, it is not able to find an appropriate method among the existing ones to handle the new decision problem.

This stimulates the demand for developing advanced methods. To develop a new method, some disciplined approach may need to be employed, such as morphological matrix [Dieter, 2000] and Theory of Inventive Problem Solving (TRIZ)[1] [Braham, 1995]. These techniques are widely accepted to generate new ideas by revealing all possible solution concepts and developing the superior one among them.

The development of a new decision making method can also emerge in the process of selecting the most appropriate method for the given problem. In the method selection process, the characteristics of the candidate methods and the given problem are thoroughly inspected, which often produces a new perspective on what capabilities are required for a method to be fulfilled to deal with the problem. However, a new decision making method may be generated in the procedure of selecting the most appropriate

---

[1] TRIZ is Russian acronym for Theory of Inventive Problem Solving.

method for a given problem. These requirements typically service as a baseline for new method development. For example, in the aircraft selection example described in §1.2.2, TOPSIS was considered as an inappropriate method for solving the given problem. The reason is that TOPSIS is a method which ranks the alternatives based on the concept that the "best" alternative has the closest distance from positive ideal solution and furthest distance from negative solution. The distance from the ideal solution is in the form of Euclidean distance, which is an equivalent to the average goodness. Therefore, TOPSIS may select an alternative with the highest average goodness as the "best" solution which is dominative at other attributes but violates one or more constraints, that is, an infeasible solution. This paradox inspires a motivation of either finding another existing method that can overcome the drawbacks of TOPSIS or developing a new method with the improved capability over the current TOPSIS. As noticed before, aircraft C is an infeasible design, thus, performing a feasibility evaluation before employing TOPSIS may smoothly solve this problem and result in a desired solution. This leads to an advanced method adapted from TOPSIS which has higher capability to handle the decision problem. This shows that the selection of the "best" appropriate decision method is able to provide useful hints for the new methods generation.

1.1.1.4 Previous Research Work on Method Selection

Over the past decades, many efforts have been made to facilitate the selection of the most appropriate decision making method for a given problem. MacCrimmon [MacCrimmon, 1973] is probably the first researcher who recognized the importance of MCDM method selection. He proposed a taxonomy of MCDM methods, created a method specification chart in the form of a tree diagram and provided an illustrative application example.

These works provided a methodological basis for the development of a comprehensive MCDM knowledge base. A taxonomy similar to the one MacCrimmon proposed was developed by Hwang and Yoon[Hwang and Yoon, 1981]. This taxonomy is also represented by a tree diagram which consists of nodes and branches connected by choice rules. Sen and Yang [Sen and Yang, 1998] developed two similar tree diagrams to help select the appropriate Multi-Attribute Decision Making (MADM) and Multi-Objective Decision Making (MODM) method among a few typically used methods. The tree diagram for selecting the suitable MADM method is illustrated in Figure 2. The tree diagram approach provides reasonable classification schemes and is easy to utilize. However, this approach has its own disadvantage: it usually gives two or more MCDM methods rather than the most appropriate method for the decision problem under consideration, and only considers limited types of decision problems, preference information and the available methods. These limitations stop the tree diagram approach from being an effective solution to the method selection problem.

Possible criteria for evaluating MCDM methods were proposed as an alternative solution to this method selection problem [Evans, 1984; Gershon and Duckstein, 1984; Hobbs, 1986; Ozernoy, 1987; Tecle and Duckstein, 1992]. Gershon and Duckstein suggested selecting the “best” MCDM method by evaluating the methods with respect to a set of criteria which fall into one of four categories: mandatory, non-mandatory, technique-dependent and application-dependent [Gershon and Duckstein, 1984]. The methods are evaluated by the criteria until the most suitable method for the given problem is found. Hobbs suggested performing the experiments in multiobjective analysis to evaluating the methods based on four criteria: appropriateness, ease of use,

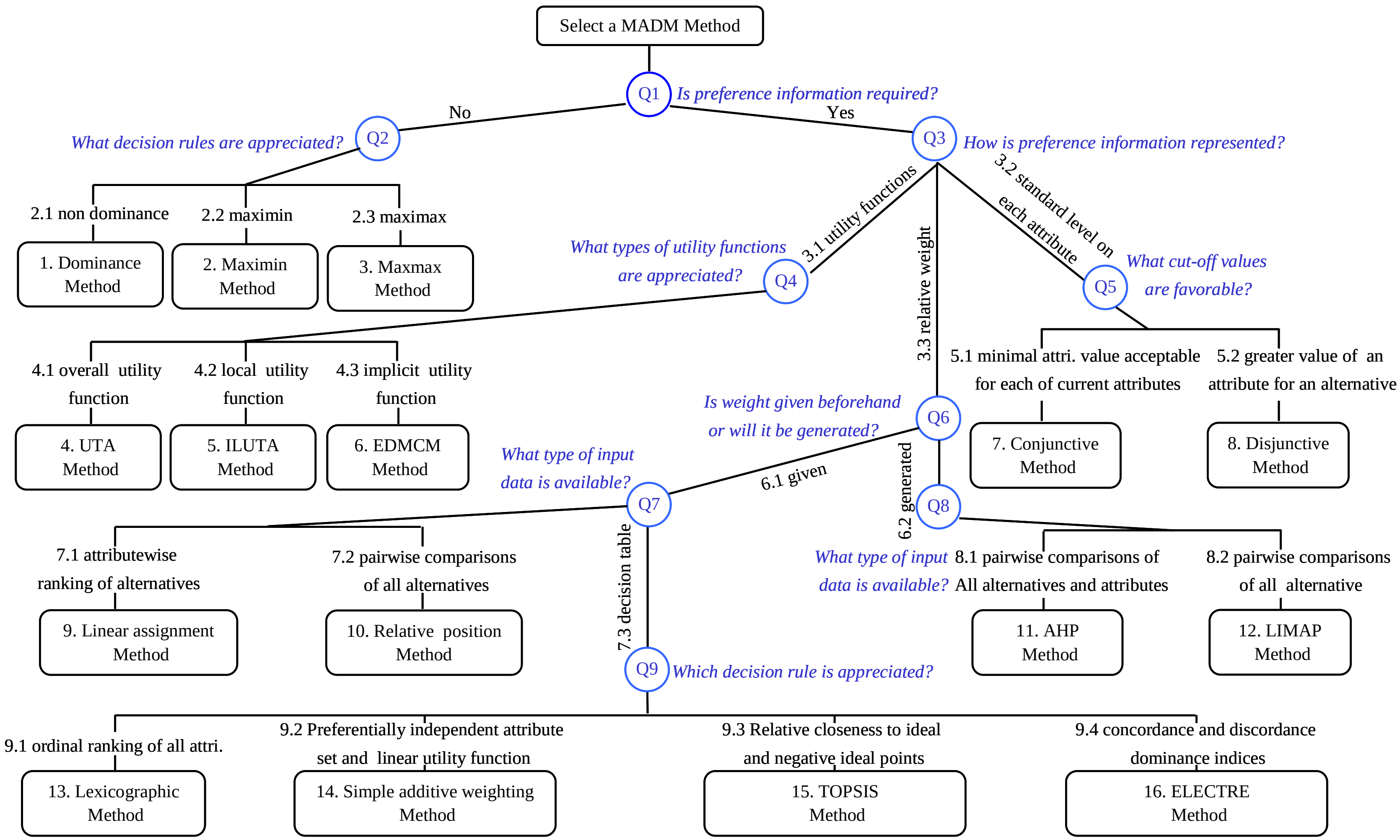


Figure 2: Decision Tree for MADM Technique Selection [Sen and Yang, 1998]

validity and sensitivity of results to choice of method [Hobbs, 1986]. Ozernoy utilized a hierarchical model which employed screening criteria and evaluation criteria for selectingthe most appropriate MCDM method [Ozernoy, 1987]. Tecle and Duckstein developed an approach based on a composite programming algorithm in order to handle the selection of the most suitable MCDM method. They proposed four categories of the criteria: DM related, technique related, problem related and solution related [Tecle and Duckstein, 1992], and these categories were adopted by the sequent researchers such as Poh and Lu et al [Poh, 1998; Lu et al., 1999]. However, a major difficulty which prevents wildly using these approaches is “the lack of universally accepted data on discrete alternative MCDM methods that would allow the quantification of the methods in terms of these criteria” [Ozernoy, 1992]. And by using these approaches different users may get totally different results because the user’s knowledge about the MCDM methods has a strong impact on the final results.

In the early 1990s, researchers began to employ the techniques of artificial intelligence to improve the quality of the decision making method selection. Ozernoy developed an expert system for choosing the best MCDM method, and presented a small example as a proof of implementation. He identified and used three types of characteristics associated with MCDM problem, DM and MCDM method, respectively. And the selection of the “best” MCDM method is considered as “a search for the best arguments supporting the match among those characteristics” [Ozernoy, 1992]. The expert system works by asking the user a series of questions and then eliminating options to the most appropriate method based on the user’s answers. Poh also employed an expert system to facilitate the selection of the most suitable MADM method, and the architecture of this system is shown in Figure 3. Compared to Ozernoy's approach, Poh's system explicitly consists of a knowledge base which is utilized by the system to provide the guidance in selecting the most suitable method [Poh, 1998]. Similar to Poh, Lu et al proposed an intelligent multiple objective decision support system that can aid DMs in the method selection [Lu et al., 1999]. These expert and intelligent system approaches simplify the method selection procedure with simple questions and allow direct selection or automated selection based on the inputs provided by the user. However, these approaches have their own limitations: they don't have a comprehensive sample of MCDM methods in their system, and they don't clearly state the limitations or failure modes of the systems. And some of them are not accessible.

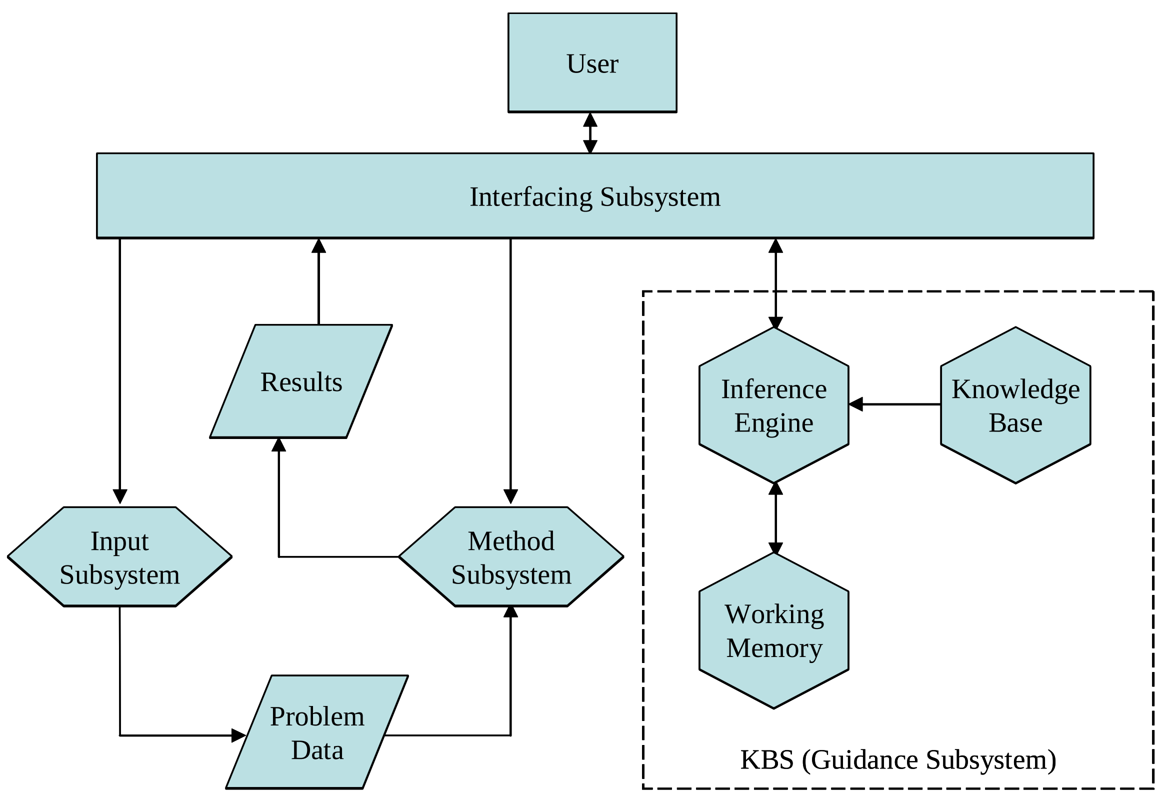


Figure 3: General Architecture of the Poh's Expert System [Poh, 1998]

The following observation concludes the previous work that has been done on decision making method selection:

Observation 5: Various approaches have been proposed to facilitate the selection of the most appropriate decision making method, but their limitations stop them from being an effective approach to handle this type of problems.

1.1.1.5 New MCDM Method Selection Approach Is Needed

The decision making problems are becoming more and more complicated for the system design with the evolution of the requirements and technology. Therefore, it is more important to select the most appropriate MCDM method for the problem under consideration since the use of an inappropriate method often leads to misleading decisions and eventually produces undesired designs which will result in high cost to the manufacture and consumer. Although the approaches described in Section 1.1.1.4 present some capabilities to find the suitable decision making method for a given problem among candidates, they have their own disadvantages in handling this type of problems. Some of them require that the user has certain knowledge about different methods (e.g. criteria approach), and some of them are too simplistic to suggest the most suitable method (e.g. tree diagram). In addition, all of the approaches don't have a comprehensive sample of the existing MCDM methods. This lack of methods in the selection pool means the selected method using these approaches may not be the most appropriate method for the problem under consideration since the most appropriate method may be existing but is excluded from being selected. Furthermore, the existing approaches are not able to produce the final solution to the given decision making problem. They either cannot find the most appropriate method for the given problem or just find and display the name of the selected method, but not to provide guidance to user how to get the final solution in these cases, as shown in Figure 4. Therefore, a new approach with more capabilities needs to be developed to facilitate the MCDM method selection.

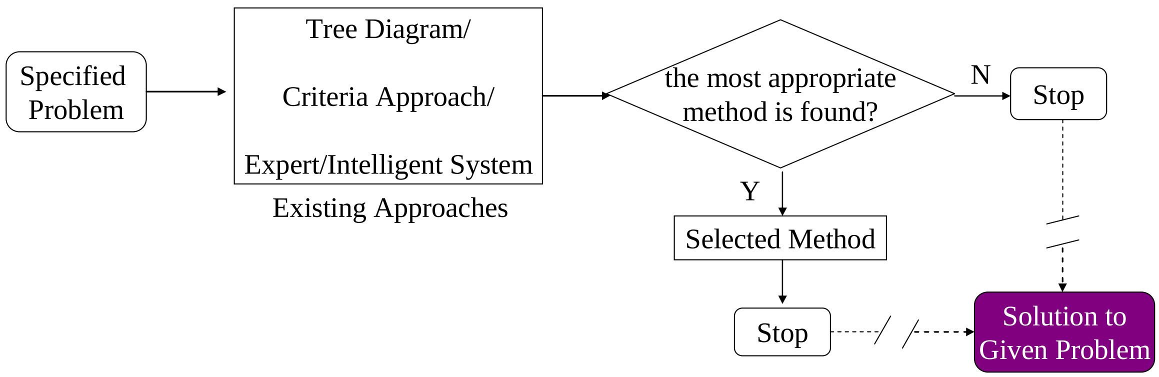


Figure 4: Limitations of Existing Method Selection Approaches

1.1.2 Dynamic Decision Making Under Uncertainty

In many circumstances, multiple decisions need to be made over time to reach a desired goal. Uncertainty is usually involved in this type of decision making process since it is hardly to deterministicly or perfectly predict the consequence of a decision after it is executed. In addition, decision made in a certain state has effect on the future state of the system and thus affects the overall goal. This sequential decision making process is illustrated in Figure 5.

As mentioned before, complex systems operation is such a decision making process. The goal of complex systems operation is to identify the best action in each state to deal with the situation at hand, as a result, the system will act on the best course of action so the objective of the operation can be maximized. Usually, the decision is made based on the assessment of a large amount of information which changes over time, thus the decision making should capture the dynamic characteristics of the system. In addition, due to incomplete knowledge and uncertain information about the operational environment, uncertainty should be considered as a major factor when making decisions.

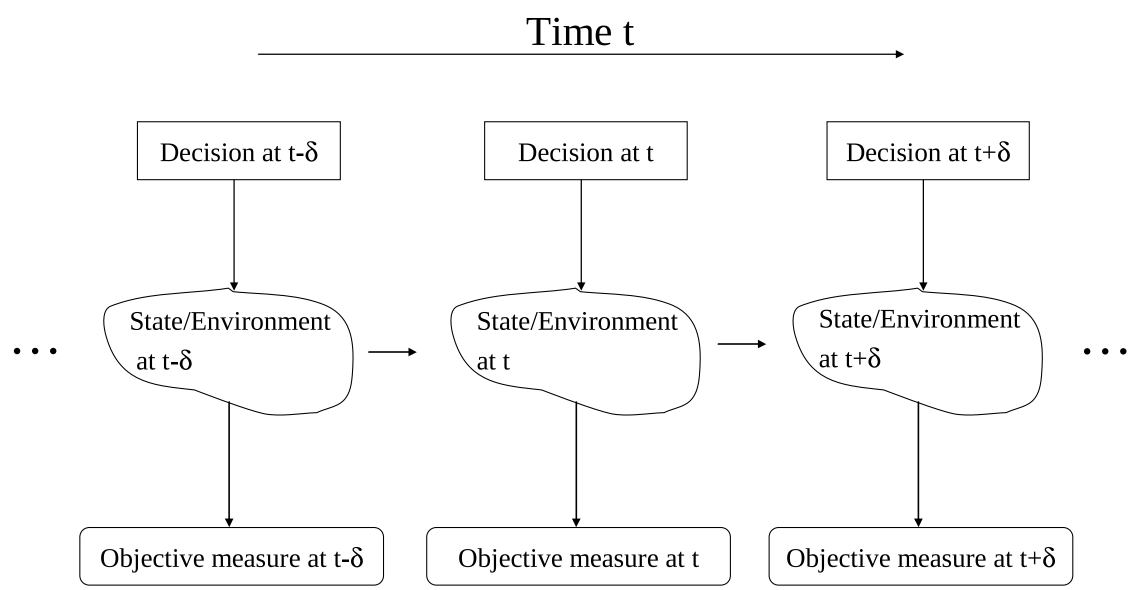


Figure 5: Dynamic Decision Making Under Uncertainty [Leong, 1993]

1.1.2.1 Integrated Reconfigurable Intelligent Systems (IRIS)

In modern ship design, more and more emphasis has been given to reducing cost and manning workload, and increasing survivability and mission effectiveness. The Office of Naval Research (ONR) Integrated Engineering Plant (IEP) concept has potential of meeting such future Navy requirements. IEP is a unified system that combines engineering and damage control services under a common control architecture. The IEP system will allow the next generation Navy ships to operate under major disruptions involving cascading failures and provide continuous mobility, power, thermal management and fluid transfer for vital shipboard systems, thus reducing manpower requirements and increasing overall ship survivability and effectiveness. This revolutionary change in naval architecture and ship engineering requires a total ship systems engineering design approach which is able to formulate and implement the design methods and tools to the ship systems and capable of extensive, autonomous decision making.

The Integrated Reconfigurable Intelligent Systems (IRIS) framework is proposed as a possible solution to formulate the IEP problem. The design of the IRIS is shaped by the integration of intelligent and reconfigurable systems, incorporating interactions and interdependencies. With the reconfigurable systems, the ship, based on the incoming information, will assess and then configure itself into the mode most adequate to deal with the situation at hand. Moreover, the ship is able to be aware of its surroundings through the gathering of data from sensors onboard the vehicle and provide guidance to a human operator as to the best course of action. In summary, an IRIS-designed ship is envisioned to be self-monitoring, self-assessing and self-reacting.

In the process of obtaining the best course of action, decision making often occurs in a dynamic environment with rapidly changing situations under which the ship is operated. In addition, the overflow of information makes it difficult to perform analysis and make proper decisions. Furthermore, in order to increase the survivability and effectiveness of the ship, the reactions are required to be taken in a dynamic manner, thus the data gathering, processing and evaluation must be fast enough to support the real-time decision making process. In general, the reactions are determined by the overall assessment which is a combination of the different assessments produced by the various systems for the same event in terms of urgency and priority. And this assessment is based on the states of mission being performed, ship status and operational environment. For example, in the case that damage occurs during the battle, the power is required to be redistributed to the different systems, such as weapons system, damage control systems, and radar systems in order to reconfigure the ship into the state most suitable to deal with the current situation. Under such a scenario, the information from various sensors indicating the states of the systems and the environmental situation varies over time. In order to ensure the ship system operates with maximum survivability and mission effectiveness, real-time decisions need to be made based on the assessments produced by using the collected information and accounting for future events by forecasting their effects to relocate the electrical power.

Traditionally, the decision making in complex system operation is completed by human DMs based on the assessment obtained by analyzing the incoming information. However, the IEP problem requires real-time decisions to meet the system's requirements, which is often very hard to be accomplished by a human operator. Thus, an advanced decision making approach is needed. This decision making approach should be capable of making dynamic decisions and capturing the uncertainty that exists in the system operation process. An autonomous decision making advisor with the abilities to handle the potentially conflicting multiple criteria and make real time dynamic decisions is capable of fulfilling these tasks. This advisor system is envisioned to assess the time-dependent information and provides the best course of action most suitable to the current state of the system with respect to the ship effectiveness, cost and survivability.

1.2 *Research Statement*

1.2.1 Research Goal

The focus of this research consists of two parts. One part of the research focus on developing an intelligent, knowledge-based, high ability decision making advisor system to select the most appropriate MCDM method among a reasonably large selection pool, and then guide the DM to reach the final decision utilizing the selected method. The advisor should be able to select the most suitable method from the candidate methods for the problem under consideration, validate the decisions made by using a specified decision making technique, and provide plausible advices that can act as the hints for developing new decision making methods if no method in the selection pool is suggested. In addition, the advisor should be capable of performing the feasibility evaluation on the decision alternatives before the decision making process proceeds. The other part of this research is to develop an autonomous decision making advisor to deal with the decision making under uncertain conditions. The advisor is implemented to a resource allocation problem for a ship system. This advisor should be able to handle the information changing over time and provide a best course of action most suitable to handle the current situation in order to increase the effectiveness and survivability of the ship.

1.2.2 Research Questions and Hypotheses

The research described and proposed in this dissertation is motivated by several key factors, which are best introduced through a series of questions. The research questions for the first part of this research are listed below and new questions will be come up with as the research proceeds.

Question 1: How to represent different methods in order to capture their essence for method selection? (Observation 1)

Question 2: How to evaluate the appropriateness of the methods for the problem under consideration? (Observation 2)

Question 3: In the case that DMs have limited knowledge about other methods

(a) how does one to determine the validity of the decision made by the DMs using the method they are familiar with

(b) is there a decision making formulation that allows DMs to select and utilize the most appropriate method to solve their decision problems? (Observation 3)

Question 4: Can advice be given if no method in the method pool is suggested for the given problem? (Observation 4)

Question 5: Can the method selection be handled in an efficient manner? (Observation 5)

To answer the questions above, the hypotheses below are made:

Hypothesis 1: A decision making method can be fully represented by its associated characteristics which are able to be identified using the developed approach. (Question 1)

Hypothesis 2: It is able to develop an algorithm to rank the decision making methods based on the problem under consideration. The selected method has the highest appropriateness to solve the given problem. (Question 2)

Hypothesis 3: A decision making formulation is required to allow the DMs to select the most suitable method among the candidate methods and then guide them to obtain the final decision even if they have limited knowledge about the selected method. (Question3)

Hypothesis 4: If no method is suggested for the given problem, advices should be given, such as finding an existing method with the capability to solve the problem at hand or combining the methods in the method pool to produce an advanced method. (Question 4)

Hypothesis 5: The proper design of a decision making advisor system can efficiently facilitate the decision making process, from selecting the appropriate method to making the final decision. (Question 5)

CHAPTER II

DESIGN AND DECISION MAKING

Design, in general, and engineering design, in particular, is a process that starts from a set of requirements, and then utilizes scientific and technical knowledge to produce a solution to a human problem. The requirements often emerge from a customer's needs which may be brought about by scarcity, technology or a change in life style. According to Asimow [1962], design is "a purposeful activity directed toward the goal of fulfilling human needs, particularly those which can be met by the technological factors of our culture." In a design process, the available information and techniques are utilized to establish and define "solutions to and pertinent structures for problems not solved before, or new solutions to problems which have previously been solved in a different way" [Blumrich, 1970]. In order to increase the probability of success of a design, both mathematical analysis and practical experience are employed in the design process, which often support the designer or engineer to make wise decisions leading to the optimal design. Modern system design usually uses this is pretty vague rigorous techniques which follow some systematic processes to reach the final design solution.

2.1 Design Process

There are a number of models that describe the stages of the engineering design process, each with associated design methods and data requirements. Among these models, the one proposed by Dieter [2000] is a good representative of the engineering design process. As illustrated in Figure 6, the design process is divided into conceptual design,

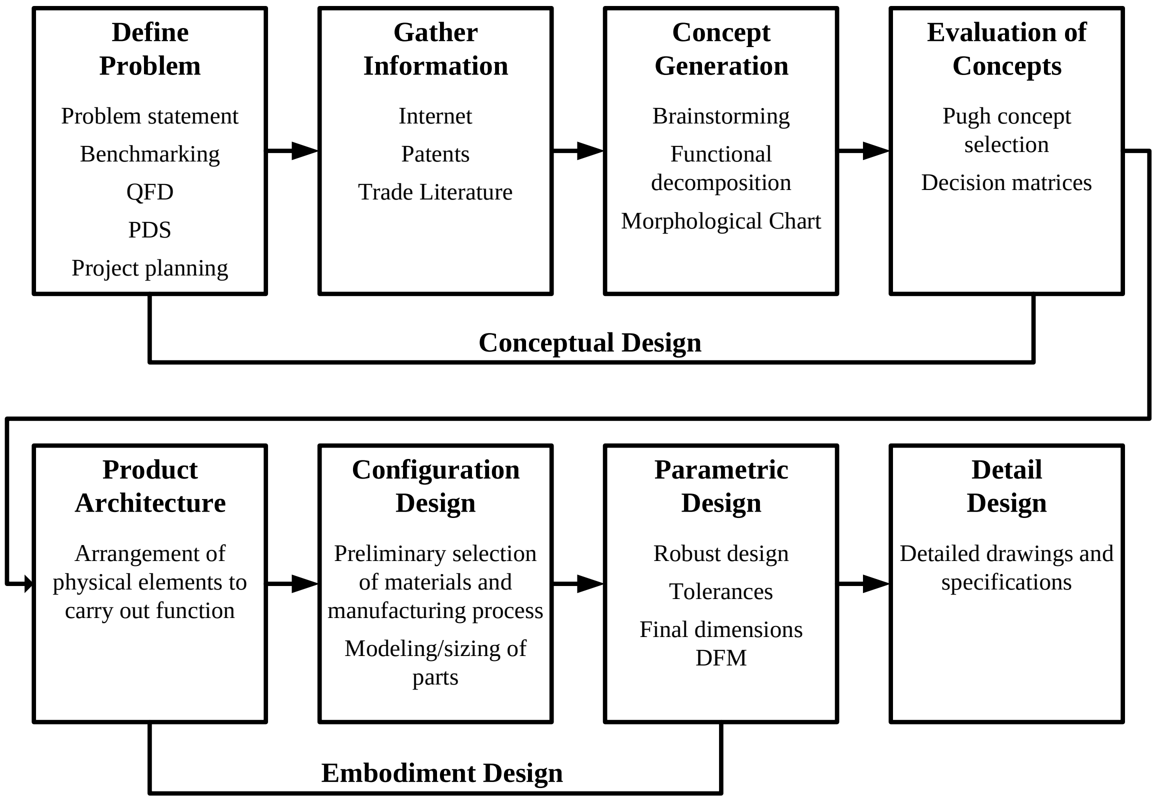


Figure 6: The Engineering Design Process [Dieter, 2000]

embodiment design and detail design, each of which has one or more steps. In the conceptual phase, the first step is defining the problem, where the Customer Requirements (CRs) are translated into Engineering Characteristics (EC's). The Quality Function Deployment (QFD) technique is applied in this step. Then the necessary information needs to be gathered to generate feasible concepts which have the potential to meet the customer requirements. In this step, some brainstorming tools, such as a morphological matrix, are usually used. Since the generated concepts have different degree of viability, the one which can best satisfy the customer requirements will be selected for embodiment design. Embodiment design is concerned with arranging physical elements of the product to carry out its function, selecting materials and manufacturing process, and conducting a parametric design study where robust design and tolerance design are completed. In the detail design stage, details, such as drawings and past specifications, are brought together to ensure the manufacturability of the design.

Aircraft design is the application of the engineering design process, with multiple disciplines involved. A three-tiered design process is generally accepted for aircraft design, which consists of conceptual design, preliminary design, and detail design. Before the conceptual design proceeds, design requirements need to be well defined, which is critical since a poorly defined problem often results in a misleading design solution. As an aircraft is a complex system, multiple disciplines analysis and optimization inevitably occur in the design process. Fielding proposed a design spiral adapted from Haberland's work, as shown in Figure 7, which is helpful to illustrate the

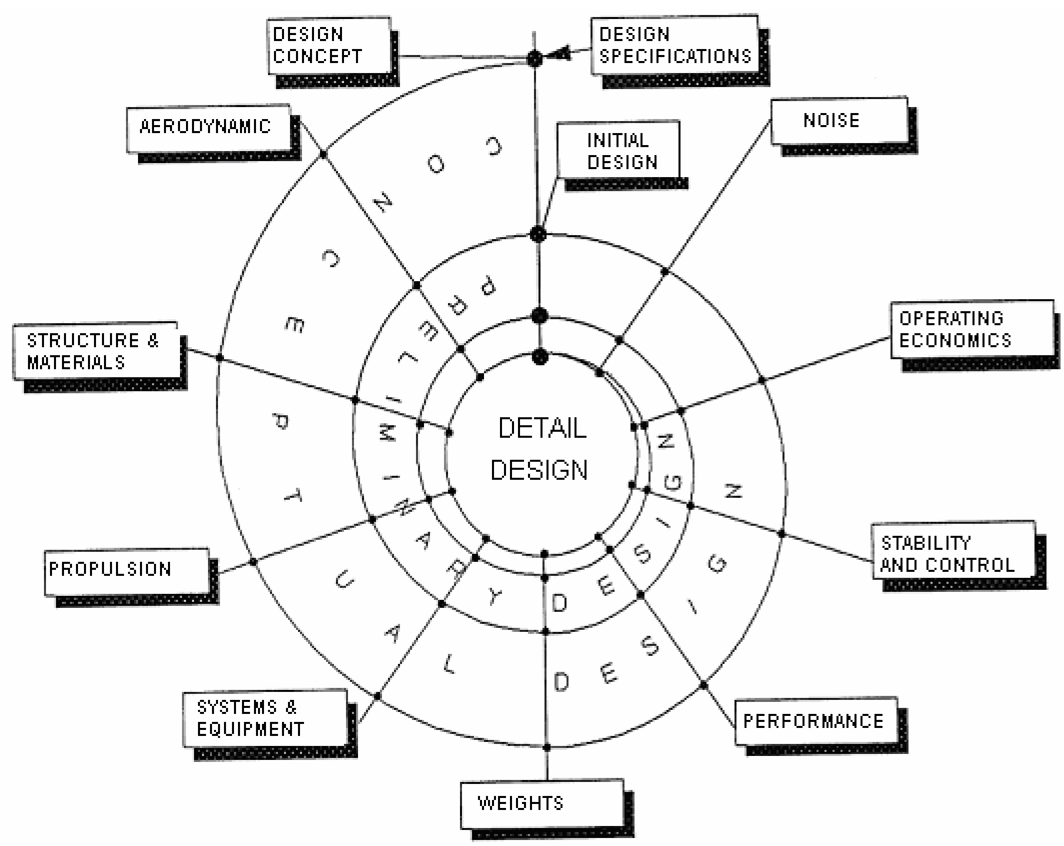


Figure 7: Aircraft Design Spiral [Fielding, 1999]

design activities in the aircraft design process. These design activities are iterative through the design process, converging to the detail design stage, and ultimately ending in the manufacture and operation of the aircraft.

Systems design is defined as “the application of scientific and engineering knowledge to produce a functional prototype model (which) defines the basic product/process design characteristics and their initial settings” [Noble and Tanchoco, 1993]. The goal of systems design is to produce design concepts that best satisfy customer requirements which are often referred to as design objectives. In systems design, particularly in modern aircraft systems design, more and more attention is paid to the conceptual and preliminary design stages to increase the probability of choosing a design that will be successful that is, both technologically feasible and economically viable. The decisions made during these early stages have a critical impact on the final design solution since poor conceptual design will lead to more changes happening in late design stages, which will result in dramatic increase in cost. To prevent costly re-designs as much knowledge as possible should be made available at the early stages of design. Probabilistic design is a suitable approach that can bring knowledge to the early design stages, capture the uncertainty effects, and provide suitable confidence in the results obtained. Probabilistic aircraft system design process for conceptual and preliminary design is illustrated in Figure 8 [Li et al., 2004]. This design process was adapted from a generic design methodology referred to as the Technology Identification, Evaluation and Selection (TIES) method [Kirby and Mavris, 2000; Mavris and DeLaurentis, 2000a] which encompasses a feasibility and viability examination process, explained in numerous technical publications. An approach called the Unified Tradeoff Environment (UTE), which uses combined sets of Response Surface Equations (RSEs) to visualize sensitivities of key design parameters to mission requirements, concept design variables, and technology k-factors was also explored in this method.

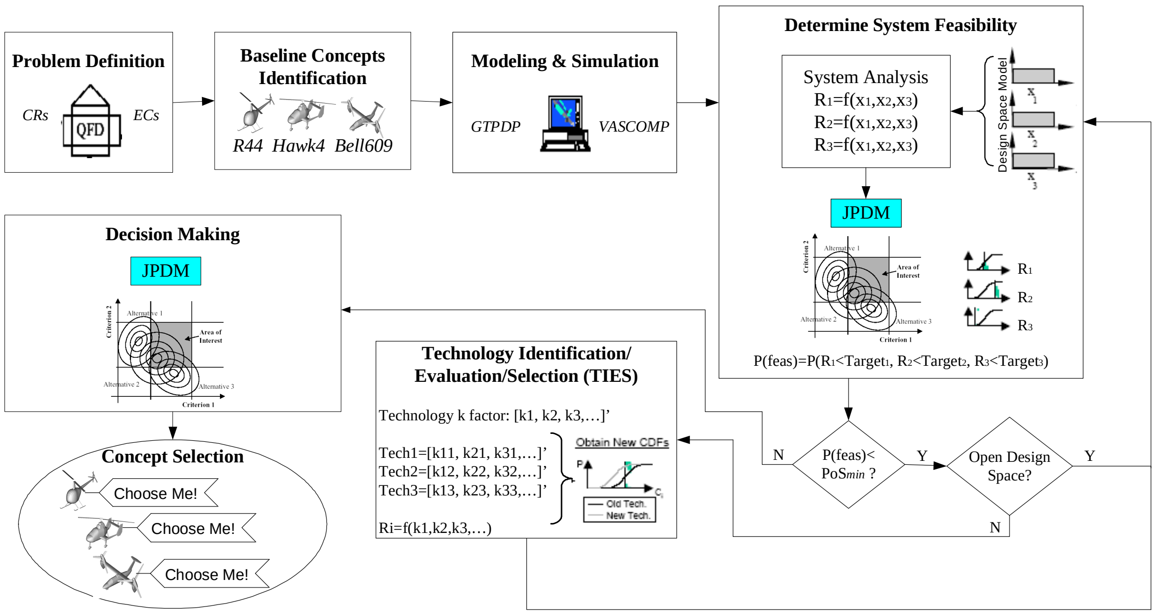


Figure 8: Modern Aircraft Systems Design Process [Li et al., 2004]

In the process presented in Figure 8 for the determination of system feasibility and concept viability, the Joint Probability Decision Making (JPDM) [Mavris and DeLaurentis, 2000b], a probabilistic MCDM technique, was employed. The first step in this method is problem definition, where a set of requirements is well defined in responsible to a customer's needs. Then a baseline concept needs to be identified as a starting point based on which the further design is carried out. In the conceptual stages of aircraft design, a rapid assessment is desired to perform the tradeoffs. To effectively facilitate the rapid assessment, a modeling and simulation environment is necessary in which some sizing and performance programs are used to help the analysis process. Further, with the introduction of uncertainty, a UTE is generated to explore the design space using statistical methods including Design of Experiments (DoE) and Response Surface Equations. Subsequently, technical feasibility is investigated based upon the customer requirements and environment or operational constraints. In this key step, the JPDM technique is used to determine whether an expensive investigation of new technologies is necessary. If the design concept is not feasible, three options are available to improve the feasibility without technology infusion: relax the active constrains, open the design space or change the concept. Usually, in a new design for advanced concept development, the options above are not allowed to be chosen because of design limitations, and therefore technologies need to be identified and infused to improve system feasibility. Finally, the most viable concept selected from the feasible solutions is obtained using the JPDM technique, and then is sent to the next design step.

2.2 Uncertainty in Systems Design

In the previous discussion, the concept of uncertainty is identified as a key factor which has to be captured to deal with decision making in modern systems design. In general, uncertainty means two or more outcomes are possible. In the context of systems design, this implies that "multiple system responses are possible when variability associated with design information (i.e. requirements, concepts, and technologies) is propagated to the system level" [Baker, 2002]. The existence of uncertainty in systems design results from the facts that most assumptions made about the operational environment of the system are uncertain, and new technologies used often have readiness/availability issues. In addition, computer model is usually not accurate enough to reflect the reality so introduces further uncertainty to the design solutions. This lack of certainty about the system responses makes decision making in systems design one of the most challenging tasks faced by decision makers.

It is apparently that uncertainty is the greatest in the early design stages as shown in Figure 9. In this figure, “knows” means certainty, “know-unknowns” indicates risk and “unknown-unknowns” signifies uncertainty. Thus, three decision making models are classified with respect to these states of knowledge. They are decision making under certainty, decision making under risk and decision making under uncertainty. Decision making under certainty implies that the system outcome is known and occurs with a probability of 1 (knows). Decision making under risk implies that the system has multiple possible outcomes and the probabilities for the occurrence of the outcomes are known (known-unknowns). Decision making under uncertainty implies that the system has multiple possible outcomes but the probabilities for the occurrence of the outcomes are unknown (unknown-unknowns). Garvey summarizes the distinction saying, “In a situation that includes favorable and unfavorable events, risk is the probability an unfavorable event occurs. Uncertainty is the indefiniteness about the outcome of a situation. We analyze uncertainty for the purpose of measuring risk.” [Garvey, 2000]

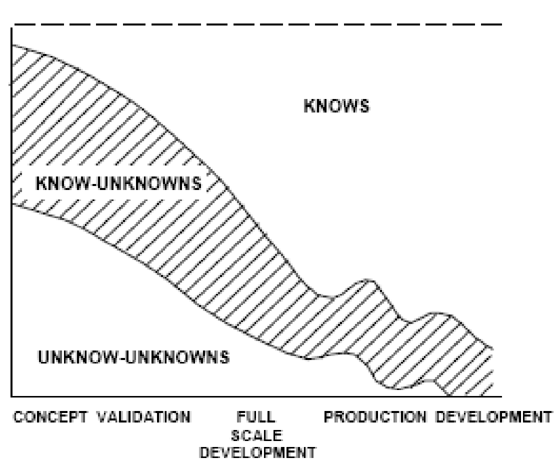


Figure 9: Risk and Uncertainty Greatest at Front End [U.S. Army, 1990]

In order to handle the risk and uncertainty which are the greatest during concept development and validation, a probabilistic design approach needs to be employed to produce robust and/or optimal design solutions. As a result, design decisions need to be made using probabilistic decision making techniques, and some of them will be described in Section 2.6.

2.3 Design is a Decision Making Process

In general, the performance attributes of the design solution are needed to meet some functional requirements and constraints. For example, to design a large commercial aircraft, multiple requirements, such as requirements on aerodynamics, propulsion, structure and noise, need to be satisfied. Usually, the design that best satisfies one individual requirement does not have the best performance on other requirements (Figure 10) [Kroo, 2004]. That is, typically there is no a design that has the best performance on all the requirements. As a result, tradeoffs need to be done when the requirements are simultaneously taken into account. This usually involves decision making activities, such as determining the preference information of the customer, establishing the decision rules of evaluating the alternatives, and selecting the "best" solution among the alternatives. Sen and Yang [1998] point out that decision making in engineering design "can be helpfully visualized as a collection of activities that relate to choice in the context of competing technical or functional requirements". Dieter also argued that "Thus, decision making is essentially part of the design process and the fundamental structure in engineering design."

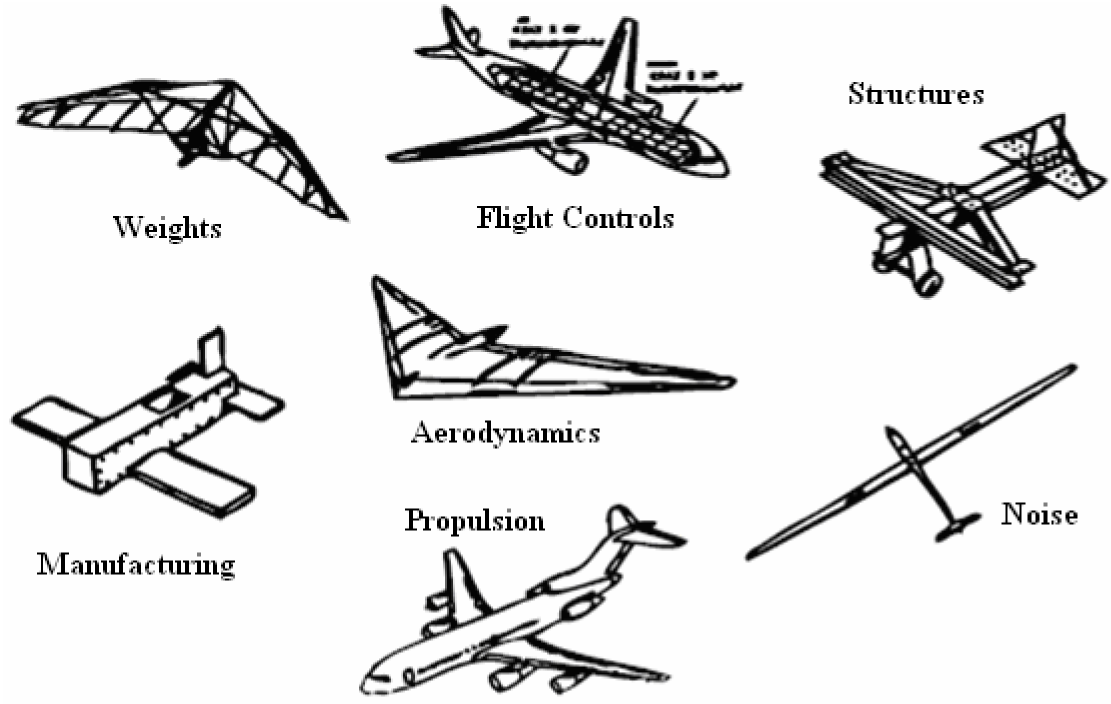


Figure 10: One can only make one thing best at a time [Kroo, 2004]

2.3.1 Decision Making in Systems Design

Decision making is "the act of making up one's mind, judging, or reaching a conclusion about something" [Webster's New World Dictionary of American English, 1996]. This definition from Webster's dictionary does not clearly indicate the relation between decision making and design. However, more and more emphasis is given to the decision making in engineering design and there is an emerging understanding that design is a decision making process.

Hazelrigg [1996] argued that "To be sure problem solving capabilities are important in engineering. Yet, ... problem solving is not the principal activity of engineering; rather it is decision making". This emphasis on decision making is supported by the statement of other researchers. Tate [1999] asserted that "In design, decision making is most important. This is because designers must make many types of decisions: for example the choice among various alternatives in order to create or select the best design, (or) the development of a set of suitable requirements". As Baker [2002] noticed, Howe [2000] clearly stated the role of decision making in engineering design:

> *Engineering design is a non-unique iterative process, the aim of which is to reach the best compromise of a number of conflicting requirements. Whether the need is for a totally new item or for a development of an existing one, the design procedure commences with an interpretation of the requirements into a first concept. This is essentially a synthesis process which involves decision making. Once the first concept has been derived it can be analyzed in the context of the requirements. The concept is refined by an iterative synthesis/analysis/decision-making sequence until an acceptable solution is achieved.*

This argument recognized strong connections between design and decision making and implied an unexaggerated conclusion: design is a decision making process [Hazelrigg, 1996]. This statement is championed by the fact that decision making permeates through the design process and is at the core of all design activities. Problem definition, for example, involves deciding what the customer requirements are and how to define constraints and targets. Other design phases such as alternative generation, design space exploration, and concept selection, rely heavily on or are pure decision-making processes [Li et al., 2004]. Furthermore, the selection of design parameters, which is a basic design fulfillment, represents the decision.

2.3.2 Decision Making in Early Design Stages

It has been recognized that, in an engineering design, most of the changes occurring in early design stages will lead to high quality design with significantly reduced cycle time [Sullivan, 1986]. On the contrary, if most of the changes happen in late design stages, e.g. re-design, the cost of making change will dramatically increase since design freedom is highly limited in these stages. Figure 11 shows the comparison of traditional serial design approach and concurrent engineering design approach with respect to a design time line [DoD, 1996]. From this figure, one can see that the cost increases exponentially when the changes happen at the late design stage. Therefore, as many changes as possible should be completed early in the design time line. To prevent the costly re-designs, as much knowledge as possible should be made available at the early stage of a design and the requisite changes should be accomplished before the cost is locked in. This paradigm shift of bringing knowledge to the early design stages to increase design freedom and reduce cost is illustrated in Figure 12, which is interpreted in numerous technical publications while Refs [Mavris et al., 1998], [DeLaurentis, 1998], and [Mavris and DeLaurentis, 2000a] provide the best overall perspective.

Therefore, as briefly stated before, more and more attention is paid to the conceptual and preliminary design stages to increase the probability of choosing a design that will be successful. The decisions made during these design stages, including identifying customer's requirements, determining the attributes of interest, and selecting analysis tools, play a critical role in the design process. They are the guidance and basis that subsequent design decisions rely upon, and have an important impact on the final design solution. Therefore, these decisions in the early design stage need to be made wisely.

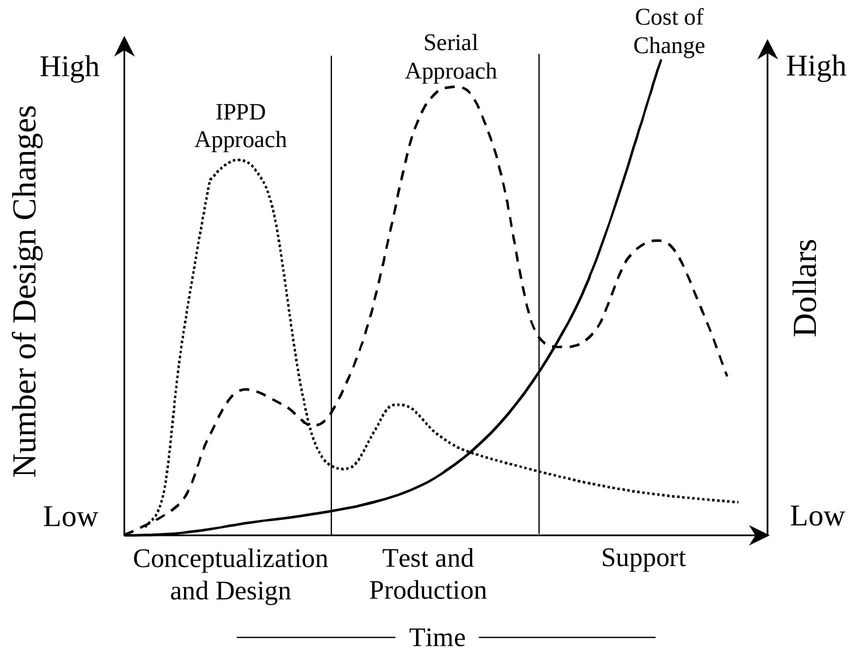


Figure 11: Traditional Serial Approach vs. Concurrent Approach [DoD, 1996; Kirby, 2001]

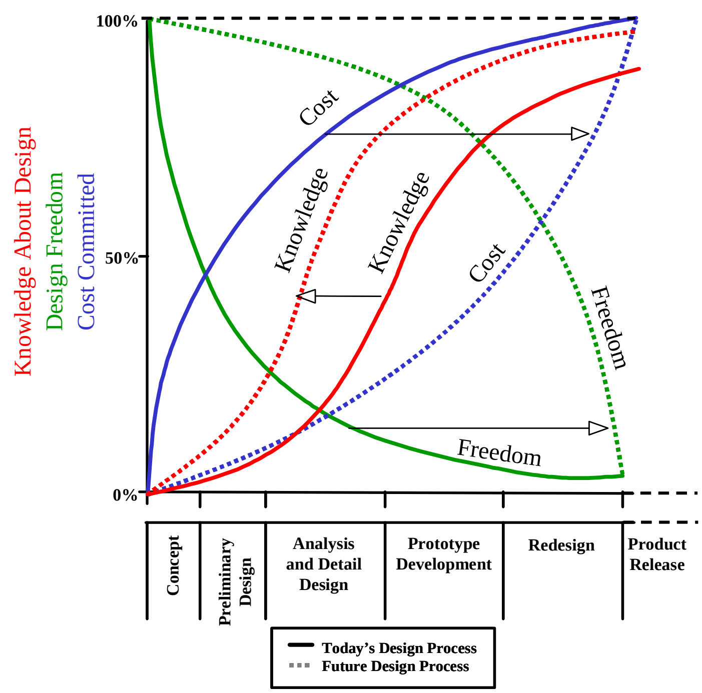


Figure 12: Cost-Knowledge-Freedom Relations [DeLaurentis, 1998; Mavris et al., 1998; Mavris and DeLaurentis, 2000a]

With knowledge brought forward in the design time line, designers are able to make more educated decisions. Integrated Product and Process Development (IPPD), illustrated in Figure 13, encourages moving information forward in the design process. IPPD is concerned with upfront activities in the early design phases and allows the designers to decompose the product and process design trade iteration through a system's life cycle [Marx et al., 1994]. The implementation of IPPD "reorders decision making, brings downstream and global issues to bear earlier and in concern with conceptual and detailed planning" [DoD, 1996], so it can allow the designer to make a better decision in the early design stages.

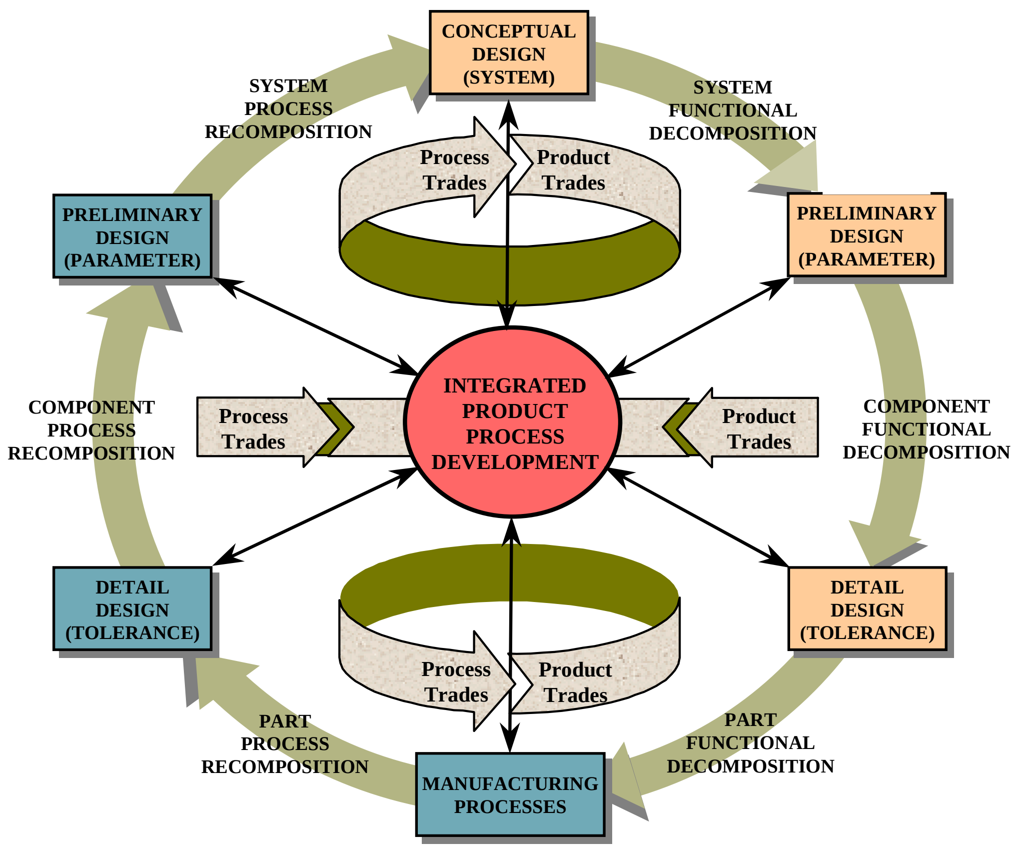


Figure 13: Hierarchical Process Flow for Large Scale System Integration [Marx et al., 1994]

2.4 Decision Making Process

In early engineering design, the decisions made during the design process are mainly based on the designer's intuition, that is, his experience, values, and preferences. With the complexity of design problems increasing, decision making is almost an impossible task for the individual DM to manage. For example, it is usually hard for a DM to make a selection among three alternatives with respect to six attributes of interest by himself. To facilitate DMs to make proper decision for complex problems, various decision making methods and techniques have been developed in the past decades, and this led to the emergence and flourishing of a new scientific field known as Decision Science in the beginning of 1970's [Matsatsinis and Y., 2003].

It is widely accepted that a good problem formulation plays an important role in determining the success of the final solution. Many researchers have made great efforts to formulate the decision problem, and tried to come up with a model to correctly represent the decision making process. Among them, the one proposed by Simon is particularly famous model, in which decision making is divided into three distinct phases: intelligence phase, design phase, and choice phase. Figure 14 shows the decision making process proposed by Simon.

In the intelligence phase, the goal is to define the problem and collect the necessary information. The DMs need to explicitly identify the customer's requirements, problem constraints, and decision criteria. The characteristics of the problem also need to be defined so that an appropriate decision making technique can be selected to solve the problem.

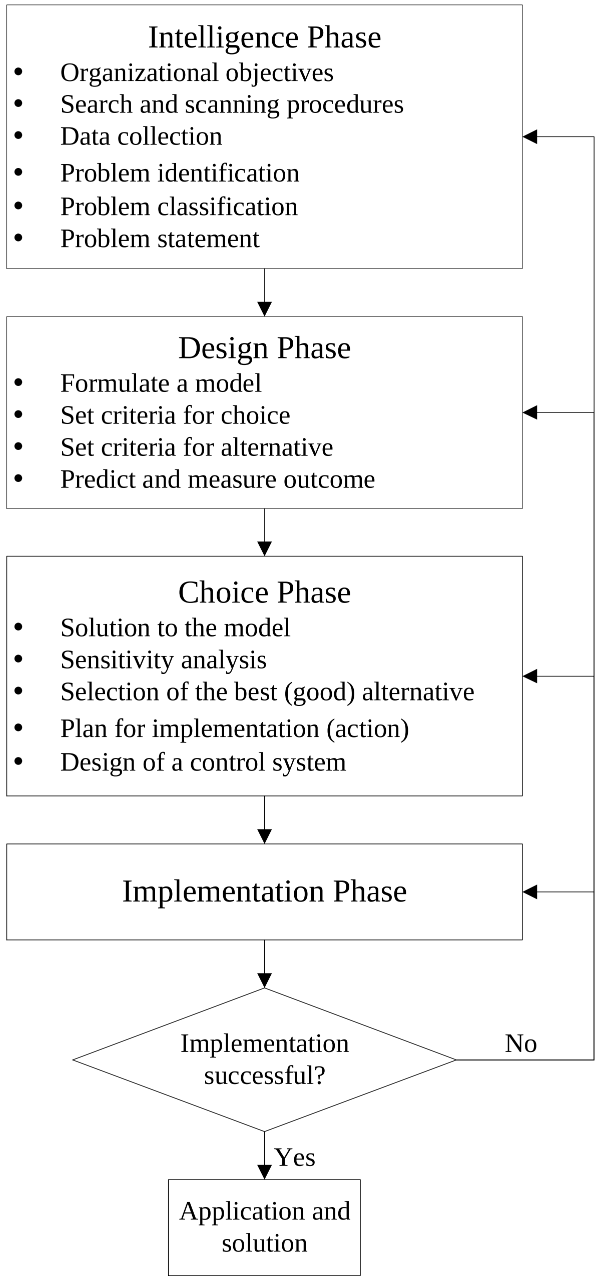


Figure 14: Generic Decision Making Process [Simon, 1960; Sprague and Carlson, 1982]

The design phase is mainly focused on molding the problem to efficiently represent the status of the problem. An investigation needs to be done to figure out which design alternatives are available for further selection. If there are no existing alternatives, design and analysis will be performed to generate the complete set of alternatives. The generation of design alternative involves design activities such as design space definition, design space exploration, and feasibility evaluation.

The choice phase is the most significant in the decision making process. In this step, the best alternative will be selected based on the priorities of the criteria defined in the intelligence phase. An appropriate decision making method or technique needs to be selected first, because different methods have different representations of the designer's preference information, various analytical algorithms and decision rules, and will suggest different "best" solutions. After the decision making method is selected, searches, evaluations and choices may need to be carried out by following the problem-solving procedure of the selected method, and the "best" solution can be obtained based on the evaluation of the given criteria.

Usually, it is accepted that the implementation of the decision is also included in the choice phase. Because of its importance and relative independence, it is considered as a separate phase in the decision making process. In this phase, the proposed solution is implemented and the result is evaluated. If the results meet the requirements, the solution will be directly applied. Otherwise, one needs to diagnose the problems that may have happened in the preceding phases, and revision and modification should be performed until a satisfactory result is obtained.

2.5 Multi-Criteria Decision Making

Each decision making activity falls into one of two categories. The first is decision making based on DM's brainstorming, experience, or intuition. In this category, DMs come up with a final decision in an empirical way without utilizing sophisticated decision making techniques or methods. In the 2nd category, for more complex problems, decisions are made with the aid of some structured decision making techniques or methods which have an analysis model and step by step problem solving procedure to be followed. These structured decision making methods often employ analytical or numerical technique to form a model which is able to facilitate the decision making process. In such a scenario, DMs reach the final decisions by firstly formulating a decision problem using the analysis model of the method and then applying the problem solving procedure to the formulated problem. The study presented in this document is concerned with decision making problems in the second category.

2.5.1 What is Multi-Criteria Decision Making?

Almost every design problem in modern engineering design inherently has multiple criteria which need to be satisfied. It is often the case that good values of some criteria inevitably go with poor values of others, so that the best design is always a compromise in some sense. In order to find the best compromise design solution, designers are required to take all the metrics of interest into account concurrently when making decisions. For example, when designing a large commercial aircraft, designers will have to consider reducing cost, increasing performance and minimizing emissions. As a result, a tradeoff has to be done, and compromise becomes an essential part of the MCDM process.

Typically, in order to solve an MCDM problem, some necessary factors need to be known beforehand: 1) the well defined, measurable criteria, 2) the preference information on the criteria, 3) the alternatives and 4) a disciplined, repeatable, transparent decision making method. The criteria can be thought of as the measure of performance for an alternative, such as speed and payload of an aircraft concept, and can be identified by analyzing the customer's requirements. The criteria need to be well defined so that the customer's requirements can be fully represented. The alternatives are the candidates among which the "best" solution is selected. They may be the concepts that are already existing, or need to be generated in the design process. Since the criteria do not have same priority to the customer, the preference information on the criteria should be defined. Relative weights, which are assigned beforehand or calculated, are a popular way to represent the preference information. There are other ways to represent the customer's preference, which will be explained in the next section. A set of appropriate alternatives has critical impact on the final solution because the final solution is one of the elements of this set. Usually, the alternatives are non-dominated solutions to the decision making problem. The decision making method is usually a systematic process which employs some decision rules and algorithms to formulate the decision problem and provide guidance to the DEM to reach the final decisions. Different decision making methods have their own advantages and disadvantages, and are suitable to solve one type of decision problem, so the selection of an appropriate method should be carefully carried out before the decision making process proceeds.

In general, a MCDM problem can be mathematically represented by Equation (1), where X is the n-dimensional vector of design variables defining a design, $f_c(X)$ (c = 1, 2, …, k) is the value of the c-th criterion at X. The problem is subject to the inequality constraints $g_j(X)$ (j = 1, 2, …, m_1), equality constraints $h_l(X)$ (l = 1, 2, …, m_2) and side constraints which all together define the design space Ω. The design alternatives in the design space are feasible solutions. For each point X in the design space, there is a corresponding k-dimensional attribute vector $f(X)$ in the criteria space. That is, the design space can be mapped into the criteria space defined by $S = \{ f(X) | X \in \Omega \}$. The objective of the MCDM problem is to find a design $X \in \Omega$ that can minimize the aggregate function $F(X)$ which is a function of criteria $f_c(X)$ (c = 1, 2, …, k). X^* is called an optimal solution iff $X^* \in \Omega$ and $f(X^*) \leq f(X)$ for any $X \in \Omega$. If X^* exists, it will be the design solution for the MCDM problem. In reality, the attributes of a product are usually conflicting so a design solution intending to improve an attribute may impact another attribute in the opposite direction. For example, to minimize the gross weight of a commercial aircraft, the use of composite material is considered as the solution. However, the cost, which is expected to be minimized too, will be increased by taking this solution. Therefore, in MCDM tradeoff has to be done among the criteria, and finding the compromise solution is the aim of the MCDM.

$$
\begin{aligned}
&Minimize: \quad F(X) = f\left[f_1(X), f_2(X), \ldots, f_k(X)\right] \\
&Subject \;\; to: \quad X \in \Omega \\
&\qquad\qquad \Omega = \left[X \left| \begin{array}{l} g_j(X) \leq 0, \;\; j = 1, \ldots, m_1 \\ h_l(X) = 0, \;\; l = 1, \ldots, m_2 \\ x_i^l \leq x_i \leq x_i^u, \; i = 1, \ldots, n \\ X = [x_1, x_2, \ldots, x_n]^T \end{array} \right. \right]
\end{aligned}
\tag{1}
$$

The MCDM techniques are broadly classified into two types: Multi-Attribute Decision Making (MADM) and Multi-Objective Decision Making (MODM) techniques. MADM includes the methods that select the “best” compromised solution from a small number of alternatives based on prioritized attributes of those alternatives. MODM relates to techniques that synthesize a set of designs that are required to meet a list of requirements. Briefly, MADM deals with the concept selection problem while MODM handles the design or synthesis problem. The relationship among MCDM, MADM and MODM is presented in Figure 15 [Sen and Yang, 1998].

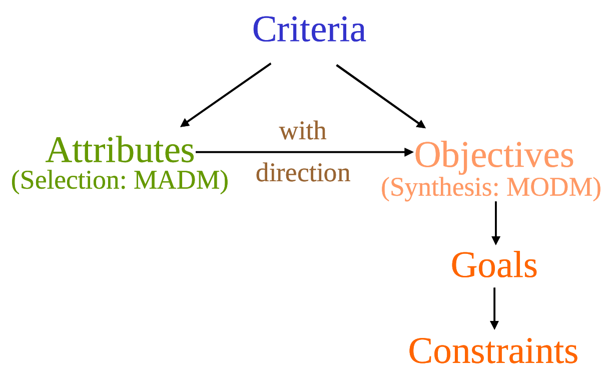


Figure 15: Multiple Criteria Decision Making [Sen and Yang, 1998]

2.5.2 Why Multiple Criteria in Aerospace Decision Making?

Aerospace systems are very complex, having interacting disciplines and technologies. The requirements for designing a successful system come from various stakeholders such as the passengers, pilot, maintenance crew, airline, manufacture, and so on. These stakeholders have different requirements based on their own needs. For example, passengers think safety is the first need to them, pilots consider the handling quality is the most importance issue that should be taken care of, while airline is the most concerned with the overall operating cost. Figure 16 shows the design environment for aerospace system design. This complicated design environment indicates that aerospace system design is multi criteria in nature. Therefore, in order to achieve the success of a design, the stakeholders' needs have to be all simultaneously taken into account. The needs include reducing costs, increasing profit, performance, environmental friendliness, and quality. As a result, to produce the best design concept, the DMs are involved in balancing the multiple, potentially conflicting attributes/criteria, and transforming a large amount of customer supplied guidelines into a solidly defined set of requirement definitions. This implies that the aerospace system design is essentially a MCDM process.

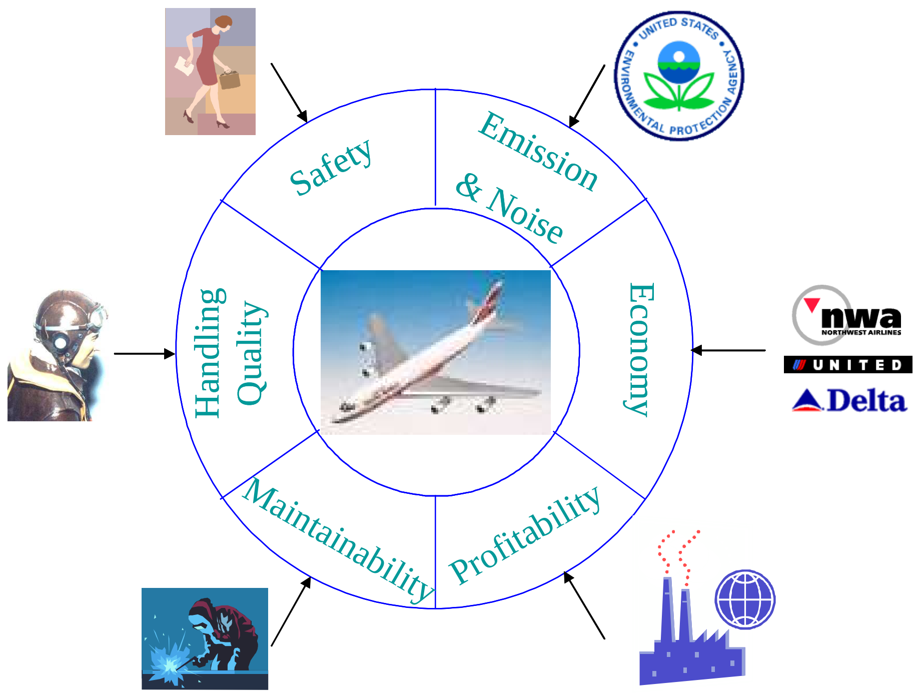


Figure 16: Multi-Criteria Decision Making in Aerospace Systems Design

2.6 Multi-Criteria Decision Making Methods

MCDM addresses decision making with multiple, possibly conflicting criteria that simply indicates attributes or objectives. MCDM problems “involve the selection of the ‘best’ alternative from a pool of preselected alternatives described in terms of their attributes” [Hwang and Masud, 1979]. These preselected alternatives are the solutions that can be best described by the concept of a Pareto frontier.

2.6.1 Pareto Frontiers

Since good values of some criteria inevitably go with poor values of others, the goal of the MCDM is to find the "best" compromise solution which has best overall performance of satisfying all the attributes. This "best" solution can be obtained from a set of the design alternatives referred to as the Pareto-optimal solution.

Pareto Optimality: The Pareto-optimal solution is defined as the solution X^* iff no $X \in \Omega$ exists such that $f_i(X) \le f_i(X^*)$ for all $i \in \{1,2,\cdots,k\}$, and $f_j(X) \le f_j(X^*)$ for at least one j, $j \in \{1,2,\cdots,k\}$.

The definition of the Pareto optimality indicates that there is no other feasible solution in the design space has the same or better performance than the Pareto optimal solution considering all criteria, and the Pareto optimal solution does not have the best performance in all criteria [Zeleny, 1982]. It is clear that Pareto-optimal solution is a non-dominated solution which is "achieved when no criteria can be improved without simultaneous detriment to at lease one other criterion" [Bandte, 2000]. The locus of the Pareto optimal solutions is known as Pareto frontier. A two-dimensional Pareto frontier is illustrated in Figure 17 for "smaller is better" criteria.

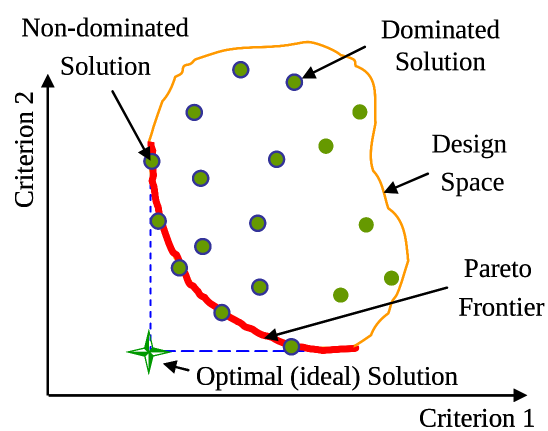


Figure 17: Two-dimensional Pareto Frontier

In general, the "best" compromise solution is selected from the Pareto frontier. Therefore, it will increase computation efficiency if the Pareto optimal solutions are first selected as candidates and then the final solution is chosen from them. However, it has been recognized that the number of non-dominated solution will increase dramatically with the number of the criteria [Deb, 2001; Bore and Mavris, 2004]. As a result, for problems with a large number of criteria, it is not worth the computational effort to find the non-dominated solution first since it will be difficult to resolve all the non-dominated solutions. Though this problem exists for large decision making problems, it does not stop Pareto frontier from being an desired concept in the realm of MCDM.

In order to deal with the more complex decision problems, researchers have focused in the past decades on developing advanced methods to facilitate the decision making process. Currently, there are over 70 MCDM methods that have been proposed. Some widely used MADM and MODM methods will be briefly explained in the following sections.

2.6.2 MADM Methods

MADM methods are developed to handle concept selection problems. In this class of problems, the "best" solution is determined from a finite and usually small set of alternatives. The selection is performed based on the evaluation of the attributes and their preference information. In the decision making process, many MADM techniques use decision matrix (or goal achievement matrix) D, shown in Equation (2), to describe the states of the attributes of each alternative. In this equation, element y_{ij} represents the value of attribute j with respect to design alternative i.

$$D = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1k} \\ y_{21} & y_{22} & \cdots & y_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{nk} \end{bmatrix} \tag{2}$$

Another important concept is the comparison matrix, which represents the DM's preference information. Equation (3) shows an n by n comparison matrix M, in which the element mil represents the relative importance of alternative i over alternative l with respect to attribute j. Therefore, for a decision problem which has n alternatives, k attributes, there will be k n by n comparison matries.

$$M_j = \begin{bmatrix} 1 & m_{12} & \cdots & m_{1n} \\ m_{21} & 1 & \cdots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & 1 \end{bmatrix} \text{where } j = 1, 2, \ldots, k \tag{3}$$

Generally, MADM methods can be classified into compensatory and non-compensatory methods based on the treatment of the attribute information. The compensatory methods allow trade-offs between criteria, assigning a number to each multidimensional representation of an alternative. The non-compensatory methods do not permit the trade-off between criteria, i.e. one unfavorable criterion value cannot be offset by reducing a favorable value of another criterion.

Several MADM methods are listed in Figure 18 [Sen and Yang, 1998], of which some typically used methods are briefly explained here and more detailed descriptions about these methods can be found in Appendix B. Sen and Yang also proposed a taxonomy in the form of a tree diagram to help decision maker select suitable decision method, as shown in Figure 2.

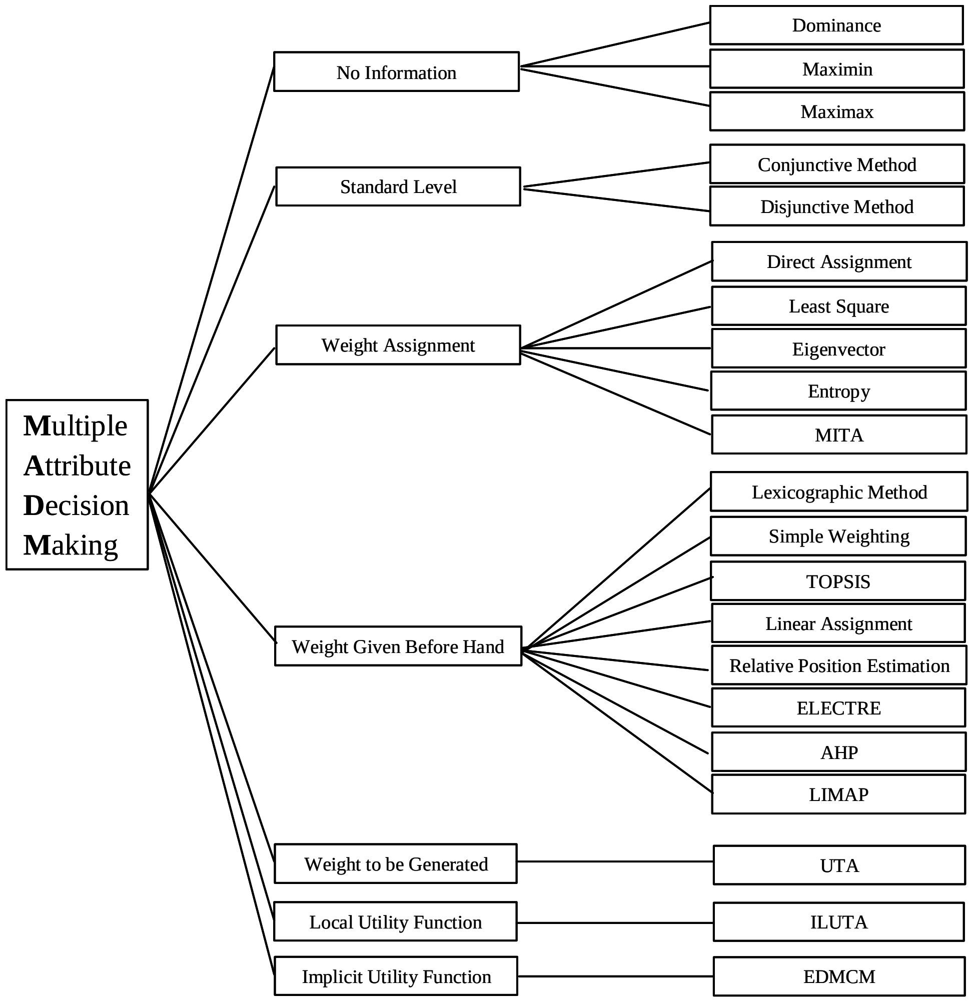


Figure 18: Classification of MADM Methods [Sen and Yang, 1998]

2.6.2.1 Preference Representation

Preference is a concept that describes the DM's predisposition in favor of one attribute over another when making choice between alternatives, based on the satisfaction or utility they provide. For instance, when shopping for a new car, one customer think the reliability and fuel consumption are the most important things that a car should have, while another buyer may consider the safety and performance are the desired characteristics that he wants. The difference in preferences will end up with diverse decisions: the first customer considers Toyota Corolla is the best choice while the latter may think BMW 330i is his desired car. This implies that preference is one of the important factors that have a critical impact on the final decision and needs to be carefully formulated.

There are several approaches to represent the DM's preference information, including weight assignment techniques, loss function, utility function and class function.

Weight Assignment Techniques

Weight assignment techniques are the widely used techniques for representing the preference information since they are easy to understand and simple to use. Typically each attribute is assigned a relative weight that represents its importance comparing with other attributes. The higher that attributes assigned weight, the more important that attribute is considered to be. Three typical weight assignment techniques are often adopted in the decision making methods: direct assignment, eigenvector method and entropy method.

Direct assignment may be the simplest way to formulate preference information. In this technique, one is allowed to "directly evaluate the relative importance of one attribute over others using certain evaluation standard" [Sen and Yang, 1998]. Usually this can be accomplished by an experienced decision maker using a 10-point scale with the definition that 0 is extremely unimportant and 10 is extremely important. This technique is popular due to its simplicity, however it is not accurate enough to represent the DM's preference therefore it is not an appropriate technique for the decision problem whose solution is sensitive to the DM's preference information.

The eigenvector method is an analytical way to elicit relative importance. The preference information can be obtained from solving an eigenvalue function shown in Equation (4).This method uses pairwise comparison between attributes, represented by a comparison matrix *M* defined by Equation (3). The weights of attributes can be calculated as the normalized eigenvector *W* as shown in Equation (4), where λ_{max} is the maximum eigenvalue of the matrix *M*.

$$MW = \lambda_{\max} W \tag{4}$$

To use this method to calculate relative importance of the attribute, all pairwise comparisons of M should be consistent. However, the comparisons normally are not consistent, especially for large comparison matrix. Saaty [Saaty, 1988] suggested an algorithm that starts from an initial weight vector and uses the concept of consistency index to obtain the final weight vector iteratively. This algorithm first produces a comparison matrix with high degree of consistency and then uses it to calculate the weight vector utilizing the eigenvalue function shown by Equation (4).

The entropy method provides another way of eliciting and representing preference information, especially for the case where the decision matrix, defined in Equation (2), is available. Assume a decision matrix of a MADM problem is represented by Equation (2), and then the best weights of the attributes are given by Equation (5).

It is worth noting that the value of w_j "reflects the degree to which the j^{th} attribute contributes in discriminating over the set of alternatives concerned" [Sen and Yang, 1998]. This can be verified by the fact that the weight of an attribute is small when all the alternatives have similar outcomes on the attribute.

$$
\begin{aligned}
& w_j = \frac{d_j}{\sum_{i=1}^{k} d_i} \quad \text{for all } j \\
& \text{where} \quad d_j = 1 - E_j \quad \text{for all } j \\
& \qquad E_j = -\frac{1}{\ln(n)} \sum_{i=1}^{n} p_{ij} \ln p_{ij} \quad \text{for all } i, j \\
& \qquad p_{ij} = \frac{y_{ij}}{\sum_{i=1}^{n} y_{ij}} \quad \text{for all } i, j
\end{aligned}
\tag{5}
$$

Loss Function

Taguchi’s Loss Function (LF) concept originated from the robust design method, which is a systematic approach to improve the product quality and reduce cost by minimizing the sensitivity to uncontrollable, or noise, factors. The loss function establishes a financial measure of the customer dissatisfaction with a product's performance as it deviates from a target value. That is, the LF measures the product quality in terms of the deviation and variability. The further the product attribute is from the target value or the higher the variability it has, the poorer its quality and the more loss it creates to society. There are three types of LF’s: larger the better, smaller the better, and nominal the best. The mathematical models of these LF are shown in Equations (6), (7) and (8), respectively, and visualized in Figure 19.

Larger the better: $L(C) = k(1/C)^2$ (6)

Smaller the better: $L(C) = kC^2$ (7)

Nominal the best: $L(C) = k(C - m)^2$ (8)

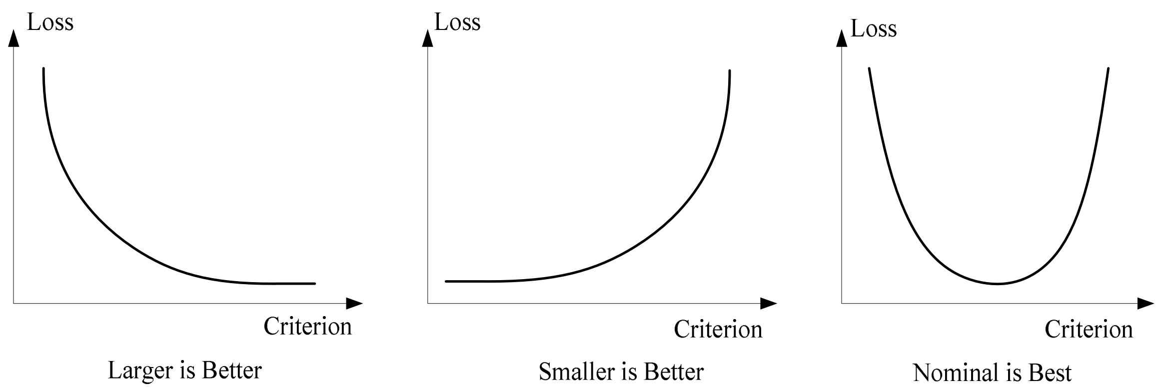


Figure 19: Classification of the Loss Function

Not only does the LF play a crucial role in robust design, but also it provides a good metric for multi-criteria decision making. The LF, with a physical meaning, directly represents the decision maker's preference. If one criterion is more important than the others, the loss due to the derivation from the target value with respect to this criterion will be higher than the loss contributed by any other criterion. Similarly, two criteria will result in the same loss if their importance is equal. Unlike the conventional weighting method that involves trial-and-error, the LF is a direct way to indicate the decision maker's preference and is simple to apply.

Utility Function

Utility, which originated in economics, is an abstract variable, indicating goal-attainment or want-satisfaction. It is also can be considered as a "measure of satisfaction or value which the decision maker associates with each outcome" [Dieter, 2000]. Utility is a concept that was introduced by Daniel Bernoulli, a Dutch mathematician in the eighteenth century. His diminishing marginal utility (for the usual person, utility increased with wealth but at a decreasing rate, which is represented as an utility function shown in Figure 20) stems from his solution to the famous St. Petersburg Paradox [Martin, 2004]. The diminishing marginal utility indicates that a person's valuation of a risky venture is not the expected return of that venture, but rather the expected utility from that venture. Bernoulli's idea profoundly influenced his and subsequent generations. In Theory of Political Economy by Jevons in 1871, the concept of utility is first explicitly explained and systematically used [Barbera et al., 1998]. Since then, utility theory has been enriched and improved by scientists from various fields and is now a mature theory applied in many areas.

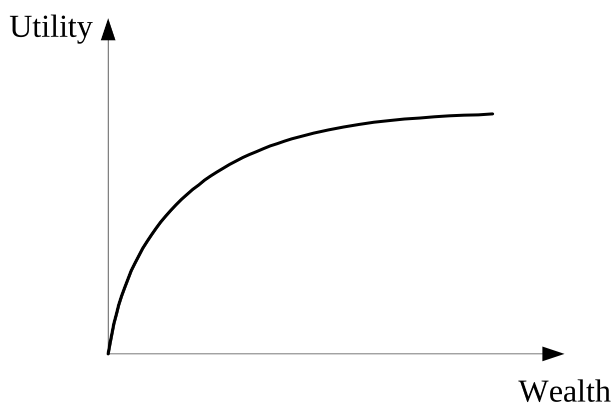


Figure 20: Utility Function

Utility function has the capability of representing a decision maker's preference information by measuring the "goodness" of the decision making criteria. The numerical value of goodness measured by utility is obtained by a function which expresses utility as a mathematical function of the decision making criterion. The utility function may be visualized as moving weights so that the relative contributions made by different attributes to the ranking of alternatives change with the attribute values themselves. A utility function used to describe decision maker's preference is clearly not unique. If the value of the utility function were to be doubled, squared, or subjected to any other strictly monotonically increasing function, it would still describe the same preference.

Class Function

Physical programming (PP) is a multi-criteria optimization method which captures designer's preference information [Messac, 1996] in a physical meaning. In this method, the designer's preference with respect to each criterion is represented as a set of ranges with different degree of desirability by using class function. An example of soft class functions for physical programming is shown in Figure 21. The horizontal axis represents the value of the criterion g_i, and the class function, which will be minimized for the criteria, $\bar{g}_i$ is on the vertical axis.

There are four types of class function: smaller is better, larger is better, value is better and range is better, as depicted in Figure 21, respectively. The class function has the degree of desirability of six ranges for each generic criterion for classes 1S and 2S, ten ranges for classes 3S and eleven for class 4S, from highly desirable to highly unacceptable in order of decreasing preference. The parameters g_{i1} through g_{i5} defining the limits of each desirability range are physically meaningful values that are provided by the designer to quantify the preference. The class function has been proved to be able to remove the weight-tweaking process that usually exists in the weighed sum method [Chen et al., 1999] and is considered a promising method to represent designer's preference information.

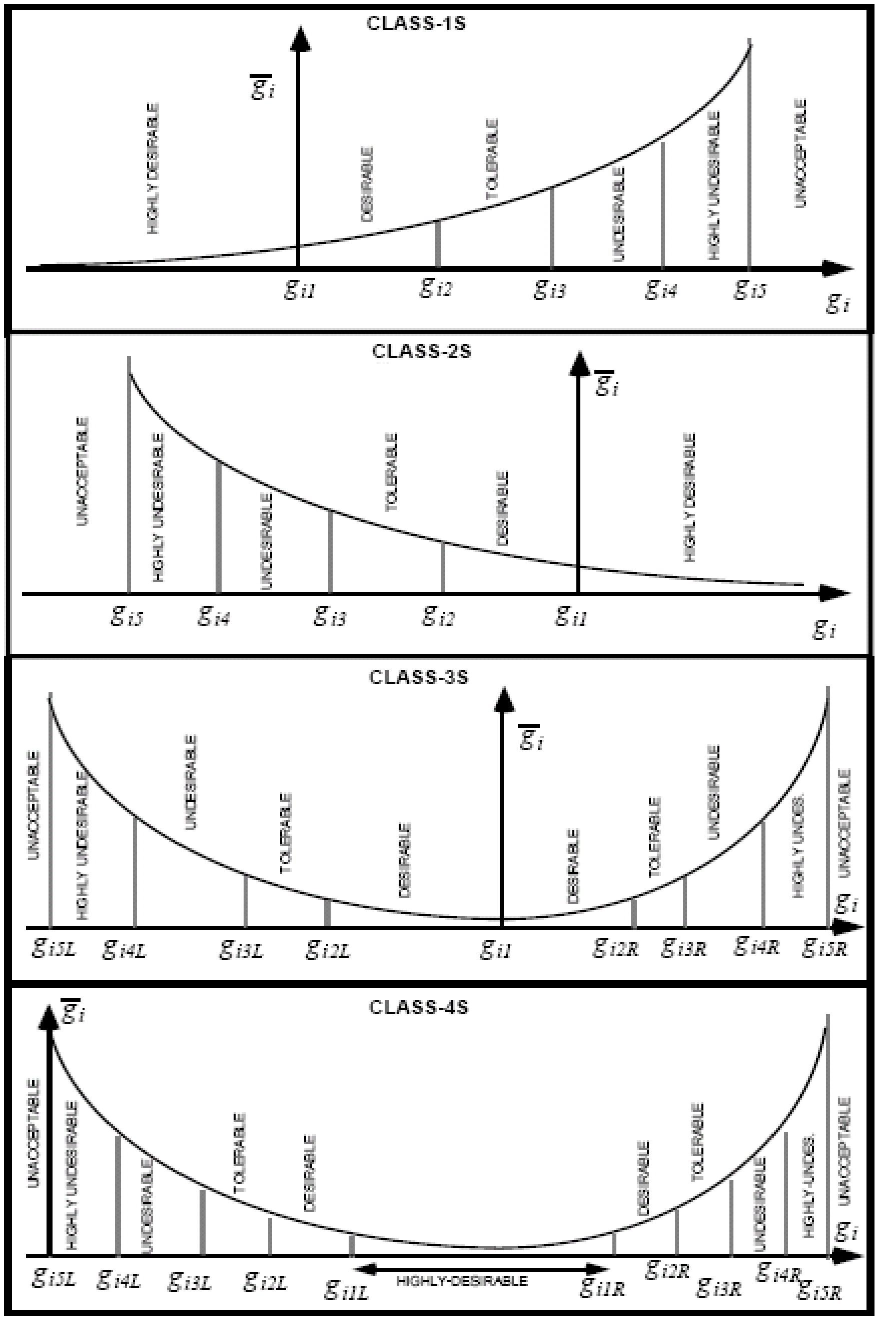


Figure 21: Soft Class Functions for Physical Programming [Chen et al., 1999]

2.6.2.2 Overall Evaluation Criterion (OEC)

The Overall Evaluation Criterion (OEC) method is also known as Simple Additive Weighting (SAW) or Weighted Sum (WS) method. The OEC is an elementary MADM method that aggregates multiple attributes into one function in which the multiple attributes are translated into a single evaluation metric. The function is a linear combination of the weighted normalized attributes. A generic OEC function is shown in Equation 9.

$$OEC = \alpha\left(\frac{Criterion\_1}{Criterion\_1_{BL}}\right) + \beta\left(\frac{Criterion\_2_{BL}}{Criterion\_2}\right) + \gamma\left(\frac{Criterion\_3}{Criterion\_3_{BL}}\right) + \cdots + \xi\left(\frac{Criterion\_n}{Criterion\_n_{BL}}\right) \qquad (9)$$

To calculate the value of OEC, the attributes are normalized by their corresponding baseline values first. By doing this one can avoid “adding apples and oranges”. If a criterion is a “benefit” criterion, its normalized value is obtained by being divided by its baseline value, on the other hand, if a criterion is a “cost” criterion, its normalized value can be obtained by dividing the baseline value by itself. α, β and γ are relative weights of the criteria and their summation is unity. These weights provide the ability to tailor the OEC to specific needs, preferences, or points of view of a customer [Mavris and DeLaurentis, 1995]. OEC is expected to be maximized, that is, the “best” solution suggested by this method has the highest value of OEC.

OEC is one of the MADM techniques that is widely used. The advantage of this technique is its simplicity: it is easy to understand and use. On the other hand, OEC does not consider the correlation between the attributes and is sensitive to the relative weights. These situations will become worse when the number of attribute increases. In addition, the values of the baseline attributes have strong impacts on the calculation of the OEC and have a critical impact on the final decision.

2.6.2.3 Technique for Ordered Preference by Similarity to the Ideal Solution (TOPSIS)

TOPSIS is one of the widely used compensatory decision making techniques. It starts with the construction of a decision matrix, where the qualitative evaluation of each attribute of the alternatives is provided. Then the matrix is normalized so that each attribute has the same unit length of vector. Thus various attributes can be compared with each other based on the normalized value. The normalized decision matrix then is weighted by the relative weights of the attributes which represent the designer's preference information. And the attributes are classified into "benefit" and "cost" attributes. A "benefit" attribute is defined as the one whose value varies in the same direction with the product's performance, while a "cost" attribute affects the product's performance in the opposite direction. Sequentially, the positive and negative ideal solutions are identified, where the positive ideal solution is composed of the maximum values of the benefit attributes and the minimum values of the cost attributes of all the alternatives, while the negative ideal solution is the opposite of the positive ideal solution. The separations of an alternative to the positive ideal solution S_i^* and the negative ideal solution S_i^- are measured by the n-dimensional Euclidean distance in the attribute space, given by Equations (10) and (11) respectively. Finally, the closeness of each design alternative to the ideal points is given by Equation (12).

$$S_i^* = \sqrt{\sum_{j=1}^{k} (y_{ij} - y_j^*)^2} \qquad i = 1, \cdots, \mathrm{n} \tag{10}$$

$$S_i^- = \sqrt{\sum_{j=1}^{k} (y_{ij} - y_j^-)^2} \qquad i = 1, \cdots, \mathrm{n} \tag{11}$$

$$C_i^* = \frac{S_i^-}{S_i^- + S_i^*} \qquad i = 1, \cdots, \text{n} \tag{12}$$

The “best” solution suggested by TOPSIS is the design alternative that is the furthest from the negative ideal solution and closest to the positive ideal solution, that is, the solution which maximizes the value of C_i^*. This selection concept is clearly illustrated in Figure 22.

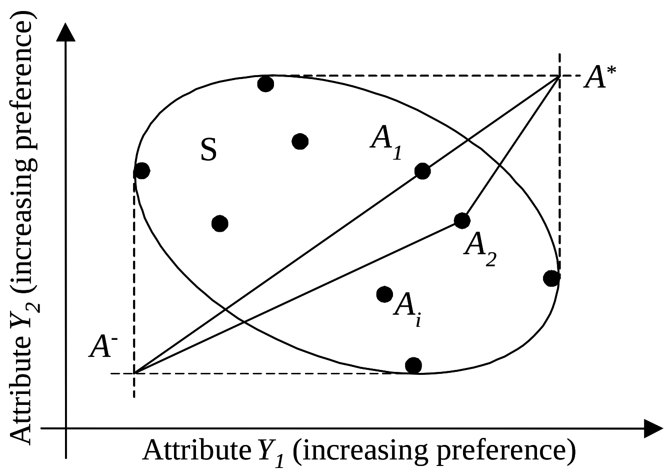


Figure 22: TOPSIS Technique

Because of its simplicity, TOPSIS has become a widely used MCDM technique. The other advantages of this technique include the full utilization of information and the systematic computational procedure, which provide indisputable ranking order for the alternatives. However, the separation of the alternative from the ideal solutions is sensitive to the weights of the attributes, thus, inaccurate weights may result in incorrect final solution and the inaccuracy will increase with the number of attributes and alternatives. Hence, typically several weighting scenarios are required to be investigated to determine the final decision. Another drawback of TOPSIS is that it does not consider the feasibility of design alternatives and has the assumption that each alternative has some probability to be selected. These disadvantages described above lead to the fact that an alternative with better average goodness with regard to all attributes may still dominate the others while it violates one or more criteria. This inconsistency induces a paradox that an infeasible alternative with better average goodness may even be selected as the best solution. In addition, TOPSIS assumes that each attribute's utility is monotonic, which is not true for problems where a particular attribute value is desired to be achieved, such as in the "nominal is the best" case depicted in Figure 19.

2.6.2.4 Analytical Hierarchy Process (AHP)

The AHP technique is proposed by Saaty in the 1970's, which intended to facilitate the MCDM problems that have a hierarchical structure of attributes [Saaty, 1980]. This method deals with the complex problem based on the concept of translating the hierarchy problem to a series of pairwise comparison matrices and obtaining the preference information for the attributes. In this method, the preference information is elicited as the pairwise comparisons between attributes or alternatives and treated using the eigenvector method. The attributes are divided into different levels, and the overall goal of the hierarchical problem is on the top level, as shown in Figure 23. The attributes at the lower level are the sub-attributes of the ones which are at the immediate upper level. Each element (attribute or sub-attribute) at a given level is associated with some or all of the elements at the immediate upper level. To perform this method, a pairwise comparison matrix, as shown by Equation (3), is formulated for each element at the single level with respect to the element immediately above. To accomplish this task, elements at the single level are compared with the other elements at the same level in terms of attractiveness or goodness with respect to the element at the immediately higher level. And then the pairwise comparison is treated using the eigenvector method (described in section 2.6.2.1), and the relative weights of the attributes at this level can be obtained. This process is repeated from top to bottom of the hierarchy until the final result is reached.

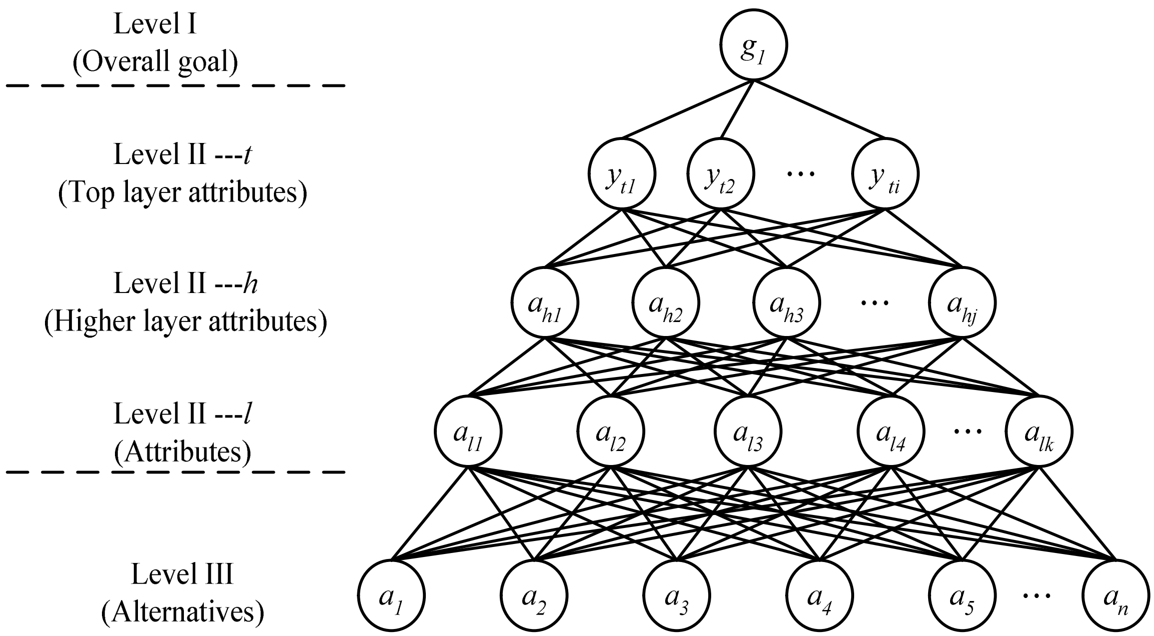


Figure 23: Structure of Analytical Hierarchy Process (AHP)

AHP is one of the powerful and flexible MCDM techniques to handle the complex decision making problem, especially with the hierarchical attributes. It reduces the complex problem to a series of one-to-one comparisons and can provide a clear rationale why the suggested design is the best. However, like other methods, it has its own limitations. It assumes that elements at any level except for the bottom level are preferentially independent. This assumption does not really hold for most decision making problems since the attributes at the same level often have correlated preference.

Also AHP requires each alternative to be compared with all others, however, many of such comparisons are redundant. This often causes an inconsistency problem, and "such inconsistency may become worse as the dimension of the comparison matrix increases. AHP also suffers from the rank reversal of alternatives depending on the number of the alternatives being assessed and this can be a disturbing factor in a normative decision making tool" [Sen and Yang, 1998].

2.6.2.5 Joint Probability Decision Making Technique (JPDM)

Joint Probabilistic Decision Making (JPDM) technique, which was developed by Oliver Brandte, incorporates a multi-criteria and a probabilistic approach to system design and can accurately estimate the probability of satisfying the criteria concurrently [Bandte, 2000].

The JPDM technique is based on multivariate probability theory, and can handle multi-criteria optimization and product selection problems. The heart of this technique is the construction of a joint probability distribution that combines the univariate distributions of each criterion (Figure 24). The probability distributions reflect the uncertainty associated with the design that is due to incomplete knowledge about the system. In the JPDM technique, the joint probability distribution is generated and serves in conjunction with a criterion value range of interest as a universally applicable objective function. The objective function, called Probability of Success (PoS), constitutes a meaningful metric that allows the designer or customer to make a decision based on the chance of satisfying the customer's requirements.

There are two models in the JPDM technique: the Joint Probability Model (JPM) and the Empirical Distribution Function model (EDF). The JPM is a parametric model that requires the user to provide relevant statistics for the univariate criterion distributions and uses these statistics to construct a joint probability density function. The bivariate normal distribution is the typical joint probability distribution in which each of the two random variables (x, y) has a normal distribution. The joint probability density function (JPDF) of the bivariate normal distribution is shown in Equation (13). The empirical distribution function, on the other hand, relies on empirical data collected by using a sampling technique such as Monte Carlo Simulation (MCS). The sampling data for each criterion are then used to build the JPDF. If the amount of the sampling data is large enough, the joint EDF yields the most accurate joint distribution prediction, since it does not rely on any approximation to generate criterion statistics. Equation (14) gives the joint probability mass function for the EDF model. The PoS, used as the objective function in JPDM, is given by Equation (15) and (16) for JPM and EDF model, respectively.

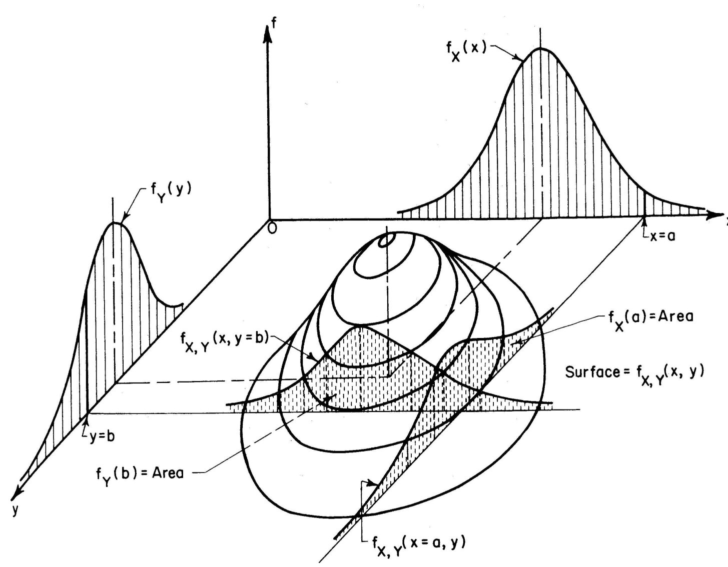


Figure 24: Joint Probability Distribution

$$f_{XY}(x,y)=\frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho_{XY}^2}}\exp\{\frac{1}{2\rho_{XY}^2-2}[(\frac{x-\mu_X}{\sigma_X})^2-2\rho_{XY}(\frac{x-\mu_X}{\sigma_X})(\frac{y-\mu_Y}{\sigma_Y})+(\frac{y-\mu_Y}{\sigma_Y})^2]\} \quad (13)$$

$$p(A) = \frac{1}{N} \sum_{i=1}^{N} I(X_i = A) \tag{14}$$

where $I(X_i = A) = \begin{cases} 1 \text{ if } X_i = A \text{ is true} \\ 0 \text{ if } X_i = A \text{ is false} \end{cases}$, $X = [x_1, x_2, \cdots x_n]^T$, $A = [a_1, a_2, \cdots a_n]^T$.

$$PoS = \int_{\Omega} f(X)dX \qquad \text{(JPM Model)} \tag{15}$$

$$PoS = \frac{1}{N} \sum_{i=1}^{N} I(X_l < X_i < X_u) \qquad \text{(EDF Model)} \tag{16}$$

where Ω is solution space $X_l = [x_{1l}, x_{2l,} \cdots, x_n]^T$ and $X_u = [x_{1u}, x_{2u,} \cdots, x_{nu}]^T$ are lower and upper limits of the criteria X.

The advantages and disadvantages of the JPDM technique are listed in Table 2 [Bandte, 2000]. The JPDM technique is explained in detailed in Appendix B.

Table 2: Comparison of EDF and JPM Models [Bandte, 2000]

	EDF	**JPM**
Advantages	• No approximation with standard distribution needed • Estimates joint probability from data directly • Most exact method • Very fast estimation of joint probability	• Only limited information needed •Can employ expert guesses in case of lack of simulation • Easy used in conceptual design
Disadvantages	• Large amount of data needed in order to be accurate •Requires modeling and simulation	•Requires approximation with standard distribution •Requires correlation function •Estimation of joint probability is time consuming

2.6.2.6 Expected Utility Theory (EUT)

Utility is a numerical measure of “goodness” of a product or process. The expected utility hypothesis was formulated in Cramer’s (1728) suggestion for resolving the St. Petersburg Paradox, and, ten years later, stemmed from Daniel Bernoulli’s solution to the paradox [Fielding, 1999] The Paradox posed the following situation: tossing a fair coin repeatedly until the first time the first “head” appears. If this happens on the k-th toss, then the prize is 2^k ducats. How much is it worth paying to be allowed to play the game? Clearly, the expected winning is:

$$E(w) = \sum_{k=1}^{\infty} \left(\frac{1}{2}\right)^k \cdot 2^k = 1 + 1 + \cdots + 1 + \cdots = \infty \tag{17}$$

However, the paradox is that “no reasonable man would be willing to pay 20 ducats as equivalent” [Bernoulli, 1954] though the expected return is infinite. Daniel Bernoulli’s solution to this paradox includes two ideas: one is that people’s utility of wealth, $u(w)$, is not increasing linearly, but increasing at a decreasing rate; the other is that a person prefers a lottery only if its expected utility of wealth is greater than what it costs, not based on the expected return of that venture. Figure 25 shows how the value and cost of the wealth change with the wealth in the paradox case.

For the St. Petersburg game, if the potential player has a utility of wealth level u given by $u(w)$ and starts with initial wealth $w_0 > 0$, then the amount that the player is willing to pay for playing the game must satisfy the Equation (18)

$$u(w_0) = \sum_{k=1}^{\infty} \left(\frac{1}{2}\right)^k \cdot u(w_0 + 2^k - a) \tag{18}$$

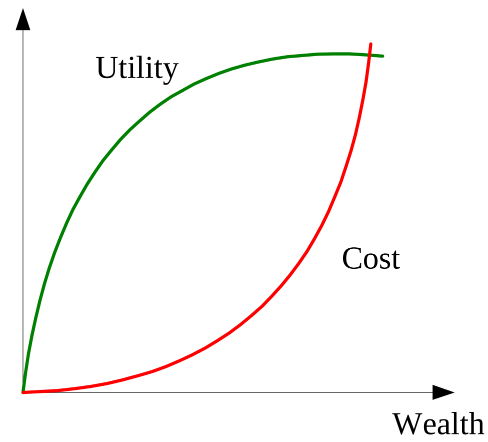


Figure 25: Expected Utility Theory

Therefore, by Bernoulli's logic, the expected utility of any risky venture takes the form below [Fonseca and L, 2006]:

$$E(u|p, X) = \sum_{x \in X} p(x)u(x) \tag{19}$$

where X is the set of possible outcomes, $p(x)$ is the probability of a particular outcome $x \in X$ and $u: X \rightarrow R$ is a utility function over outcomes. Equation (19) describes the essence of the EUT technique.

EUT is often used for decision making under uncertainty and risk through comparing the expected utility which is obtained by adding the expected utility values of outcomes multiplied by their probabilities. This method maintains that, facing uncertainty, people behave or should behave as if they are maximizing the expectation of utility of possible outcomes.

EUT is known as a rational model that can well describe the DM's behavior. Several types of tests have been performed and discovered the capabilities of this theory. However, this technique has its own weakness, for instance obtaining an accurate utility function for each attribute is not a easy task and it is difficult to keep the consistency between utility values of the attributes.

2.6.3 MODM Method

Multi-Objective Decision Making (MODM) methods are designed to handle the MCDM problems where the "best" design is selected from a large set of alternatives which satisfy the given requirements and objectives. That is, optimization will be performed to maximize or minimize the associated objectives, and the final selected solution is a design with the best values of the objectives. In aerospace system design, these objectives are typically the attributes of the system, evaluating the system's performance, cost or operational environment. In general, these objectives are often conflicting so the optimal solution is usually a compromise concept that can best simultaneously satisfy the different objectives. Figure 26 lists some MODM methods that are capable of dealing with this class of problems. These MODM methods are classified into different groups "mainly based on the types of preference information and timing for eliciting preference information" [Sen and Yang, 1998].

A decision tree for MODM technique selection was also developed by Sen and Yang [Sen and Yang, 1998], as illustrated in Figure 27. By using this figure, user can construct a choice rule to select a method by examining the decision rule or the computational procedure of the methods.

.

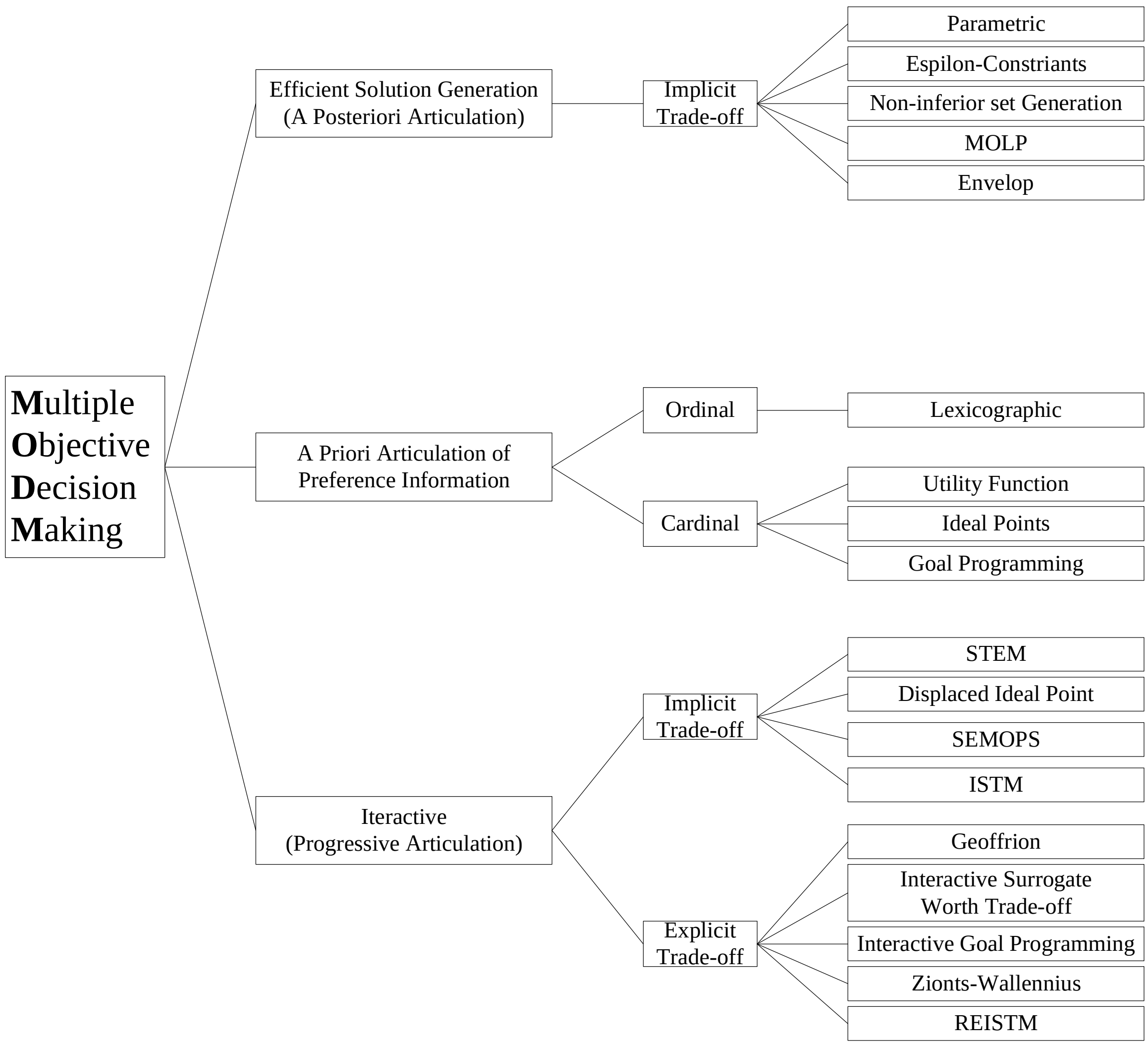


Figure 26: Classification of MODM Methods [Sen and Yang, 1998]

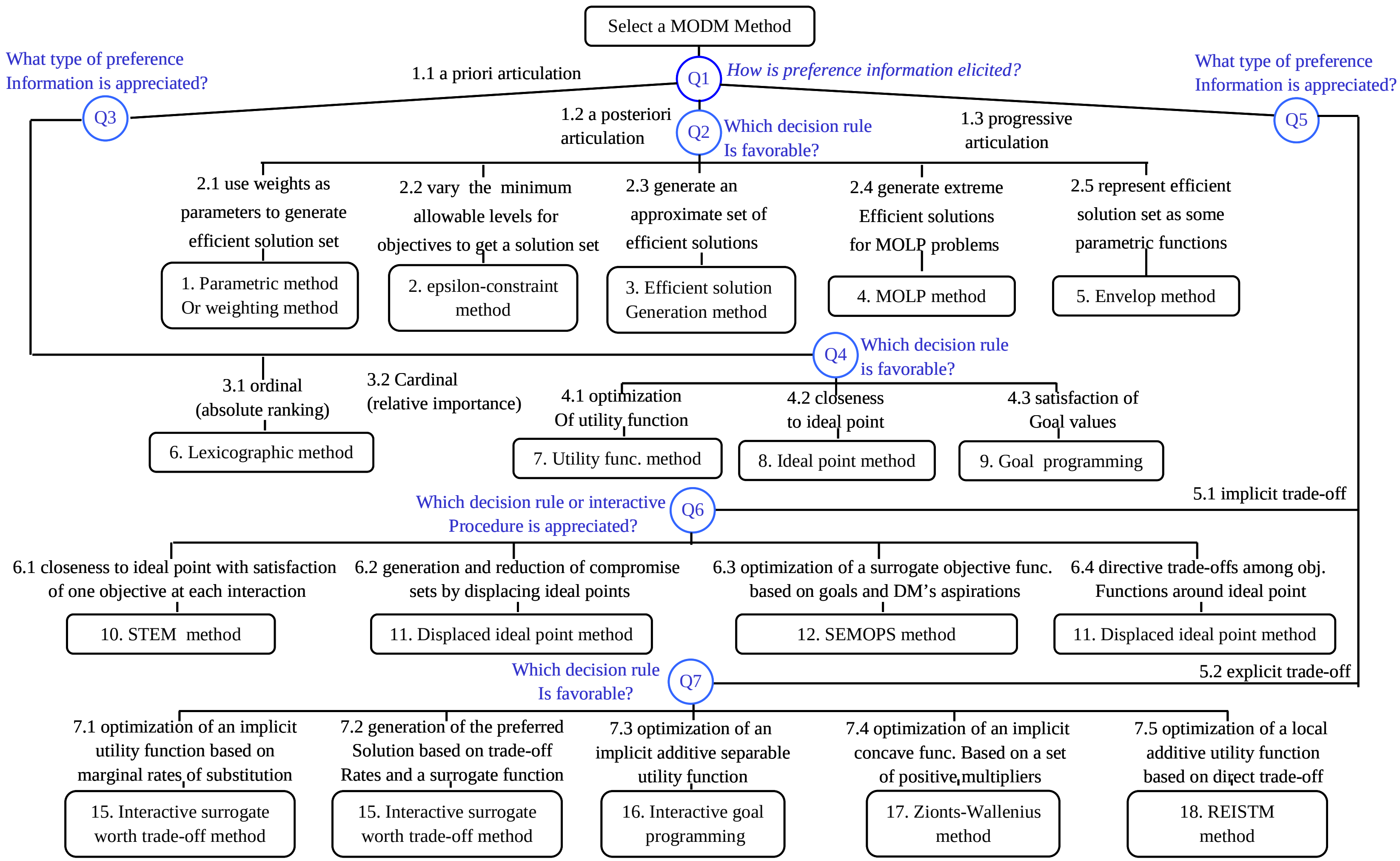


Figure 27: Decision Tree for MODM Technique Selection [Sen and Yang, 1998]

2.6.3.1 Additive Weighting Method

The additive weighting method, also known as parametric method, is one of the most elementary and commonly used techniques. This method employs a weighted sum of the objectives as the objective function and minimizes the function to obtain the Pareto optimal solutions. The objective function for a problem with N objectives is given by Equation (20).

$$\begin{aligned} &Min \quad F(X) = \sum_{n=1}^{N} w_i f_i(X) \\ &s.t. \quad X \in \Omega \quad W = [w_1, w_2, \cdots, w_N] \end{aligned} \tag{20}$$

where the W is the weight vector representing the relative importance of the objectives. The Pareto optimal solutions can be generated by solving the Equation (20). Since there is infinite number of Pareto optimal solutions, a final solution can be obtained when the weight vector is given. Therefore, the weights are used as parameters to identify the "best" solutions. Figure 28 and Figure 29 depict the scheme of the method of a two objectives with a convex and nonconvex feasible space, respectively. This figure shows that the optimal solutions are the points where the hyperplane $L = \left\{ f(X) \middle| \sum_{n=1}^{N} w_i f_i(X) = c \right\}$, c is a constant, is tangential to feasible space Λ. The slope of the L is $-w_1 / w_2$, and the thick lines indicate the Pareto optimal solutions. In Figure 28 the feasible space is convex, so the preferred solution can be directly found by the method. However, in Figure 29, since the feasible space is nonconvex, the Pareto optimal solutions between A and B are not able to be identified by the method no matter what weight vector is used.

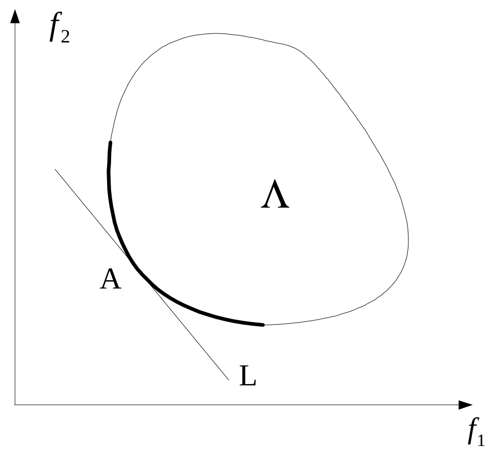


Figure 28: Additive Weighting Method with a Convex Set [Hwang and Masud, 1979]

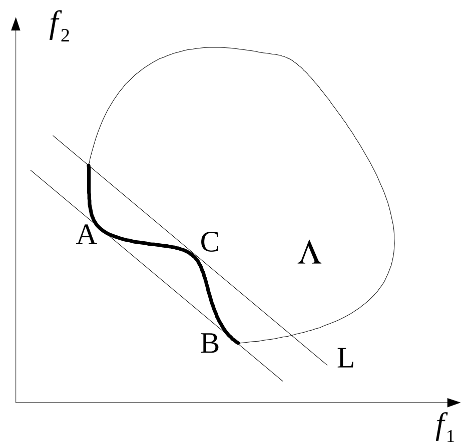


Figure 29: Additive Weighting Method with a Nonconvex Set [Hwang and Masud, 1979]

2.6.3.2 Goal Programming (GP)

The Goal Programming (GP) is a MODM technique that requires the decision maker to determine goals for all the objectives that are expected to be achieved. This method utilizes the concept that the best compromise design should be the one which has the minimum weighted sum of deviations from the set goals.

In most cases, the goals of the objectives are not “hard” or restricted and have some ambiguity. This often happens in reality, such as when designing a commercial aircraft, a goal can be "the payload should be approximately 1500lb". Goal programming method allows treating these kinds of goals as "soft" constraints, which is not too restrictive and can be violated. Hence, the "best" solution is the design which has minimum weighted deviation from the ideal solution where all the goals are exactly met. This method also allows setting the preemptive weights to the objectives and the preemptive weights may have different achievement levels.

This method can be formulated in the Equation 21

$$\begin{aligned}
&Min \quad \left\{ \left[\sum_{j=1}^{k} (w_j^+ d_j^+ + w_j^- d_j^-)^p \right]^{\frac{1}{p}} \right\} \\
&s.t. \quad X_b \in \Omega_b \\
&\Omega_b = \begin{Bmatrix} f_j(X) - d_j^+ + d_j^- = \hat{f}_j & \mathrm{j} = 1, \cdots, \mathrm{k} \\ d_j^+ \cdot d_j^- = 0 & X_b = \left[X^T \; d_1^+ \; d_1^- \cdots d_k^+ \; d_k^- \right] \\ d_j^+, \; d_j^- > 0 & X \in \Omega \end{Bmatrix}
\end{aligned} \tag{21}$$

where d_j^- , d_j^+ are deviation variables representing under-achievement and over-achievement of the goal. w_j^- , w_j^+ are the relative weights for the corresponding deviation variables. $\hat{f}_j$ is the goal of the j-th objective.

To solve Equation 21, an ordinal ranking of the objectives is required, which leads to a sequence of problem below:

$$\begin{aligned}
&\min \; a_l = p_l h_l(D^+, D^-) \\
&s.t. \quad X_b \in \Omega_b, l = 1,2,\cdots, L
\end{aligned} \tag{22}$$

where $D^+ = [d_1^+, d_2^+, \cdots, d_k^+]^T$, $D^- = [d_1^-, d_2^-, \cdots, d_k^-]^T$, and p_l is the preemptive weight. L is the number of the priority levels.

Firstly, a_1 is minimized to obtain a_1^*. Then a_2 is minimized, but subject to an additional condition: $a_2 \le a_1^*$. This process is repeated until a_L is minimized so that the compromise design of the MODM problem X^* can be obtained. In this process, the Simplex algorithm can be used to solve the problem whose objectives are linear functions of design variables X, while any single objective nonlinear optimization technique can be utilized iteratively to solve the problem with nonlinear objectives.

GP is considered one of the best methods to find the best compromise solution for a MODM problem. However, it is often a difficulty for the decision maker to set the goals for all the objectives, and this method is not able to discover all the efficient designs for a non-convex problem [Sen and Yang, 1998]. In addition, the ordinal ranking means the higher ranking objective may not be detrimented while minimizing lower ranking ones, which limits the possible solutions[Charnes and Cooper, 1977].

2.6.3.3 Physical Programming (PP)

Physical Programming (PP) is a technique closely related to goal programming that uses a set of soft class functions, as shown in Figure 21, to represent the decision maker's physical preferences. The objectives are classified in an intuitive manner from highly desired to unacceptable and then is used to construct an aggregate objective function which is weighted sum of the class functions. The method is formulated by the Equation (23)

$$\begin{aligned} &\text{Min}: \log_{10}\left[\frac{1}{n_{sc}}\sum_{i=1}^{n_{sc}}\overline{g_i}\left[g_i(x)\right]\right] \\ &s.t. \quad \Omega \end{aligned} \tag{23}$$

where Ω is the design space and n_{sc} is the number of soft classes.

Physical programming offers a problem formulation and solution framework that conforms to real-life design. It allows the decision maker to define their preference in physically meaningful terms which capture the DM's physical understanding of the desired design outcome. This removes the frustrating process of weight tweaking entirely, which often happens to the traditional methods.

PP also has its own drawbacks. Firstly, PP requires a priori selection of range parameters for each of the objective functions. In problem formulation phase, the decision maker's time is mostly consumed in exploring the implication of the various physical meaningful preference choices. Secondly, PP only provides information for one design scenario at a time. To capture a variety of design scenarios, a set of preference structures should be built and tested. Thirdly, PP is a deterministic design method and does not capture the uncertainties due to incomplete information existing in design space and operational environment.

2.6.4 Intelligent Decision Support System

Decision Support Systems (DSS), originally developed to aid managers in the decision making processes at the beginning of 1970's [Little, 1970], are considered a set of procedures for data and reasoning management. It covers a wide variety of systems, tools and technologies, and integrates them into a computer system to facilitate the decision making process. Various definitions have been given to this term by the researchers in the early days after this term just emerged. Keen and Scott-Morton [1978] proposed the following classic definition: "DSSs combine the intellectual abilities of humans with the abilities of computer systems in order to improve the quality of the decisions made. DSSs are computer-based systems that are used in order to support decision makers in ill structured problems".

Sprague and Carlson considered DSS a set of procedures, which focuses on expanding the DM's cognitive space regarding the confronted problem with the aid of a computer [Sprague and Carlson, 1982]. The definition was extended by Andriole [1989], Sage [1991] and Adelman [1992], to the final formulation below:

> *Decision Support Systems are interactive computer-based systems (software), which use analytical methods such as decision analysis, optimization algorithms, etc, in order to develop appropriate models that will support decision makers in the formulation of alternative solutions, the resolution of the reactions amongst them, their representation, and finally in the choice of the most appropriate solution to be implemented.*

Therefore, DSS is computer-based information system that uses data and multi-criteria decision making (MADM and MODM) models to organize information for decision situations and interact with decision makers to expand their horizons. It highly alleviates the DMs' burden in dealing with the problems which are semi structured or ill-structured, and supports all the phases in a decision making process. In addition, the systems are able to store and process a large amount of knowledge at much higher speed than the human mind, and therefore can considerably improve the decision making quality. Various DSSs were proposed in the past decades, and the systems mainly aimed at easing the DM's tasks in decision making process.

2.6.4.1 Distributed Decision Support Systems

It is clear to see that in today's engineering design DMs seldom make decision alone, since the decision making problems are becoming more and more complicated. This complexity inspires the idea that decomposing the complex decision making problem into partial problems and handling each by different groups of experts. This motivation results in the emergence of the Distributed Decision Support System (DDSS), a specific DSS to handle the Distributed Decision Making (DDM) situation. DDM is defined as a "decision making process in which the participating people own different specialized knowledge, execute different specialized tasks, and communicate with each other through a computer environment, which aims at the support of the entire process" [Chi and Turban, 1995].

With the development of Information System (IS), the utilization of DDM is dramatically expanded. The ISs have the on-line and real time information capabilities through which the DDM can be fulfilled easily and efficiently because the ISs offer immediate response and easy information exchange. Most of the current information systems provide such capabilities that can be characterized as distributed on-line systems. More recently, the web-based DSSs are viewed as clients linked to a server hosting the DSS application, and have great potentials to inspire new distributed, cooperative or collaborative decision support strategies impacting the very core structures of the DSSs.

2.6.4.2 Artificial Intelligence

After the first calculating machine, the abacus, was invented by the Chinese in the twenty-sixth century BC, the ability to mechanize the algebraic process intrigued humans, and eventually great progress was made as the digital computer was invented by Charles Babbage in 1856. The digital computer was rapidly employed in many areas and alleviates some of the onerous and tedious work that people engage. At almost the same time, researchers make efforts to create machines with some sort of intelligence. In 1950, Alan Turin, the "father of Artificial Intelligence (AI)" presented the famous Turing test to show that it is possible for a machine to think as a human being [Rich, 1983]. Eventually, artificial intelligence become an area of computer science that focuses on making intelligent machines, especially intelligent computer programs, that can engage on behaviors that humans consider intelligent. Today with the rapid upgrading of the computer and 50 years of research, AI has been utilized in various fields, such as decision making, game playing, computer vision, speech recognition, expert systems and so on.

2.6.4.3 Expert System

Expert system (ES) is viewed as the most well-known application field of artificial intelligence. ESs are problem-solving programs that combine the knowledge of human experts and mimic the way human experts reason. The goal of the expert system is to emulate the problem-solving process of an expert whose knowledge was used in developing the system.

MYCIN, developed at Stanford in 1974, was one of the first programs to address the problems of reasoning with uncertain or incomplete information. "MYCIN provided clear and logical explanations of its reasoning, used a control structure appropriate to the specific problem domain, and identified criteria to reliably evaluate its performance" [Luger and Stubblefield, 1998]. Nowadays, many of the ES development techniques in use were originated from the MYCIN project

Figure 30 presents the typical structure of an expert system, which consists of three modules: user interface, inference engine, and knowledge base. The operation procedure starts from the user's task querying through the user interface. After receiving the query from the user, the inference engine manipulates and uses information in the knowledge base to form a line of reasoning. And then the response is provided by the ES via the user interface. Further input information may be required from users until the system reaches a desired solution.

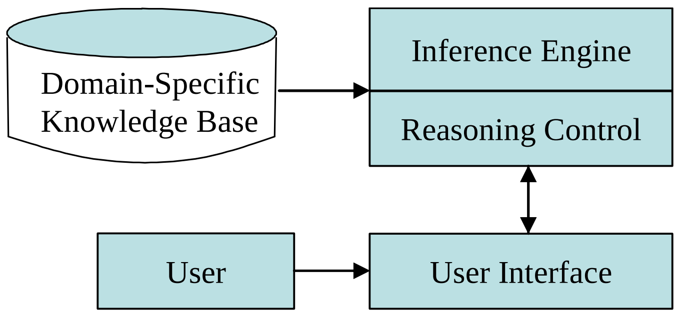


Figure 30: Typical Structure of an Expert system

The user interface system allows the user to interact with the system to accomplish a certain task. It manages the interaction, which can be menus, natural language or any other type of data, between the system and users. A user can be 1) an expert, who maintains and develop the system, 2) an engineer, who employs the system to solve their specific problem or 3) a student, who is trained for the problem solving procedure.

The inference engine is the control mechanism that applies information present in the knowledge base to task-specific data to arrive at a conclusion. It organizes and controls the steps taken to solve the problem. The most widely used problem-solving method at this point is IF-THEN rules, and the ESs that use the rules for reasoning are called rule-based systems. In rule-based systems, inference engines utilize the idea that if the condition holds then the conclusion holds to form a line of reasoning. There are a few techniques for drawing inferences from a knowledge base such as forward chaining, backward chaining and tree search. Forward chaining starts from a set of conditions and moves to a conclusion while backward chaining has the conclusion first and tries to find a path to get the conclusion. Tree search is applied when the knowledge base is represented by a tree, and the reasoning process is performed by checking the nodes around the initial node until a terminal node is found.

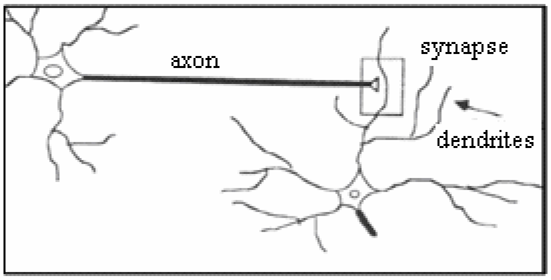
The knowledge base is the core of the advisor system. Its main purpose is to "provide the guts of this system --- the connections between ideas, concepts, and statistical probabilities that allow the inference engine to perform an accurate evaluation of a problem" [Boss, 1991]. The knowledge base stores facts and rules, which include both factual and heuristic knowledge and support the judgment and reasoning of the inference engine. "Factual knowledge is that knowledge of the task domain that is widely shared, typically found in textbooks or journals, and commonly agreed upon by those knowledgeable in the particular field while Heuristic knowledge is the less rigorous, more experiential, more judgmental knowledge of performance. In contrast to factual knowledge, heuristic knowledge is rarely discussed, and is largely individualistic. It is the knowledge of good practice, good judgment, and plausible reasoning in the field. It is the knowledge that underlies the 'art of good guessing'" [Engelmore and Feigenbaum, 1993].

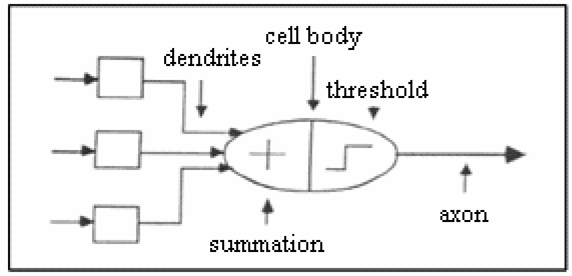
2.6.4.4 Neural Network

Neural Networks are an information processing technique that is inspired by the way that biological nervous systems, such as the brain, process information. The structure of the neural networks consists of a large number of highly interconnected processing elements or neurons to simulate the human reasoning process. A neural network system can learn by example, like a human, to resolve problems, and can be configured for a specific application, such as pattern recognition or data classification, through a learning process. Just as in biological systems, learning involves adjustments to the synaptic connections that exist between the neurons [Neural Network, 2002].

Neural networks appear to be a recent development, however, they were created before the advent of the digital computer. The first biggest step towards neural network came in 1943 when the neurophysiologist Warren McCulloch and the logician Walter Pits published a paper on how neurons might work and modeled a simple neural network with electrical circuits [Anderson and McNeill, 1992]. Since then, various research activities on neural networks have emerged. In the late 1950's Frank Rosenblatt, a neuro-biologist of Cornell, intrigued by the operation of the eye of a fly, began work on the Perceptron. The result from this research is the oldest neural network which is still in use today. In 1959, Bernard Widrow and Marcian Hoff of Stanford developed a model called MADALINE that is an adaptive filter to eliminate echoes on phone lines. MADALINE is considered the first neural network to be applied to a real world problem. In 1982, John Hopfield of Caltech developed an approach to create useful devices instead of simply modeling brains. [Anderson and McNeill, 1992] Today, neural networks become an interesting area that attracts various researchers.

A neural network, inspired by the structure of the brain, consists of highly interconnected entities, called nodes or units. Each unit is designed to mimic its biological counterpart, the neuron, and each of them accepts a weighted set of inputs and responds with an output. Figure 31 (a) illustrates a human neuron unit, and (b) shows a simplified model of a real neuron [Stergiou and Siganos, 2005].





(a) **(b)**

Figure 31: (a) Human Neuron Unit, **(b)** Artificial Neuron Model

An artificial neural network typically consists of hundreds of such processing units shown in Figure 31. These units are wired together in a complex communication network as shown in Figure 31 [Stergiou and Siganos, 2005]. A typical neural network consists of three groups: input layer, hidden layer and output layer. The input layer is the unit where raw information is fed into the network. The input layer is connected to the hidden layer whose activities are determined by the activities of the input layer and weights on the connection between input and hidden units. The output layer is connected to the hidden layer and its activities are determined by the units in the hidden layer and weights on the connections between the hidden layer and the output layer.

In a neural network, a node or unit fires (sends off a new signal) if it receives a sufficiently strong input signal from the other nodes to which it is connected. The strength of these inputs may be varied in order for the network to perform different tasks. Unlike traditional computers which use a CPU to execute a rigid set of rules (the program or software) sequentially, neural networks are composed of many rather feeble processing units which are interconnected into a network. Their computational power depends on working together on any task, therefore there is no central CPU following a logical sequence of rules [Intelegen Inc., 2005]. This type of computation is related to a dynamic process of node firings and the structure of the neural network is much closer to the physical workings of the brain.

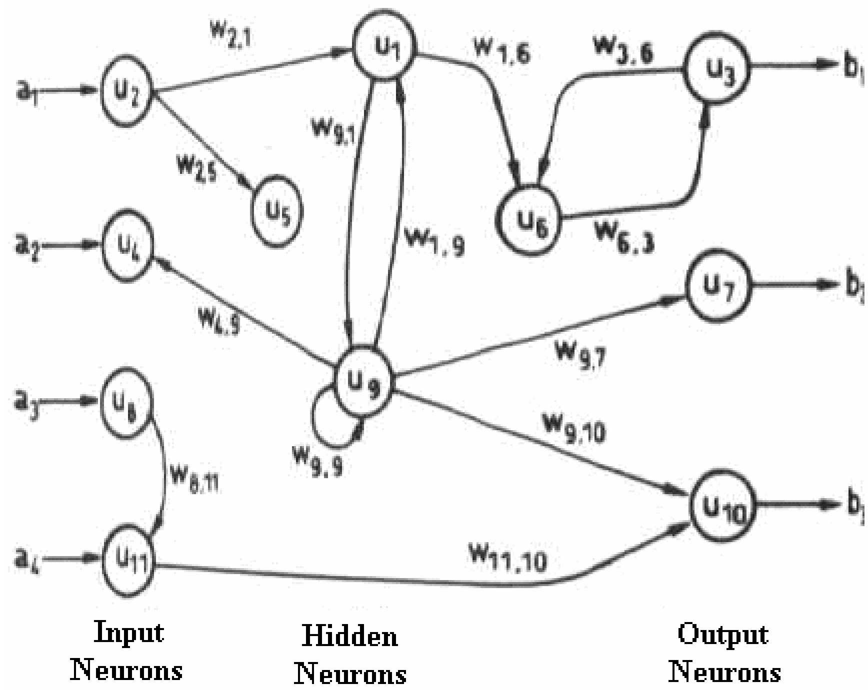


Figure 32: An Example of Complicated Neural Network

Today neural networks are being applied to an increasing number of real-world problems of considerable complexity such as medical diagnosis, process modeling, financial forecasting and so on. It has been shown to be particularly useful in solving problems where traditional artificial intelligence techniques involving symbolic methods have failed or been proven inefficient. Neural networks are also applied in the decision making field to help the DMs get the desired decision for a complex problem.

2.6.4.5 Data Mining

Nowadays, more and more attention is paid to analyzing data as the world is becoming data-driven. Data mining, also known as Knowledge Discovery in Databases (KDD), is one of the solutions for data exploration. It is defined as "the nontrivial extraction of implicit, previously unknown, and potentially useful information from data" [Frawley et al., 1991]. Data mining has been coined to describe a variety of techniques to identify rules/patterns of information, or decision-making knowledge in a large amount of structured or unstructured data, and extract these in such a way that they can be put to use in the areas such as decision support, prediction, forecasting and estimation[Chapple, 2006]. Hence, the use of data mining can uncover the patterns or rules inherent among the set of data, which helps organizations make better and timelier decisions.

Analogously, data mining is finding the proverbial needle in the haystack, where the needle is the desired piece of intelligence and the haystack is the large data warehouse which is built up over a long period of time. The data warehouse is a database where the data are organized and presented as information to the DM in order to aid the decision making. Figure 33 illustrates the relationship between a data warehouse and data mining. Typically, the data to be mined are extracted from a data warehouse into a data mining database or data mart. However, a data warehouse is not a requirement for data mining. In many cases, building up a data warehouse is an enormous task and time-consuming. An alternative way to mine data is to directly extract data from the source databases into a read-only database which functions as a data mart.

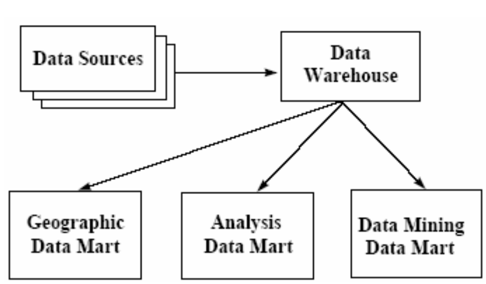


Figure 33: Data Mining and Data Warehouse

CHAPTER III

SOLUTION APPROACH TO METHOD SELECTION

As stated in section 2.3.1, modern engineering design is essentially a decision making process. From problem definition to final concept selection, decision making permeates through all the design phases. Especially, in the concept selection stage, the concept that best satisfies the conflicting criteria will be identified by the decision maker with the support of a multi-criteria decision making technique. Therefore, the method used to make decisions on concept selection appears very important to reducing the desired solution to the design problem and thus needs to be carefully selected.

Various methods with the intentions of facilitating the decision making process have been developed. However, instead of easing the decision making process these numerous methods complicate the decision problem because the large number of methods offers difficulties in selecting an appropriate method. It has been proved that an inappropriate method is not able to capture the essence of the problem under consideration and may result in an undesired solution inconsistent with the DM's preference. Hence, it is necessary to find an approach which is able to identify the most appropriate decision making method for the problem and then provide the guidance to decision maker to obtain the final solution using the selected method.

The study presented in this section introduces an intelligent, knowledge-based Multi-Criteria Decision Making Advisor (MIDAS) system which supports the DMs to fulfill the above tasks. The system is capable of aiding the DMs in selecting the most appropriate method for the problem under consideration, validating the correctness of a decision made using a specific method, and providing advices for generating a new decision making method if there are not suitable methods in the MCDM library in which candidate methods are stored. This high ability system can not only help the DM find the most suitable method but can also guide him or her to reach the final decision by following the rigorous procedure of the selected method.

3.1 Adapted Decision Making Process

The decision making process illustrated in Figure 14 does not consider the selection of the most appropriate method for the design problem in the choice phase. It has an assumption that the desired solution can always be obtained which does not hold in many cases. In addition, the process does not take into account the scenario that there is not an appropriate decision making method available for the given problem. However, these issues often happen in the real decision problems and need to be resolved before the decision making proceeds. Figure 34 presents an adapted decision making process which employs the MIDAS to support the decision phase. One can see that the advisor plays a central role in evaluating the alternatives developed in the design phase and selecting the "best" solution which is going to be carried out in the implementation phase.

The adapted decision making process consists of four phases: intelligence phase, design phase, decision making phase and implementation phase. It begins with defining the design problem in the intelligence phase, where the customer requirements, design constraints and targets are identified and the Customer Requirements (CRs) are translated into Engineering Characteristics (ECs) by using the Quality Function Deployment (QFD) [2000] technique. The works done in this phase discover the essence of the design problem based on which the further design activities will be carried out. If the

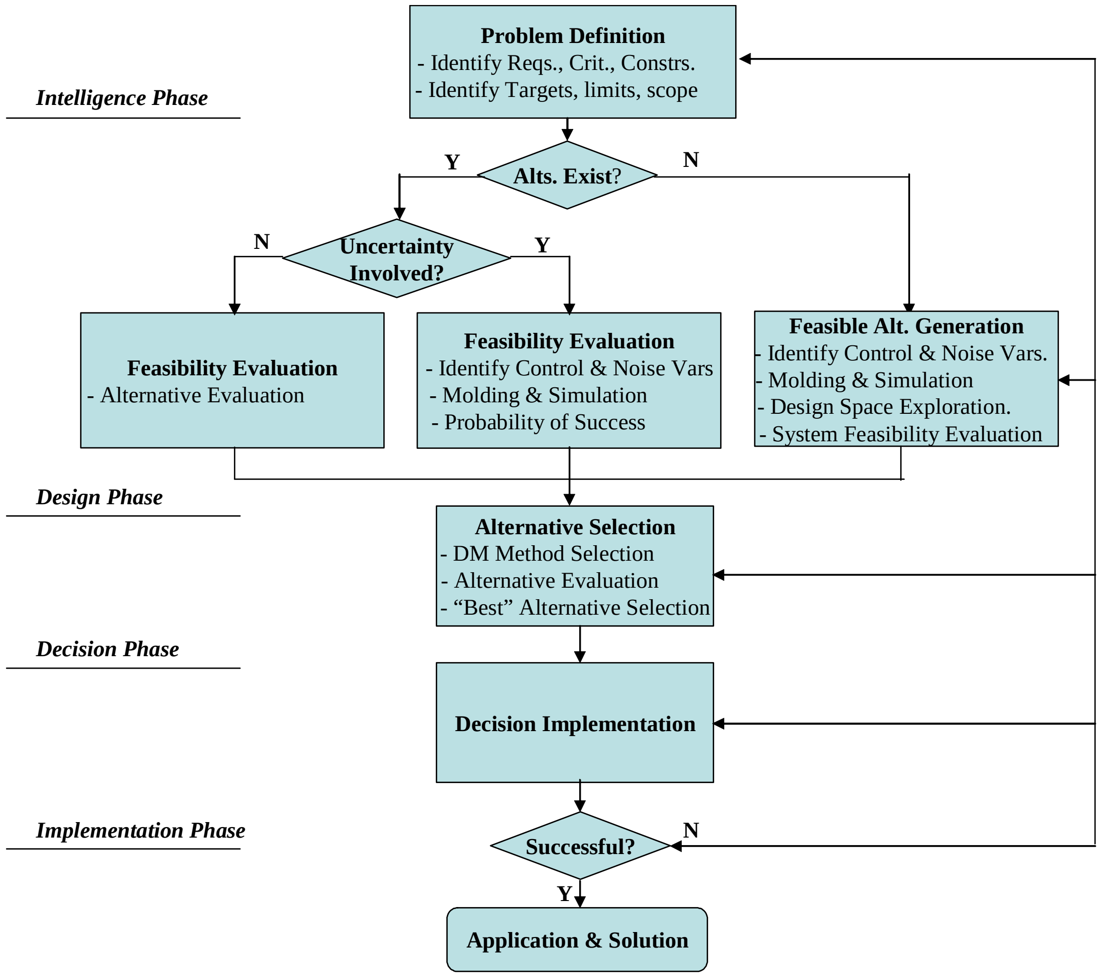


Figure 34: Adapted Decision Making Process

alternatives exist, the feasibility evaluation will be performed to determine whether the requirements and constraints defined in the intelligence phase are satisfied. This will be a pure concept selection problem. If the alternatives do not exist and there are only a set of requirements need to be satisfied, the alternative need to be generated, which results in a design problem. In this case, a generic design methodology referred to as the Technology Identification, Evaluation and Selection (TIES) [Kirby and Mavris, 2000; Mavris and DeLaurentis, 2000a] is employed for the design problems. This method encompasses a feasibility and viability examination process, explained in numerous technical publications. An approach called Unified Tradeoff Environment (UTE) [Baker, 2002] which uses combined sets of Response Surface Equations (RSEs) to visualize sensitivities of the key responses to the mission requirements, concept design variables, and technologies was also explored in this method. After the feasible alternatives are available for selection, the MCDM advisor takes over all the tasks in the decision making phase. An appropriate method will be selected and aid the DMs to make wise decision on selecting the "best" alternative. Finally, the selected alternative is obtained as the final solution for further implementation.

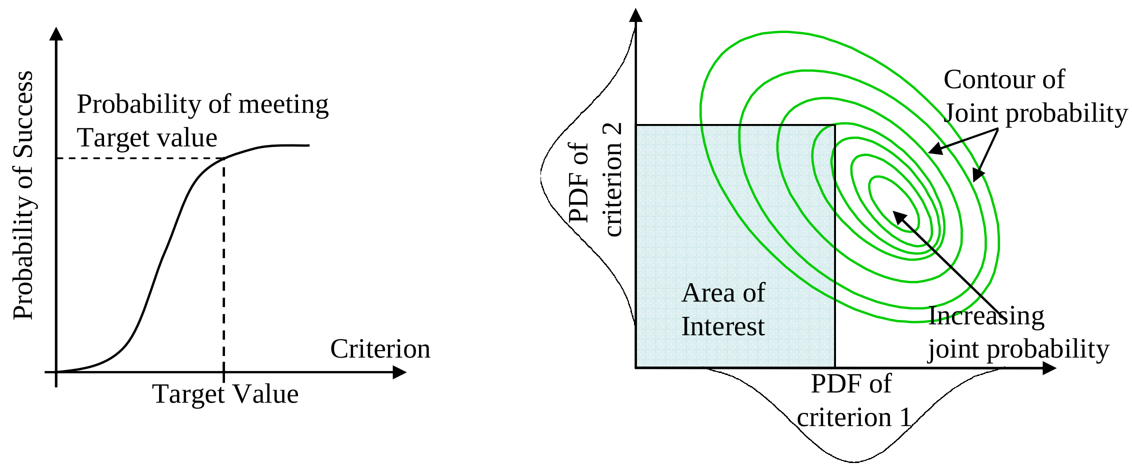
3.1.1 Feasibility Evaluation

By definition, a feasible solution is any solution in the feasible region of an optimization problem [Atallah, 1999; Feasible Solution, 2005] where the feasible region is the set of all possible solutions which satisfy all the constraints. For a concept selection problem, a feasible solution refers to the alternative that meets all the customer's requirements, constraints and targets. That is, a feasible alternative has to simultaneously satisfy all the criteria defined in the problem definition step, and any violation of a criterion will keep the alternative from being a feasible solution.

In the decision making process depicted in Figure 34, it is worth emphasizing the importance of the feasibility evaluation, because many existing decision making methods do not take feasibility into account, as a result, may suggest an infeasible alternative as the "best" solution. If this solution is implemented without recognizing its infeasibility, considerable cost or loss may be caused. This unsuccessful implementation often frustrates the DMs and leads to a fault conclusion that the method is incapable of handling the problem under consideration.

For instance, in the case described in section 1.1.1.2, TOPSIS suggests aircraft C as the “best” solution, but one can clearly see that this concept only has a safety of 0.2 which seriously violates the safety requirement - a minimum of 0.8. Hence, in reality no airline will buy this aircraft and risk their business. The paradox in this example may result in an assertion that TOPSIS is not a capable method to solve this problem since it selected an infeasible concept as the best solution. The assertion may not be prudent enough because the conclusion was drawn without considering the underlying reasons causing the undesired result. In fact, TOPSIS is a technique that does not take feasibility evaluation into account, so it attempted to select the “best” concept from all the alternatives no matter if they are feasible or infeasible. This observation discovers that TOPSIS considers that every alternative has some degree of probability to be selected as the final solution. Therefore, in the case that an alternative with better average goodness with regard to all criteria may still dominate the others while it violates one or more criteria, TOPSIS will select this alternative, an infeasible solution, as the best solution. This issue not only occurs to the TOPSIS technique, but may also happen to other decision making methods, such as AHP and OEC. Therefore, in order to obtain a desired solution for the concept selection problem, a feasibility evaluation of the alternatives is necessary to be performed before the decision making process proceeds.

In the adapted decision making process, one can see that before the MCDM advisor takes over all the decision making tasks, the feasibility evaluation is performed to screen the alternatives that will be sent to the decision making phase for the final selection. There are three scenarios that need to be considered when the feasibility is evaluated. One is that all the alternatives exist and no uncertainty needs to be concerned. In this scenario, each of the alternatives is examined, and a feasible alternative is required to satisfy all the constraints simultaneously. The second scenario is that all the alternatives are available and uncertainty exists in the problem. In this case, the control and noise variables are identified first, and then the distributions of the criteria for each alternative are obtained by using a modeling and simulation environment which represents a probabilistic analysis approach. The Cumulative Distribution Functions (CDF) of the criteria can be used to determine the feasibility of an alternative. To evaluate the feasibility of the alternatives, the PoSs of an alternative for the criteria need to be calculated based on the CDFs and specified target values of the criteria. If the PoSs of the criteria are greater than the given confidence levels, the alternative can be considered as feasible solution. The PoS of concurrently satisfying all the criteria can be obtained by using the JPDM technique. By utilizing this technique, the joint PoS for each alternative is calculated and the alternatives whose joint PoSs are greater than the prefefined threshold values are considered feasible. The CDF of a single criterion and joint probability density function of two criteria are illustrated in Figure 35. In the third scenario, there is no alternative available for selection, and the alternatives need to be developed using some design methods. The study presented in this document employs the TIES method to generate the alternative designs. During the alternative generation, one of the key steps is to use the JPDM technique to determine whether an expensive investigation of new technologies is necessary. This implies if an alternative is found infeasible, technologies need to be identified and infused to improve the system feasibility. Hence, the TIES method essentially consists of a feasibility examination process, the alternatives generated using this method are certainly feasible and no further evaluation of their feasibility is required.



a) Cumulative Distribution Function b) Joint Probability Density Function

Figure 35: Feasibility Evaluation Techniques

After the system feasibility evaluations are accomplished, the infeasible alternatives are removed from the alternative list and only the feasible alternatives are sent to the next decision making step. This operation ensures that in the decision making phase the methods selected by the advisor will suggest a feasible alternative as the best solution for the given problem since the infeasible alternatives have no probability to be selected.

3.2 Multi-criteria Interactive Decision-making Advisor and Synthesis process (MIDAS)

The goal of the multi-criteria decision making process is to select the "best" compromise solution from the feasible alternatives based on the evaluation of the given criteria. To obtain a desired solution, an appropriate decision making method has to be chosen first and then aid the DMs to solve the decision problem by providing the necessary guidance.

Since there are various decision making methods available, an effective approach should be developed to accomplish the method selection problem. An intelligent, knowledge-based advisor system referred to as Multi-Criteria Interactive Decision-Making Advisor and Synthesis process (MIDAS) is proposed in this study to fulfill the above tasks. For a given problem, the MIDAS process starts by identifying the characteristics of the problem and defining the decision maker's preference information. Then the advisor can use the knowledge and information present in knowledge base and rank the methods stored in method base in term of appropriateness score. If no method has a score greater than the threshold score, the advisor needs to find a way to suggest new method for the given problem. If more than one methods have a score greater than the threshold value, more information will be needed to narrow down the selection, otherwise the method with the highest score will be chosen as the most appropriate method and used to produce the final solution to the problem under consideration.

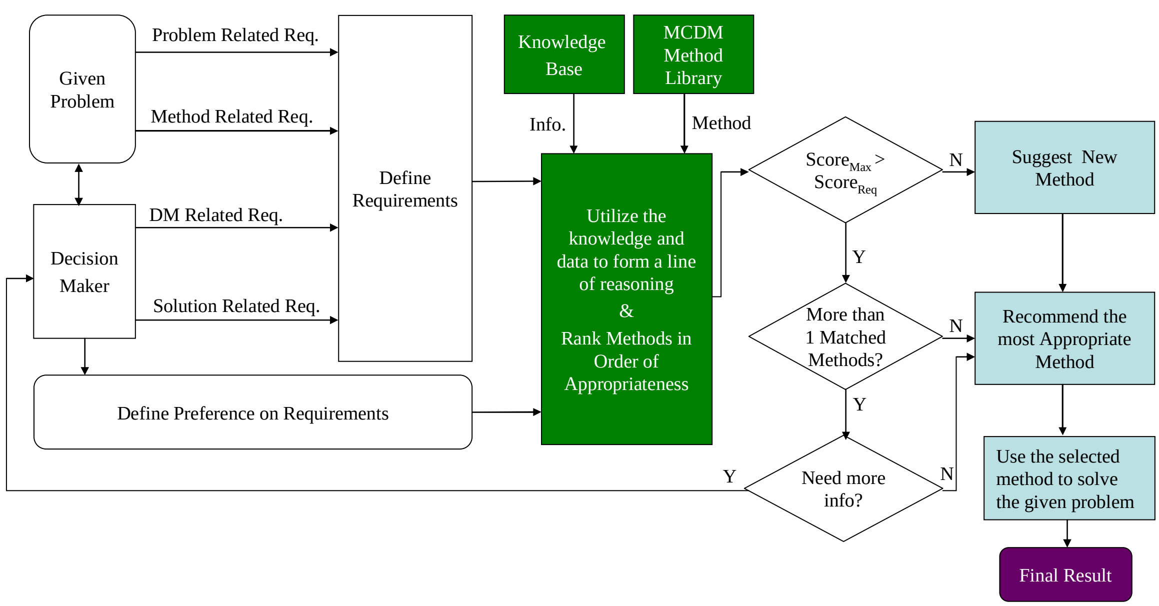


Figure 36: MIDAS Process

The MIDAS is designed to alleviate the DMs' burden of identifying the appropriate decision making method and support them in obtaining the high quality decision through the decision making process. It is capable of finding the most appropriate method for the decision making problem and then using the selected method to produce final result. In addition, it can provide guidance to generate new method if there is no method in method base is appropriate enough for the given problem. Apparently, MIDAS fills in the gaps existing in the current method selection approaches as shown in Figure 4.

3.2.1 Architectural Framework of MIDAS

As illustrated in Figure 36, the operation of MIDAS is supported by two data bases – knowledge base and method base, and a reasoning module that utilizes the information in the data bases to accomplish the method selection task. Thus, the MIDAS process is realized by a knowledge-based advisor system which consists of a user interface allowing the interaction between users and the system, an inference engine managing the execution of the system, an MCDM library storing the widely used decision making methods and a knowledge base providing the information required in the method selection process, as shown in Figure 37. To complete certain task, the user sends a query to the system through the user interface, and, based on the specified task, the system will request the necessary information from the user. After the user provides the information (inputs) to the system, the inference engine will analyze the inputs and utilize the information and knowledge stored in the knowledge and method bases to form a line of reasoning. Thus certain conclusion will be drawn for the original task query and the outputs will be presented to the user through the user interface. During the process, additional information may be required from the user so that iterations may occur in order to produce an explicit and convergent conclusion.

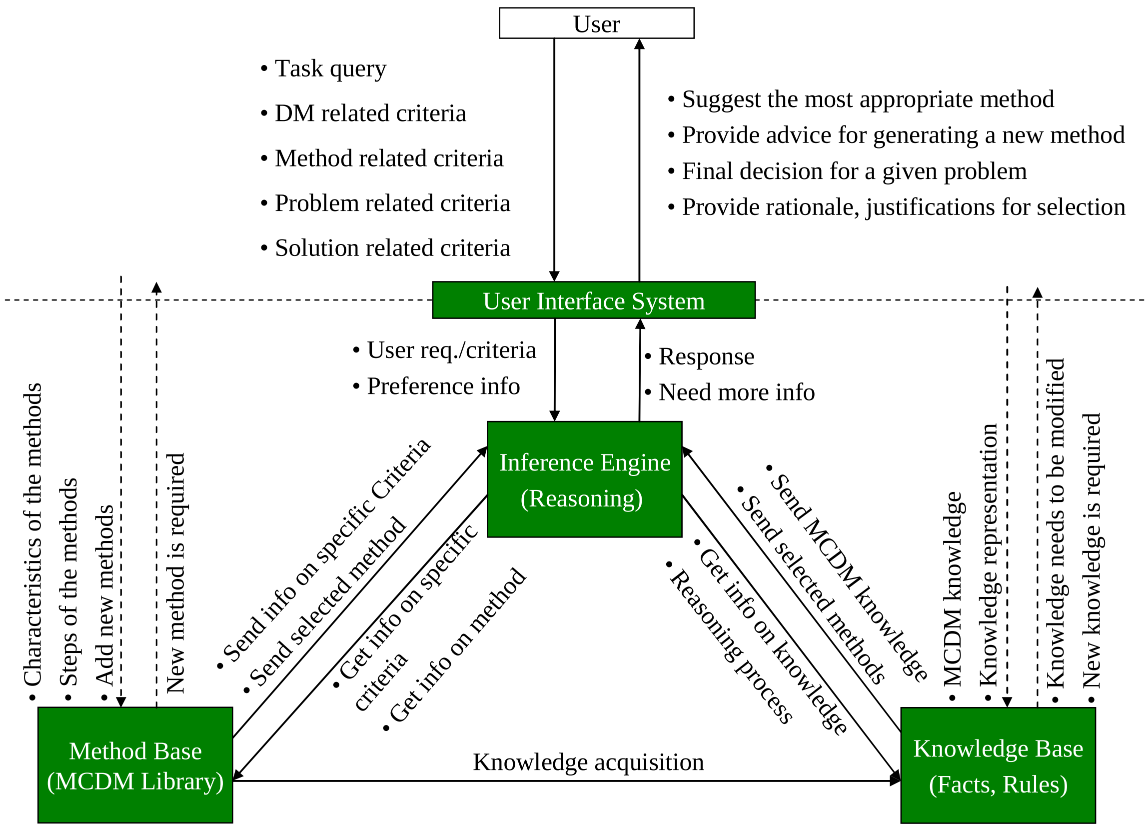


Figure 37: Architectural Framework of MIDAS

3.2.1.1 User Interface

The user interface system of the MIDAS allows the user to interact with the system to accomplish a certain task. First, the user informs the system through the user interface that there are certain tasks to be completed, such as selecting the most appropriate method, validating the decision made using another method and solving the current decision problem utilizing the selected method. After the system receives the task query, it will present the user a questionnaire with the decision options to the individual question.

To complete the process, the user is required to give the corresponding answers, select the desired options, and provide the supplemental information to the system as the inputs. Based on these inputs provided by the user, the advisor will perform the necessary analysis and inference, and finally the results will be displayed to the user through the interface. These activities can all be completed with the user-advisor interaction through the user interface system. Figure 38 shows the user interface of the MIDAS.

The advisor system is designed to interact with four types of users: 1) experts, who use the system in order to get a second opinion on a decision making problem, obtain aids in handling some tedious or difficult tasks that the computer is more efficient to deal with, or wish to find the reasons to reach a decision by following the system's reasoning process, 2) engineers, who need the advice supplied by the system to improve their decision quality and employ them to solve their specific decision problem, 3) students, who use the system to learn the knowledge about the problem solving procedure, reasoning process or some other subjects, where the advisor plays the role of a tutor, or 4) developers, who maintain and develop the system, such as adding new advanced decision making methods to the system when they emerge.

The user interface system provides a convenient communication between users and the advisor system through various graphic screens. The user is able to easily manipulate the system by inputting the required information and commands using the user interface. The advisor responds to the user by outputting some data and graphs through the interface to complete the interaction.

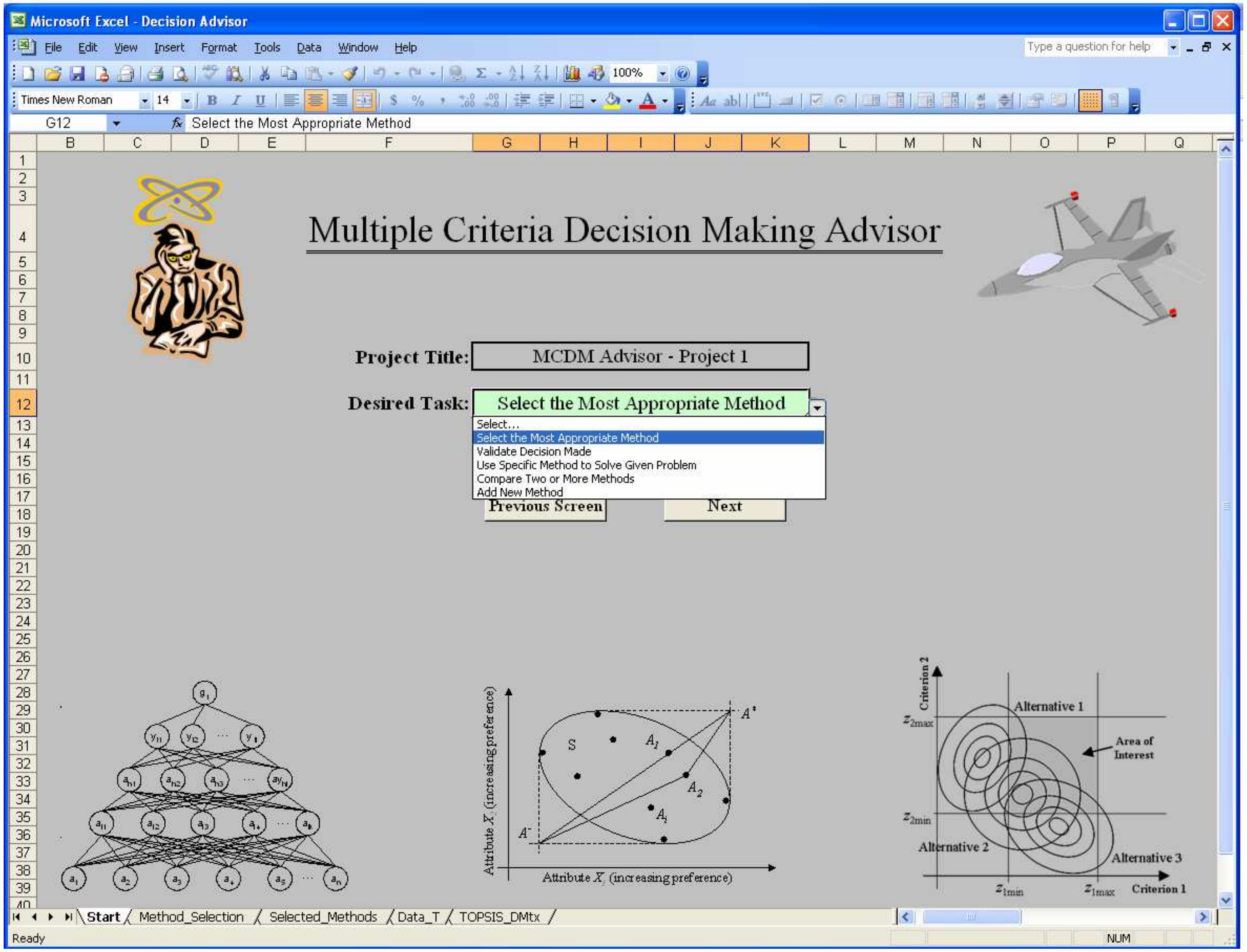


Figure 38: MIDAS User Interface

3.2.1.2 Inference Engine

The inference engine of the MIDAS system is the control mechanism that applies the information present in the knowledge base and method base to the task-specific data to arrive at a conclusion through reasoning. In the reasoning process, the inference engine organizes and controls the steps taken to solve the problem, manipulates the knowledge contained in the knowledge and method bases and handles the execution of the system. It first interprets the inputs that the user entered through the user interface in order to determine which rules or facts will be applied to the current problem. This is often accomplished by the application of statistical methods or pattern matching methods. After receiving the query from the user, it manipulates and uses information in the knowledge base to form a line of reasoning, and then support the system to produce the conclusion for the original task query.

The inference engine is responsible for managing the execution order of the various tasks, deciding when and in which order the data of the knowledge base will be used. It is capable of evaluating the alternative search paths and providing insights derived from the knowledge base. Also, the inference engine is able to maintain a consistent representation of the emerging solution.

There are two typical inference techniques that the inference engine uses: forward chaining or data driven inference and backward chaining or goal driven inference. In the method of forward chaining, one proceeds from a given situation toward a desired goal, adding new assertions along the way, while in the backward chaining, one starts with the desired goal and attempts to find evidences that support the goal. The two methods have their own advantages and disadvantages. Table 3 presents an example of a typical situation that the two inference methods are used in maintaining the temperature of an enclosure between two limits. Although both methods are valid, the forward chaining inference engine is more direct, especially when the situation becomes more complex and involves several variables. When the temperature changes new data is received - this event may be used to trigger the rule. With a backward chaining inference engine, one would have to constantly check if it is too hot or too cold [Inference Engine, 2005]. This will increase the computational cost and highly decrease the control efficiency.

In the study presented in this document, both inference techniques are utilized. When selecting the most appropriate decision making method, the forward chaining is used because the selection start from the fragmentary inputs which reflect the characteristics of the given problem. Then, based on the situation (all the inputs), the desired goal (selecting the most appropriate method) will be reached by employing the forward chaining reasoning process. On the other hand, in the case of providing the advice for generating new methods, the backward chaining is utilized since the advice is obtained by examining the properties of the problem and the candidate methods. That is, the desired goal (finding the capable method that can solve the problem at hand) needs to be supported by some evidences (capabilities required to handle the problem).

Table 3: Comparison of Two Inference Engines

Step	Backward Chaining	Forwarding Chaining
1	It is too hot.	The temperature has changed.
2	Why?	Check if the heater should be changed
3	Is the heater on?	
4	Yes.	
5	Turn it off (Repeat for "It is too cold")	

3.2.1.3 Knowledge Base

The knowledge base is the core of the advisor system. Its main purpose is to provide the basis of the system - the connections between ideas, concepts, and information that allow the inference engine to perform an accurate evaluation of a problem [Boss, 1991]. In the knowledge base the facts and rules are stored in some format, which include both factual and heuristic knowledge and support the judgment and reasoning of the inference engine. Factual knowledge is the knowledge that is "widely shared, typically found in textbooks or journals, and commonly agreed upon by those knowledgeable in the particular field" [Feigenbaum et al., 1993]. For example, "AHP is good at handling the decision making problem with hierarchical attributes" is a piece of factual knowledge. On the contrary, heuristic knowledge contains special knowledge that is less rigorous, more experiential and more judgmental. This type of knowledge is rarely discussed and is largely individualistic. For instance, a heuristic knowledge can be "if uncertainty needs to be captured, try to use the JPDM technique". These rules can be in the form of complex structure or an interconnected group of rules [Curry and Moutinho, 1991].

In the advisor system, the knowledge acquisition is performed carefully in order to obtain an accumulation of high-quality knowledge. Knowledge is acquired from expert and other documented sources. The knowledge acquisition process is expected to get as much knowledge as possible for the problem since the more knowledge existing in the knowledge base the more competent the advisor system is. Once the knowledge is endowed to the system, necessary operations are taken to ensure the quality of the knowledge. These operations include the evaluation, validation and verification of the acquired knowledge [Parsaye, 1988].

After the knowledge is obtained through the knowledge acquisition process or is elicited by the expert, it needs to be organized and represented in an appropriate manner. There are several ways to represent knowledge, such as a representation method, product rules, formal logic, object-attribute-value, and so on. Among these methods, production rules may be the most popular way of knowledge representation because almost every piece of knowledge can be written as a rule. In addition, they are simple and efficient in solving some problems, for example, diagnosing problems. The rules have the following form:

IF

Conditions (assumptions)

THEN

Action (conclusion)

The above form implies that when the conditions are satisfied then a conclusion is arrived at or an action is triggered. Figure 39 illustrates a simple example to show how a rule in the knowledge base works. For this example, there are three rules which represent the knowledge associated with the user's preference information:

- **Rule 1:** If user's preference information is represented by relative weight, the candidate methods are TOPSIS, AHP and JPDM
- **Rule 2:** If a utility function is employed to show the user's preference, the methods that may be appropriate to solve the problem are EUT and MAUT
- **Rule3:** If the user utilizes the relative weight to express their preference over the criteria and assign preemptive weight to certain criteria, then goal programming is likely a suitable method to handle the problem under consideration.

When the user has a problem at hand and wants to find an appropriate method to produce a desired solution, he or she may send a request to the MIDAS to fulfill this task. The advisor system will present a list of questions to the user in order to capture the essence of the problem. One of the questions may be: How is the preference information over the criteria represented? Based on user's answer to the question, the advisor will find the right rule from the knowledge base to draw a conclusion, thus one or more decision making techniques will be selected as candidate for further examination. For instance, if a user's answer to the question is "relative weight", the condition of rule 1 is satisfied. Sequentially, the advisor will fire this rule and draw a conclusion that TOPSIS, AHP and JPDM are the candidate methods for the given problem. Similarly, if a user's answer is "utility function", then the advisor will find rule 2 is satisfied and then consider EUT and MAUT as the suitable techniques to deal with the problem by obeying this rule.

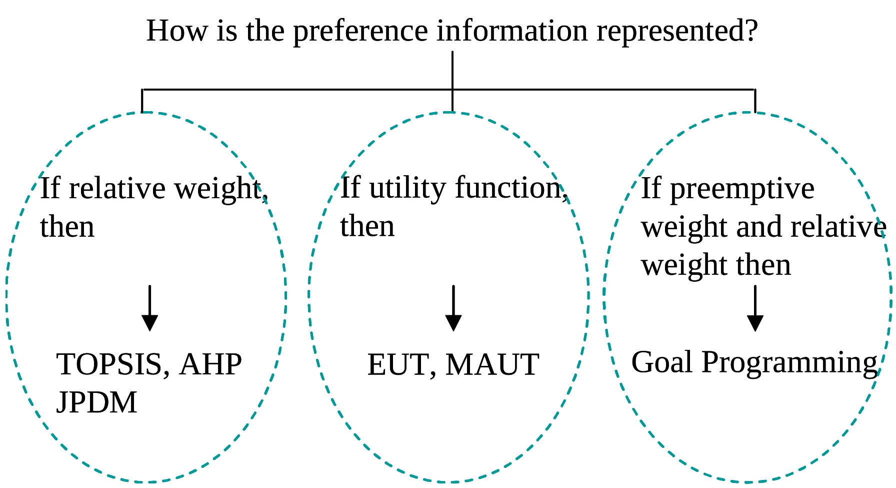


Figure 39: Example of Decision Rules

The rules that compose the knowledge base should be concise and have clear meaning, and be tangible to every stage of operation. Each rule describes a certain knowledge case and thus the represented knowledge is characterized by independence and a high level of transparency. The necessary knowledge associated with selecting the most appropriate MCDM method, validating the decision made and generating a new decision making method is formulated in the knowledge base and stored as a set of rules.

3.2.1.4 Method Base (Method Library)

The method base, also referred to as a MCDM library, is the other important component of the MIDAS system which can provide knowledge to support the reasoning process of the system. The library stores the information associated with a number of widely used MCDM methods. The method which is the most suitable to handle the problem under consideration is selected from the library and then provides the guidance to the DMs to facilitate the problem solving procedure. In this study, each method is represented by two sets of data: one indicates the characteristics of the method; the other provides the problem solving steps of the method. The characteristics of the MCDM methods are divided into four classes: DM related, method related, problem related and solution related characteristics, and each category of characteristics is independent of the others.

Decision Maker Related Characteristics

DM related characteristics are those which reflect the DM's level of knowledge, ability and preference on selecting a MCDM method to solve the given problem. The choice of these characteristics depends on the DM's previous experience or intuition with the method, or depends on the judgment or opinion obtained from the previous work with the method [Roman et al., 2004]. Some of the characteristics are related to the DM's knowledge about a specific method, and some of them are associated with the DM's time availability, that is, how much time the DM would spend to arrive at the final decision. In addition, these characteristics indicate the DM's willingness to accept the assumptions and limitations of the method. And these characteristics also include the ones that reflect the DM's preference form. For example, some DMs would express their preference information in a ranking or scale form, but some of them desire it would be quantitative data. This difference in preference form is dependent upon the individual DM's desire or how far along in the design process the decision is being made.

Method Related Characteristics

Method related characteristics play a central role in the selection of the most appropriate MCDM method. The reason is that the characteristics of the method determine what information the method needs to construct the decision model, what aspects of the given problem can be taken into account and how the decision is made, therefore, eventually it determines the decision making quality of the problem. These characteristics are those relating to the solution process of the MCDM method. Some of them are listed below:

- MADM, MODM or MCDM: Is the method able to handle the MADM problem or MODM problem, or both (MCDM)?
- Feasibility evaluation: Does the method evaluate the feasibility of the alternatives?
- Preference representation: How is preference information over the criteria represented? Is it represented by relative weight, utility function or another preference function (e.g. class function and loss function)?

- Input requirements: What input data are required by this method (decision matrix, comparison matrix or response surface equation)?
- Uncertainty: Is the method able to capture the uncertainties existing in the problem?
- Dynamic behavior: Can the method handle a problem with dynamic behavior, such as changing in attributes or requirements?
- Objective or subjective criteria: Can the method handle the objective and/or subjective criteria?
- Decision rule: What metric does the method use to rank the alternatives, relative importance, utility, POS or other metrics?
- Discrete or continuous data: Can the method deal with the discrete and/or continuous parameters?
- Hierarchical architecture: Can the method handle the problem with multi-level criteria?
- Implementation: What hardware and software are required to implement the method? How easy and how long is the implementation?

Problem Related Characteristics

Problem related characteristics are those depending upon the real decision making problem, such as the number of alternatives, attributes/objectives, and constraints, the amount of information available, and whether it is linear or nonlinear. That is, the problem related characteristics address the features of problems associated with the alternatives, attributes, design space and feasible space. Below are some example problem related characterizes:

- Alternative: Do the alternatives exist for the problem?
- Attribute: Do the attributes or objectives used to evaluate the alternatives have multiple levels? Can they be quantified?
- Design Space: Are the design variables discrete or continuous? Is there any soft constraint in the problem?
- Feasible Space: Dose any alternative have certain probability of being selected as the "best" solution?

In order to obtain a desired solution for the problem under consideration, an MCDM method must be able to address the key characteristics of the problem. This implies that the method selection is based on the concept that the characteristics of the method should "best" satisfy the applicable problem related criteria, otherwise, the application may yield a misleading result. Figure 40 presents an example to demonstrate this concept, where a total of 19 characteristics are identified and 17 MCDM are in the method base for selection. As an example, the method of Electre is characterized by the characteristics of the "subset" AND "normalized scale". The "NOT" nodes indicate the exclusion of one of the characteristics from another in any MCDM method.

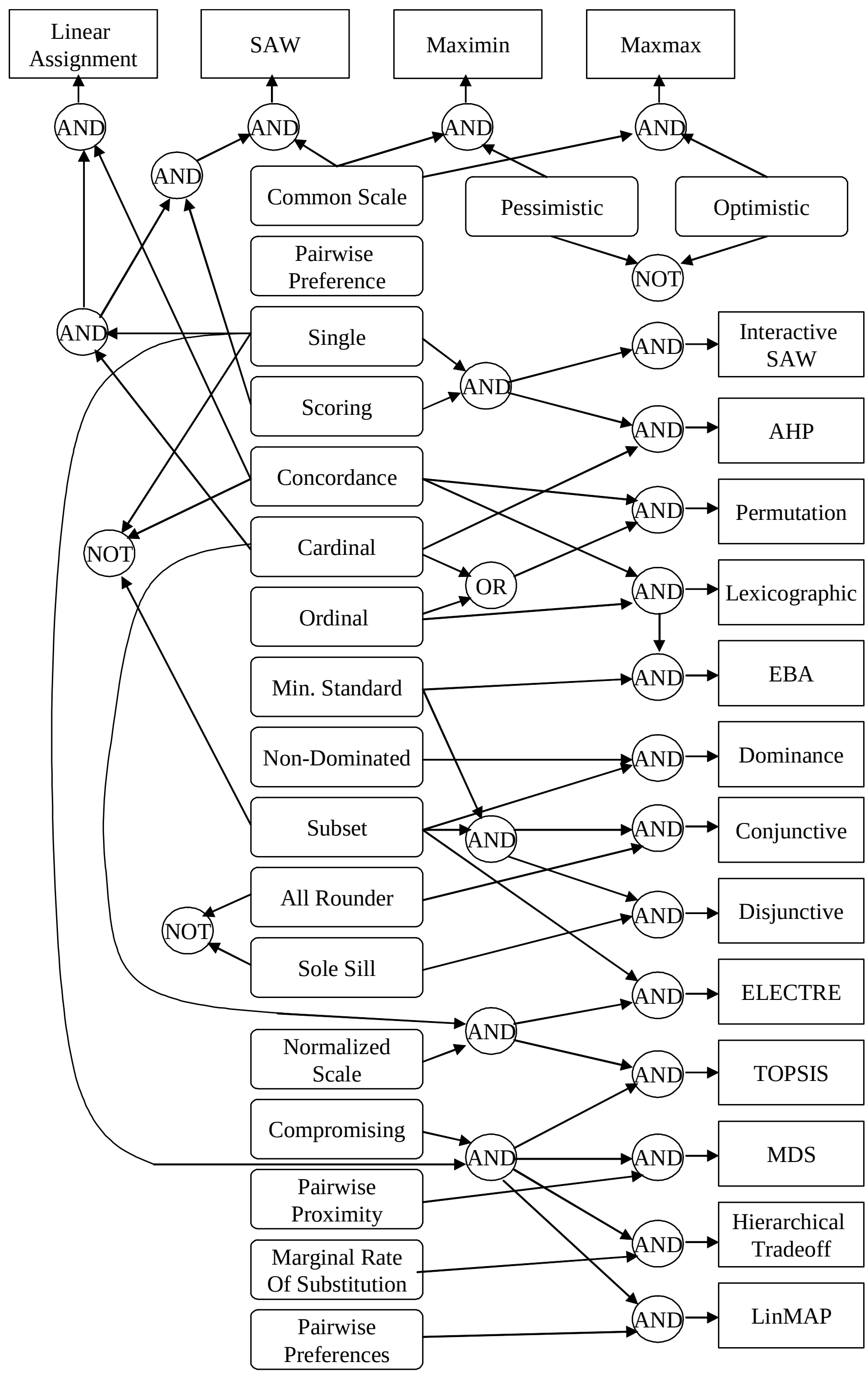


Figure 40: Example of Relationships between Method and their Characteristics [Poh, 1998]

Solution Related Characteristics

The choice of one MCDM method over another is related to the appropriateness of the results obtained from the use of that method for the problem. These characteristics are captured in the solution related characteristics which are related to the types of solution produced by the methods. For example, the solutions obtained from different methods have different sensitivity (how sensitive are the results to the changes in weighting, or selection of a datum point?) and robustness (how robust are the results to the changes in preference information?).

Problem Solving Procedure

Once the most appropriate method is found, the solution of the decision problem needs to be obtained using the method. In some case, one may not know to utilize the selected method to formulate the given problem and create the corresponding solution. Since all methods have a systematic model, they have a step by step problem solving procedure. Therefore, if the problem solving procedure of the selected method can be provided to the users, they can follow it to reach the final solution. The method base also contains the problem solving procedure of each method in the library which can be used by the MIDAS to provide guidance to the users to facilitate the decision making procedure. For example, below is the problem solving procedure of TOPSIS technique.

Step 1: Construct the decision matrix for the problem. The element of the decision matrix y_{ij} represents the value of attribute j with respect to design alternative i.

Step 2: Normalize the decision matrix whose elements are given by:

$$z_{ij} = \frac{y_{ij}}{\sqrt{\sum_{i=1}^{n} y_{ij}^2}} \quad i = 1, \cdots, n; \quad j =, \cdots, k \tag{24}$$

Step 3: Formulate the weighted normalized decision matrix whose elements are given by

$$x_{ij} = w_j z_{ij} \quad i = 1, \cdots, n; \quad j =, \cdots, k \tag{25}$$

Step 4: Define the set of "benefit" attributes J^* and the set of "cost" attributes J^-

Step 5: Define the positive ideal solution a^* and the negative ideal solution a^- as:

$$\begin{aligned} a^* &= \left\{ \left(\max_i x_{ij} \middle| j \in J^* \right), \left(\min_i x_{ij} \middle| j \in J^- \right) \middle\| i = 1, \cdots, n \right\} \\ &= \left\{ x_1^*, x_2^*, \cdots, x_k^* \right\} \end{aligned} \tag{26}$$

$$\begin{aligned} a^- &= \left\{ \left(\min_i x_{ij} \middle| j \in J^* \right), \left(\max_i x_{ij} \middle| j \in J^- \right) \middle\| i = 1, \cdots, n \right\} \\ &= \left\{ x_1^-, x_2^-, \cdots, x_k^- \right\} \end{aligned} \tag{27}$$

Step 6: Calculate the separation of a design to the positive ideal solution S_i^* and to the negative ideal solution S_i^- measured by the n-dimensional Euclidean distance in the attribute space

$$s_i^* = \sqrt{\sum_{j=1}^{k} (x_{ij} - x_j^*)^2} \quad i = 1, \cdots, \mathrm{n} \tag{28}$$

$$s_i^- = \sqrt{\sum_{j=1}^{k} (x_{ij} - x_j^-)^2} \quad i = 1, \cdots, \mathrm{n} \tag{29}$$

Step 7: Calculate the relative closeness of each design to the ideal point

$$C_i^* = \frac{S_i^-}{S_i^- + S_i^*} \quad i = 1, \cdots, \mathrm{n} \tag{30}$$

Step 8: Rank the alternatives based on the magnitude of closeness C_i^*. If $C_i^* > C_j^*$, then a_i is preferred to a_j

This problem solving procedure can be invoked by the MIDAS when the method is selected for the given problem. In order to be able to handle the most of the decision making problems, the methods in the MCDM library are the typical method which is widely used in the current multi-criteria decision making realm. As the complexities of the decision making problems increases, new advanced methods with improved ability are intrigued to continuously emerge with time. Therefore, it is not possible to include these advanced decision making methods in the MCDM library at the time when the advisor system is developed. To keep the system from being obsolete, the new methods are allowed to be added into the MCDM library for further use, eventually increasing the capability of the advisor.

3.2.2 Capabilities of MIDAS

Engineering decision making is a process that utilizes the available information and certain techniques to arrive at a desired solution. Typically, the available information is problem related and is used to derive the requirements and define the constraints of the decision problem, which is one of the critical steps in the decision making process. With the problem well defined, the other important task is to formulate the problem by using an MCDM analysis model which can capture the essence of the problem. Since various decision making techniques are available and each of them employs a different model to represent the problem, the method most suitable to solve the given problem needs to be identified in order to obtain a desired solution for the problem. The MIDAS presented in this study can select the most appropriate decision making technique for the problem under consideration, guide the users to solve their specific problems, validate the decision made, and help in generating a new method that is suitable to handle the problem under consideration if no existing method is recommended.

3.2.2.1 Decision Making Method Selection

Typically, there is not a universal method that can handle all types of the decision making problem since different problems have various issues that need to be addressed. One specific decision making method is usually suitable to solve one class of problems with certain characteristics. This leads to the fact that different methods often create different solutions for the same problem. Therefore, selecting the most appropriate method is a key step in the decision making process to make successful decision.

Basically, a decision making problem has a few characteristics, such as characteristics associated with uncertainty, feasibility and hierarchy. A decision making technique may not handle the problem because it does not have capabilities to deal with some aspects of the problem. For example, TOPSIS does not take in to account uncertainty that often exists in some problems, AHP is not able to deal with the dynamic behavior of the problems, and JPDM can not accurately represent the DM’s preference information [Li et al., 2004]. If a user has no knowledge about these decision making methods, it is difficult for him/her to pick the method suitable for the current problem. On the other hand, if a user has a decision making method in mind but that technique is not suitable to deal with the problem at hand, he may end up with a misleading solution by utilizing that method. Therefore, it is necessary to find a way to select the most suitable decision making technique for the problem under consideration.

On the other hand, different decision making techniques have their own requirements, assumptions and limitations. For examples, different techniques require different input data, preference information and decision rules. Hence, if a problem with certain properties is solved using a decision making technique which is designed for this type of problem or whose characteristics best meet the characteristics of this type of problem, a more appropriate solution can be obtained. This is the concept that the MIDAS uses to select the most suitable decision making technique. To find the best appropriate decision making technique for the given problem is one of the abilities that the MIDAS can accomplish.

Table 4 shows six techniques that are decomposed in terms of their characteristics and requirements. In this table, it can be seen that TOPSIS does not perform the feasibility evaluation, it can only be used to deal with the product selection problem, and it needs a decision matrix to help it organize the input data. The relative weight represents its preference information and is given in advance. TOPSIS is able to handle the discrete

Table 4: Characteristics of Decision Making Techniques

	TOPSIS	**AHP**	**EUT**	**JPDM**	**MAUT**	**Goal Programming**
Feasibility Check?	No	No	No	Yes	No	Yes
Optimization/ Selection?	Selection	Selection	Selection	Both	Selection	Optimization
Deterministic/P	Deterministic	Deterministic	D/P	Probabilistic	Deterministic	Deterministic
Input Data Available	Decision Matrix	Comparison Matrix	N/A	N/A	N/A	N/A
Complexity	Single Level	hierarchical	Single Level	Single Level	Single/Hierarchic	Single Level
Preference	Relative Weight	Relative Weight	Utility Function	Relative Weight	Utility Function + Relative	preemptive weights +Relative Weight
Weight	Given	Calculated	N/A	Assigned	Assigned	Assigned
Info. Req.	N/A	N/A	Probabilities + Utility Function	Interest of Area	Utility Function	Goals
Decision Rules	Closeness to Ideal Solution	Ordinal Ranking	Maximize Utility	Maximize POS	Maximize Utility	Minimize the viration to the set of goals
Visulization	Yes	Yes	No	Yes	No	No
Dynamic/Static	Static	Static	Static	Static	Static	Static
Discrete/Cont.	D/C	D/C	C	D/C	D/C	C
Complete/Incomp	Complete	Complete	Incompete	Incomplete	Complete	Complete

attributes, but can not be used to solve the problem with dynamic behavior. It also can be seen from this table that the TOPSIS evaluates the alternatives based on the decision rule of maximizing the closeness to the ideal solutions. Therefore, it implies that TOPSIS is a good method for a decision making problem with single attribute level, weighting preference and discrete attributes. It is not an appropriate method for the problems that need uncertainty analysis and dynamic consideration.

To select the most appropriate decision making technique, the advisor starts from asking the DM some questions, which are related to different aspects of a decision making problem. For each question, the advisor provides two or more options for the DM to choose as the answers to the corresponding questions. Table 5 lists the options of the answer to some of the questions. After the questions are answered, the advisor will analyze this information and rank the methods in order of appropriateness index which is given by Equation (31). Finally the methods with appropriateness index greater than the threshold will be recommended as appropriate methods to solve the problem under consideration.

$$AI_i = \frac{1}{n}\sum_{i=1}^{n} w_i I_i \tag{31}$$

where n is the number of criteria used to examine the characteristics of decision making methods or the given problem. Each such characteristic is corresponding to one examination criterion which has two or more values, as shown in Table 5. $W = \{w_1, w_2, \cdots, w_n\}$ is the weighting vector on the examination criteria. $I_i = \{b_i, b_{2,} \cdots, b_n\}$, and b_i is defined as:

$$b_i = \begin{cases} 1 \text{ if } c_{ji} = a_i \\ 0 \text{ if } c_{ji} \neq a_i \end{cases} \quad \mathrm{i}=1, 2, \cdots, \mathrm{n};\ \mathrm{j}=1, 2, \cdots, \mathrm{m} \tag{32}$$

where a_i is the value of the i-th characteristic of the decision problem, and c_{ji} is the value of i-th characteristic of the j-th method in the method library which stores m methods for selection.

Table 5: Options of the Answers to the Questions

	1	2	3	4	5	6
Feasibility Check?	Yes	No				
Optimization/ Selection	Selection Only	Optimization Only	Optimization + Selection			
Uncertainty Analysis?	Yes	No				
Risk Analysis?	Yes	No				
Input Matrix Available	Decision Matrix	Comparison Matrix	None			
Complexity	Hierarchical	Single	Hierarchical + Single			
Preference	Relative Weight	Utility Function	Relative Weight + Utility Function	Class Function	None	
Weight	Given	Assigned	Calculated	None		
Info. Required	Interested of Area	Utility Function	Goals	Probabilities + Utility Function	None	
Decision Rules	Maximize Clossness to Ideal Solution	Maximize the Utility Function	Maximize POS	Ordinal Ranking	Minimize the Variation to the Set of Goals	Minimize the Aggregate Function
Visulization	Yes	No				
Dynamic/Static	Danamic	Static				
Subjective/Obj. Varable	Subjective Only	Objective Only	Subjective + Objective			
Complete/Incomp.	Complete	Incomplete				

3.2.2.2 Decision Validation

A DM is usually familiar with one or more decision making methods, and thus he or she tends to use these method to deal with any decision problems under consideration. As one can see, a decision method good at handling one type of problem usually incapable of handling other types of problems. Therefore, the use of the decision making methods that the decision maker is familiar with often produces inappropriate decisions, as a result, results in misleading solutions. This intrigues that the decision validation should be performed before the decisions are implemented. The MIDAS is able to validate the decisions made by using a specific method.

The validation process is similar to the method selection process except the decision solution is known in advance. In order to validate the decisions, one must verify the decision making method used is appropriate. At this point, the selection of the most appropriate method becomes one part of the decision validation problem. First, the advisor asks the DM to answer some questions related to the problem he/she solved. Based on the DM's inputs, the advisor will utilize the information and data in the knowledge base to determine the most appropriate method existing in the MCDM library. If the method suggested by the advisor is the same as the one the DM ever used to solve the problem, it implies that the decisions made may be valid. Otherwise, if the advisor recommends a different method from the one the DM used, it indicates the decisions made are not appropriate and need to be refined using the selected method. The MIDAS can also provide guidance to the DM in the problem solving procedure when the selected method is utilized. This capability allows the DM to make decision using the specific method without know how this method works.

3.2.2.3 Decision Making Using a Specific Method

After a decision making method is selected as the most appropriate method to deal with the problem under consideration, the DM will employ this method to formulate the problem and produce the desired decision solution. However, there are various methods out there and it is impossible for DM to know each of them, therefore, the decision maker may not know the selected method well enough and is not able use it to get the problem solved. This situation requires that the DM is allowed to use the method to obtain the final solution without knowing the method. This requirement results from the fact that it is not possible for the decision maker to understand every method and it is not worth learning the method and programming only for solving one specific problem. Otherwise, it will be time-consuming, inefficient and may cause more errors due to the limited knowledge and experience about the method.

The MIDAS is capable of providing guidance for the DM when a specific method in the MCDM library is selected. For each method in the MCDM library, the advisor has an explicit step by step problem solving procedure for the DM to follow. This procedure can be completed through the corresponding user interface. To go through the procedure, the DM is only required to input some basic information associated with the problem, such as the number of the alternatives, the number of the attributes, and the preference information. Then the decision maker can follow the explicit guidance provided by the advisor to reach the final solution. For example, Figure 41 depicts the step by step decision making process using the TOPSIS technique. The only information needs to be inputted by the DM is the data highlighted in blue color. Once the necessary data are obtained in step 1, the following steps are accomplished by simply clicking the corresponding command button. By executing these actions, the steps of the TOPSIS technique described in section 3.2.1.4 will be automatically achieved. And the final results of the problem will be presented to the user through the friendly user interface. This simple operation allows the DM makes decisions using the TOPSIS technique without knowing how the method. This type of user interface exists for each method, and the new interfaces can be developed for the new methods which are added to the library to increase the MIDAS's decision making capability.

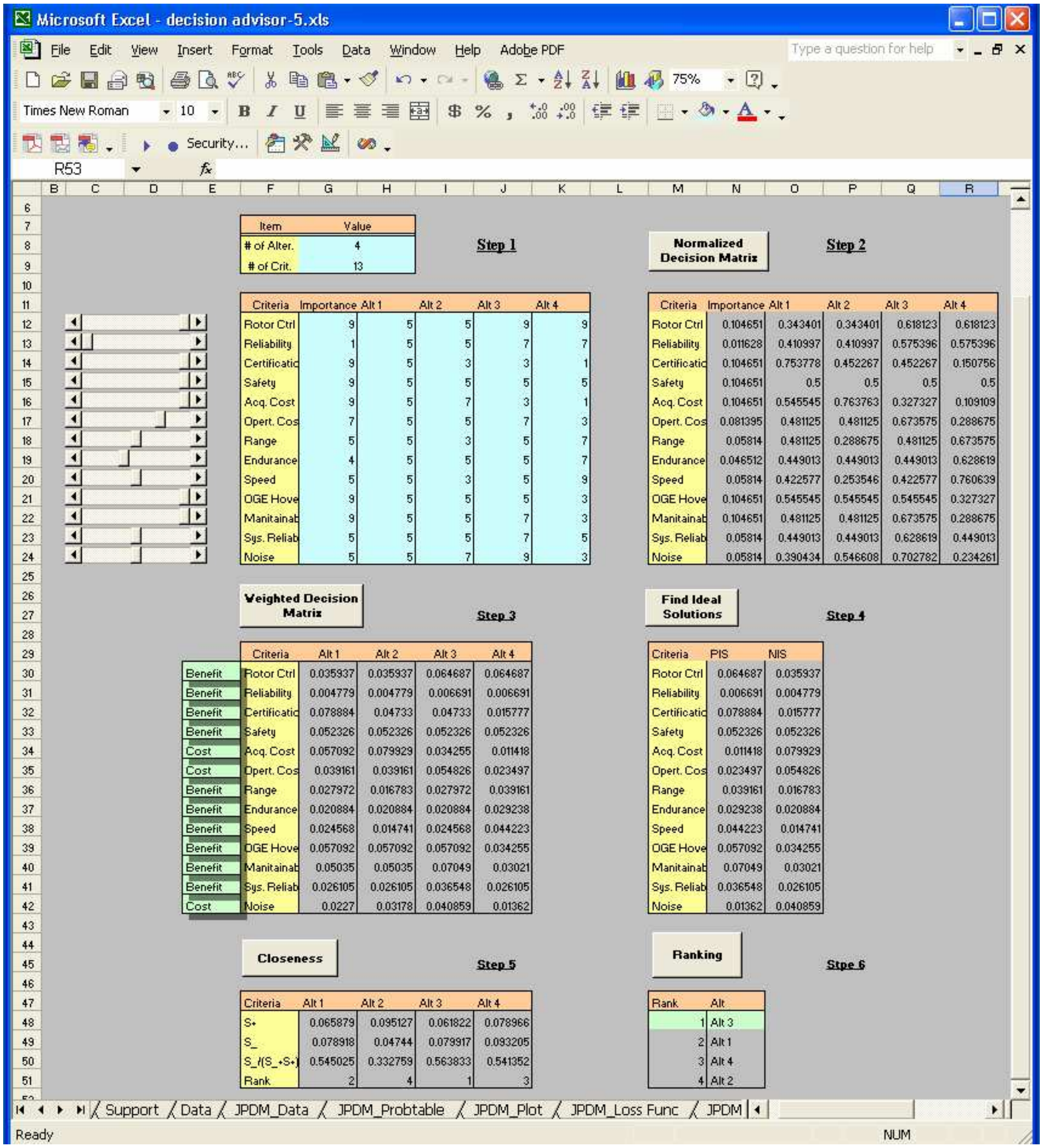


Figure 41: User Interface for Step by Step Problem Solving Procedure of TOPSIS

3.2.2.4 New Method Generation

In some cases, the decision advisor may not be able to find an appropriate method for the given problem from the MCDM library. This may occur when the problem is more complicated than the types of the problems typically considered by the advisor, or just because of the limited number of the methods in the MCDM library. This issue inevitably happens when the advisor deals with some type of problems because the MIDAS is not able to include all the existing methods in the library.

Fortunately, the MIDAS is capable of handling this issue. When the advisor can not find an appropriate method for the problem under consideration, it will analyze the answers and information that the DM provided for the problem. Based on the analysis, the advisor will find out what capabilities are required for a method to be fulfilled to deal with the problem through the morphological matrix shown in Table 6. Then it will give the DM some advice for solving the current problem. The advices can be to suggest the DM to find an existing decision making method with some certain capabilities or characteristics, which is not in the MCDM library. If there is not such existing technique or the expected technique can not be found by the DM, the advisor will suggest the DM to create a new technique capable of handling the current problem and the advice provided by the advisor will act as the hints for developing the new technique. These hints include the suggestion of combining two or more existing techniques in the library to generate an advanced new technique with higher abilities.

Table 6: New Method Generation

	1	2	3	4	5	6
Feasibility Check?	Yes	No				
Optimization/ Selection	Selection Only	Optimization Only	Optimization + Selection			
Uncertainty Analysis?	Yes	No				
Risk Analysis?	Yes	No				
Input Matrix Available	Decision Matrix	Comparison Matrix	None			
Complexity	Hierarchical	Single	Hierarchical + Single			
Preference	Relative Weight	Utility Function	Relative Weight + Utility Function	Class Function	None	
Weight	Given	Assigned	Calculated	None		
Info. Required	Interested of Area	Utility Function	Goals	Probabilities + Utility Function	None	
Decision Rules	Maximize Clossness to Ideal Solution	Maximize the Utility Function	Maximize POS	Ordinal Ranking	Minimize the Variation to the Set of Goals	Minimize the Aggregate Function
Visulization	Yes	No				
Dynamic/Static	Danamic	Static				
Subjective/Obj. Varable	Subjective Only	Objective Only	Subjective + Objective			
Complete/Incomp.	Complete	Incomplete				

CHAPTER IV

IMPLEMENTATION OF THE MIDAS

The focus of this chapter is to apply the Multi-Criteria Interactive Decision-Making Advisor and Synthesis process developed in Chapter III to a Personal Air Vehicle (PAV) concept selection problem as a proof of implementation. In this problem, the advanced PAV concepts need to be derived from three baseline rotorcraft configurations, and the advanced concept with the highest viability was selected as the best concept measured by the given criteria under a defined uncertainty model. The PAV concept selection can be accomplished in the decision phase as shown in Figure 34 and is a well-suited application for the MIDAS since the system can fulfill all the decision activities in this phase.

In order to better understand the problem, the advanced PAV concept development is briefly explained. Then, the application begins with selecting the most appropriate decision making method for the concept selection problem, the selected method is then used to identify the most viable PAV concept. Furthermore, the method is improved as an illustration of a new method generation application. These outlined implementations help to demonstrate the practicality of the advisor system.

4.1 Proof of Concept

4.1.1 Personal Air Vehicle Concept Development

Great innovations in transportation systems have occurred dating back to the exploration of from exploring the first paths for commerce to the current air and interstate highway system. Electronic commerce, increasing populations and the information revolution brought about by the internet are placing new demands on today's transportation resources. The current transportation systems, represented by the centralized hub-and-spoke air transportation system and the ground highway systems, are challenged in this era in which time has become a scarce commodity. It is becoming increasingly important to find innovative concepts that can alleviate today's transportation problem. A Personal Air Vehicle concept, with door-to-destination, airborne, personal transportation capabilities, is part of a possible solution to the challenge.

It is a fact that since the first flight many efforts have been made to develop personal air vehicles, and most of the proposed concepts are flying cars which has the capability to complete both ground and air transportation logs. Figure 43 and Figure 81 list some tested PAVs developed by individuals from 1910s to present. In order to enhance the transportation system capability, NASA also made efforts to explore the concepts of Personal Air Vehicles to meet the future civil and possible military missions [NASA LaRC, 2002]. These revolutionary PAV concepts are basically developed from the state-of-art baselines with the infusion of the advanced technologies.

(a) Curtiss Autoplane

(b) Waterman Arrowbile

(c) Pitcairn Whirlwing

(d) Stout/Spratt Skycar IV

(e) Fulton Airphibian

(f) Taylor Aerocar

(g) ConVairCar

(h) Bryan Roadable

Figure 42: Roadable Personal Air Vehicles (1910s – 1970s) [Lewe, 2005]

(a) Moller Skycar M400

(b) MACRO Skyrider X2R

(c) See-through of Cityhawk

(d) Sky Technologies Aircar

(e) Sokol A400

(f) LaBiche FSC-1

(g) Groen's Hawk 4

(h) CarterCopter

Figure 43: Recent PAV Concepts [Lewe, 2005]

4.1.1.1 Problem Definition

The Personal Air Vehicle was envisioned as a potential replacement for automobile transportation which could provide a solution for the increasingly congested highways. For this personal transportation purpose, a PAV is intended to provide a significant improvement in mobility in as compared to the current transportation system and meet all regulatory requirements. As a personal vehicle, it must be easy to operate, safe as well as reliable. Also, travelers are always interested in reaching the destination in shorter time, so shortening the travel time is important. In addition, PAVs should be affordable to the consumer such that the vehicles can penetrate the market and achieve wide utilization. With increasing utilization, environment requirements such as noise and emission should be considered as well. Take-off and landing field lengths will be important considerations if air vehicle operations are to become more widely distributed within communities. Quantitative targets for all these requirements and criteria are currently emerging from ongoing system studies. These targets correspond to various segments of interest for the PAV market. Some selected requirements and criteria are shown in Table 7.

Table 7: PAV Criteria

REQUIREMENT	CRITERION	TARGET/ CONSTRAINT
Performance		
Speed	Cruise Speed (kts)	Ref. to Mission Profile
Noise	Flyover Noise (dB)	<79
Travel Time	Total Travel Time (hr)	<3.5
Takeoff Length	Total Distance to clear 50’ obstacle (ft)	Ref. to Mission Profile
Safety	Accident Rate : Number of fatal accidents per 1,000,000 FH	<5
Reliability	MTBF : Mean Time Between Failure (hr)	>80
Maintainability	MTTR : Mean Time To Repair (hr)	>50
Easy to Operation	TTR : Training Time Requirements (hr)	<20
Mobility	TTBT : Total Time Before Takeoff (hr)	<0.3
Economics		
Price	Acquisition Price ($)	Minimize
Cost	Direct Operating Cost ($)	Minimize

As mentioned previously, PAVs provide a routine doorstep-to-destination personal travel, which is a system solution involving air and ground transportation, generically depicted in the mission profile shown in Figure 44. This indicates that a PAV must complete the main mission from access portal A to access portal B, that is, from one airport location to another. PAV options have been categorized into 4 groups based on their takeoff and landing distance: Vertical Take-off and landing (VTOL), Extremely Short Take-Off and Landing (ESTOL), Short Take-Off and Landing (STOL), and Conventional Take-Off and Landing (CTOL) [DeLaurentis et al., 2002]. The constraints for cruise speed and takeoff length were defined for various options in the mission profile.

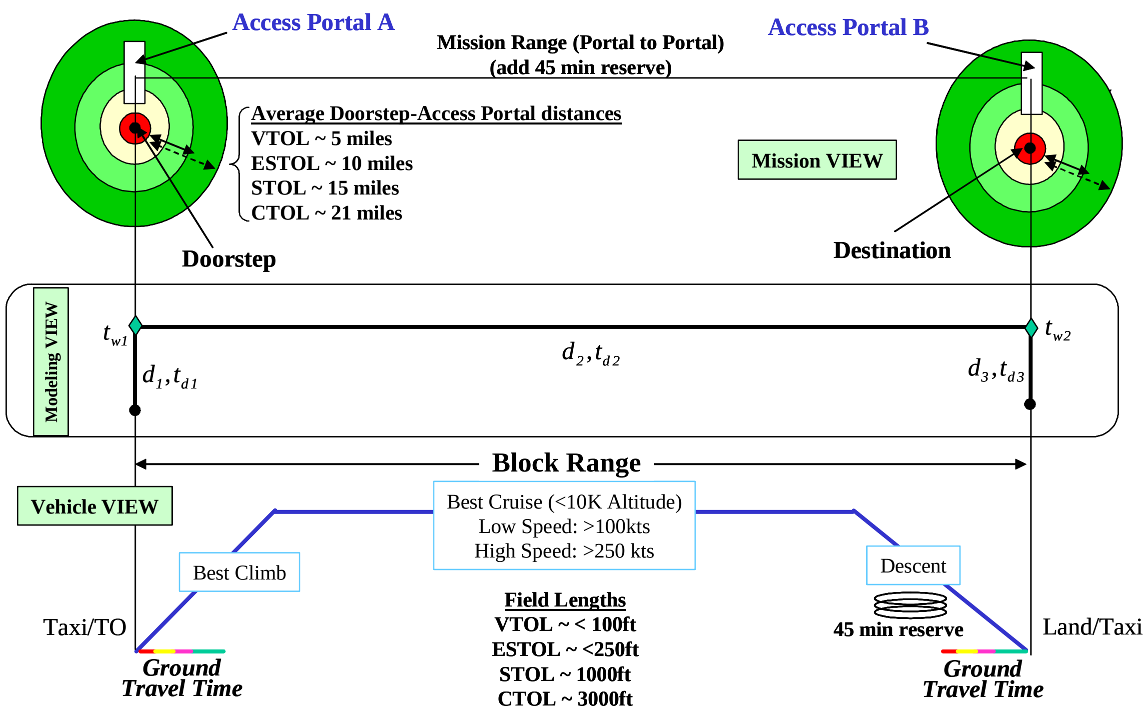


Figure 44: PAV Mission Profile

4.1.1.2 Baseline Concept Analysis

Baseline Concept Identification

For a personal use vehicle, mobility and safety are the most important requirements. Generally speaking, rotorcraft vehicles have advantages in these important areas. Currently, a VTOL vehicle, such as a rotorcraft vehicle, is the only concept that can provide doorstep-to-destination transportation in a single mode. It is the only air vehicle concept that can directly contribute to reducing ground transportation congestion. Furthermore, the ability to autorotate and land safely when engines fail provides more safety to the passengers compared with conventional CTOL aircraft. Thus, rotorcraft have the potential to be among the safest and easiest to operate vehicle concepts of all PAVs. Finally, by incorporating a foldable rotor system design, rotorcraft may have the potential to serve as a roadable, or “dual mode” vehicle. With the advantages mentioned above, the rotorcraft sector appears to be a worthy area for detailed study.

All design studies require a baseline, both to provide a departure point for design space investigation and to serve as a constant datum by which generated alternatives can be compared. In this study, focusing on V/STOL aircraft, one helicopter configuration, (Robinson R44), one gyroplane configuration, (Groen Hawk4), and one tiltrotor configuration, (Bell 609) were selected as baselines (Figure 45) [Robinson Helicopter Company, 2003; Bell Helicopter Textron, 2004; Groen Brothers Aviation, 2004]. Each of the three configurations has its own advantages to perform the PAV mission and represents the tried and tested technologies of today.

Figure 45: PAV Baseline Concepts

Modeling and Simulation

Two sizing and performance programs were applied to analyze the performance characteristics of the baseline for the three configurations. The Georgia Tech Preliminary Design Program (GTPDP), with the capability of providing a rapid assessment of the performance of a single main rotor (with or without a wing) with a single tail rotor or a coaxial configuration with conventional turbine engines [Schrage et al., 1986], was used for the R44 helicopter and Hawk4 gyroplane configuration. VASCOMP, the V/STOL Aircraft Sizing and Performance Computer Program developed by Boeing for NASA Ames where it was subsequently enhanced through the years, was applied to size the Bell 609 tiltrotor configuration [Schoen et al.]. After the baseline concepts were selected, the effort was concentrated on calibrating GTPDP by modeling the baseline aircraft. This calibration exercise emphasized not so much the overall fidelity of the tool, but concentrated on the sizing algorithms within the tool to match the given class of vehicles to be examined to actual data.

Design Space Exploration and Feasibility Evaluation

After the sizing environment was created, the design space was explored using Response Surface Methodology (RSM). The goal of the RSM is to generate the response surface equations, which capture the relationship between the analysis input variables and metrics of interest (objectives), and determine the system feasibility. The RSEs are constructed by executing the design of experiments, which are the combination of different values of the input variables. The parametric environment embodied by these equations is termed a United Tradeoff Environment (UTE).

The variables of interest were first identified as the input variables and listed in Table 8. The minimum and maximum values of each variable define the design space of interest and directly affect the metric values. The metrics are necessarily associated with outputs of the specific analysis codes, GTPDP or VASCOMP. These outputs are referred to as responses in the RSM terminology and related to PAV mission requirements, vehicle attributes, and are used individually or in combination to evaluate system feasibility and viability. The list of responses throughout this study is presented in Table 9.

Table 8: Input Variables for Construction of UTE

Variable	Description	Unit	R44		HAWK4		Bell 609	
			min	max	min	max	min	max
ALT	Altitude	ft	1300	1500	1300	1500	8000	12000
ROC	Rate of Climb	fps	15	18.4	22.5	27.5	30	36
LR	Labor Rate	$/hr	40	50	35	45	40	50
PL	Payload	lb	200	1600	200	1600	200	1600
RANGE	Mission range	nm	100	500	100	500	100	500
VC	Cruise speed	kts	90	140	100	150	250	350
VT	Rotor Tip Speed	fps	493	603	428	524	630	750
ARHT	Hori. tail AR	%	2.22	2.87	2.94	3.8	3.67	4.75
DL	Disk loading	psf	2.6	2.9	2.45	2.65	13.59	16.61
CFUEL	Cost of Fuel	$/lb	1.8	2.2	1.8	2.2	1.8	2.2
IRPY	Interest Rate/Year	%/yr	7.2	8.8	7.2	8.8	7.2	8.8
UTIL	Utilization	hrs/yr	260	1300	260	1300	260	1300

Table 9: Responses to Be Tracked in UTE

Variable	Unit	Description
GW	lb	Gross Weight
WEMPTY	lb	Empty Weight
WFUEL	lb	Fuel Weight
IP	shp	Installed Power
MTIME	hr	Total Mission Time
AC	$	Acquisition Cost
DOC	$/hr	Direct Operating Costs
NOISE	db	Noise

According to the Pareto principle, roughly 80% of the variability of a response is due to 20% of the variables. Hence, a screening test was performed to identify the variables which have main effect on the objectives. After performing the screening test, the main variables were found to be payload, range, cruise speed, disk loading and utilization. An automated design environment built around the specific sizing and performance codes (GTPDP and VASCOMP) was used to create a metamodel (RSEs) of the design space based on the range of input variables. By using the DoE technique, a number of experiments were generated, resulting in different combinations of values for the 5 input variables. After all the required runs of GTPDP/VASCOMP have been completed, the resulting data was used to regress relationships of the responses to the 5 inputs. These relationships take the form of 2nd order polynomial equations, (Response Surface Equations).

Once the RSE metamodel was created, the design space could be better visualized in the 2-D "design contour plot", which plots contours of the responses versus any two design variables, in the form of a dynamic tradeoff environment. Constraints can be set on these contours, to show the feasible design space. Because the design space is represented as a metamodel, contours can be quickly updated to reflect the effects of changing requirements. The feasible design space for the R44 is shown in Figure 46. As can be seen in Figure 46, there is no feasible design space, so new technologies need to be infused to meet the PAV requirements.

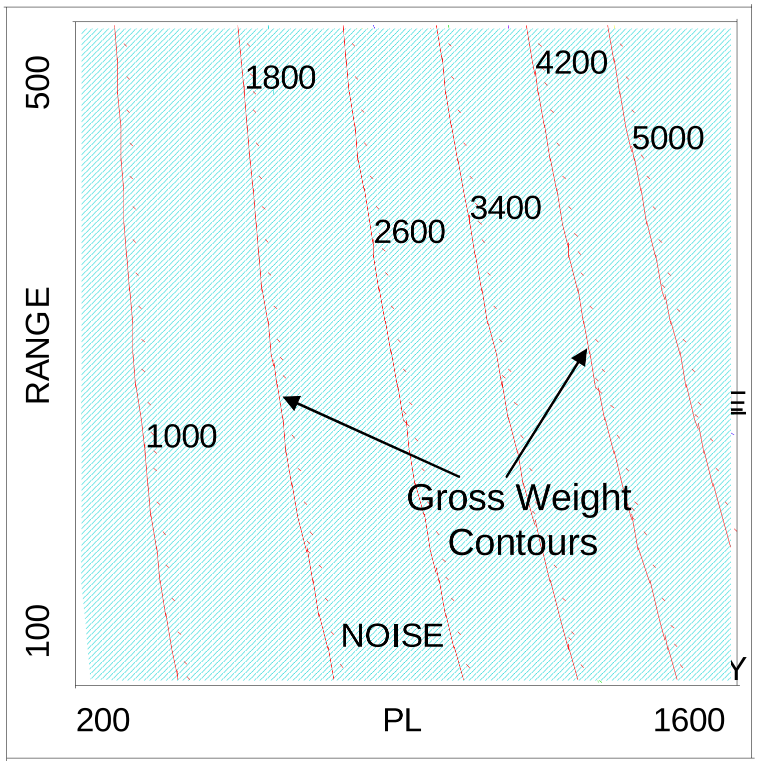


Figure 46: Feasible Design Space Exploration for R44

4.1.1.3 Advanced Technology Concepts

For the technology studies, the potential technologies that may improve technical feasibility and economic viability of the vehicles are identified first. For this study, the main technology areas examined for improvement are as follows: engine characteristics, component weight, direct operating cost, aerodynamic characteristics, power available and required, and noise characteristics.

The impact of a technology can be quantitatively assessed with technology metric "k" factors, which modify disciplinary technical metrics. A "k" factor is a multiplier on a given disciplinary metric that is used to simulate generic application of advanced technologies. Sets of "k" factors, representing the corresponding technologies, are applied to the state-of-the-art baselines, generating the advanced technology version of the PAV concepts. These factors can later on be mapped to actual technologies being applied. The simulation of advanced technologies in the form of "k" factors enables a dynamic mapping and visualization of the Technology Impact Forecasting (TIF) space. The variables of interest for the advanced technology concepts were identified and listed in Table 10.

Table 10: UTE Variable Definitions and Design Space

Var.	Description	Notation	Unit	R44		Hawk4		Bell 609	
				min	max	min	max	min	max
var1	Fuel Flow Ratio	K_FFR	~	0.8	1	0.8	1	0.8	1
var2	Weight Factors	K_WR	~	0.8	1	0.8	1	0.8	1
		K_WE	~	0.8	1	0.8	1	0.8	1
		K_WD	~	0.8	1	0.8	1	0.8	1
		K_WA	~	0.8	1	0.8	1	0.8	1
var3	DOC Factors	K_DOCE	~	0.8	1	0.8	1	1	1.2
		K_DOCD	~	0.8	1	0.8	1	1	1.2
var4	Airframe Drag Area	K_DRAG	~	0.8	1	0.8	1	0.8	1
var5	Noise Factor	K_NOISE	~	0.8	1	0.8	1	0.8	1
var6	Disk Loading	DL	psf	2.6	2.9	2.45	2.65	13.59	16.61
var7	Payload	PL	lb	200	1600	200	1600	200	1600
var8	Mission Range	RANGE	nm	100	500	100	500	100	500
var9	Cruise Speed	VC	kt	90	140	100	140	180	300
var10	Utilization	UTIL	hrs/yr	260	1300	260	1300	260	1300

The design space, with the effect of full benefit of the technology infusion, is illustrated in Figure 47 through use of the prediction profiler. From this point, these advanced technology versions of the R44, Hawk 4, and Bell 609 are called as the advanced helicopter, advanced gyroplane, advanced tiltrotor concepts respectively. As can be seen, a feasible design space emerges with the impact of adding technologies when the same constraints are applied. Compared with Figure 46, with the impact of technologies infusion, Figure 47 clearly presents that advanced technology concepts make a big improvement in performing the PAV mission.

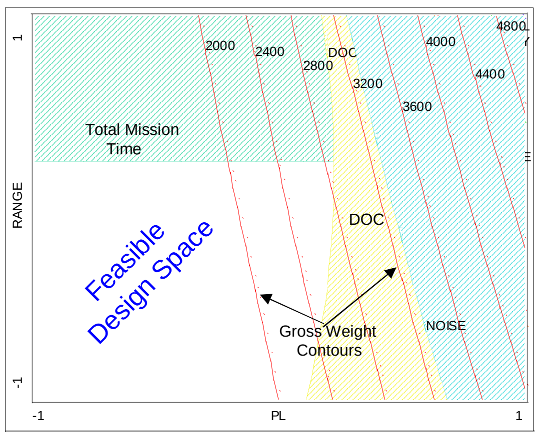


Figure 47: Feasible Space Emerges with Technologies Applied

4.1.2 PAV Concept Selection

The study in Section 4.1.1.3 has shown that all the advanced technology concepts developed from the baselines are feasible concepts capable of performing the PAV mission as shown in Figure 44. Determine the concept among the advanced helicopter, advanced gyroplane and advanced tiltrotor which can best satisfy the customer's requirement can be accomplished through the concept selection process. This is a pure decision making problem and is always a challenge to the engineer. In order to obtain a desired solution, an appropriate decision making method should be identified and then help the DM to reach the final solution. The multi-criteria decision making advisor has the capabilities to fulfill these tasks.

4.1.2.1 Decision Making Method Selection and Decision Validation

The MIDAS is an advisory system that can help the DMs to identify the most suitable decision technique to solve their problem. For the advisor to function properly and effectively, the necessary information associated with the problem must be provided so that the essence of the problem can be captured by the advisor. Misleading or incomplete information may result in the selection of an inappropriate method. This requires that the DMs should understand the problem under consideration, including what criteria were used to evaluate the alternatives are, what the preference information is and how it is represented. This information needs to be collected and organized so that a firm basis can be formed to essentially represent the problem. Analogically, this resembles the problem definition step in the engineering design process, which plays a critical role in determining a successful design. The study presented in this implementation is based on the assumption that the decision maker is able to fully understand the problem.

Since the development of the PAV advanced concept occurs in the early design stage, each concept carries a family of alternatives instead of a point design to avoid a rapid design freedom drop off and cost lock-in. The design alternatives of each concept are in the design space, with the infusion of advanced technology, defined by the combination of Table 8 and Table 10. The relationship between input variables and metrics of interest is captured by a metamodel referred to as response surface equations. Thus, the metrics of interest listed in Table 9 will be derived by utilizing the RSEs and distribute over the design space. That is, the quantification of each metric is represented by a distribution rather than a single value which exhibits the uncertainty nature of the problem. Figure 48 illustrates the noise distribution of advanced helicopter in the form of Probability Distribution Function (PDF) as an example of metric distribution. The nominal metrics of interest will act as the criteria used to evaluate PAV advanced technology concept, thus in order to solve this concept selection problem, a selected method must have the capability to manipulate the uncertain metrics. This uncertainty feature is a key characteristic that needs to be taken into account when one selects the decision making methods for the PAV advanced technology concept selection problem.

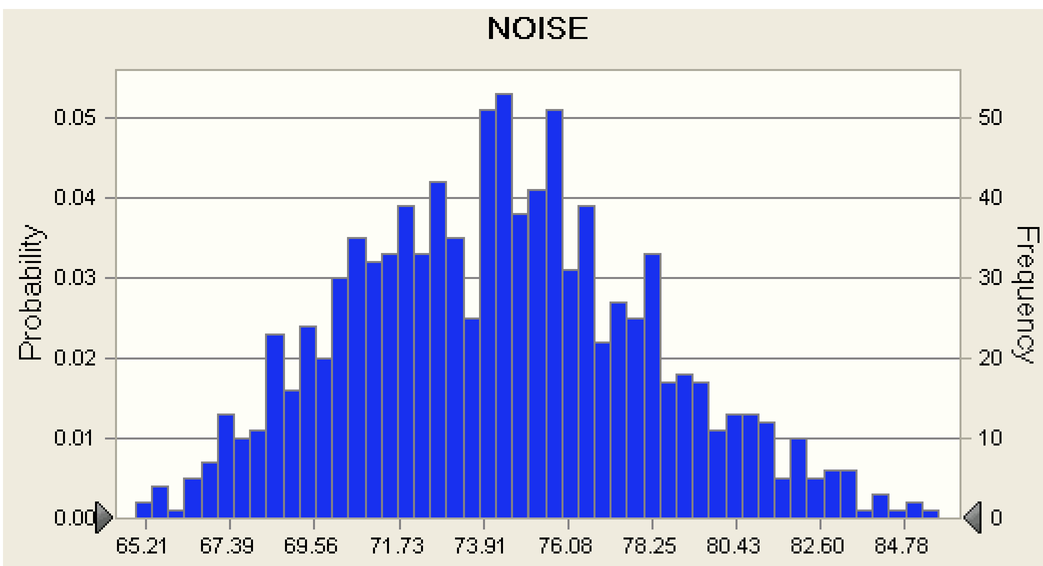


Figure 48: Noise Distribution of Advanced Helicopter Concept

As demonstrated in Section 0 and 4.1.1.3, the advanced technology concepts were developed by employing the TIES method, which encompasses a feasibility examination process. Therefore, all the three PAV advanced technology concepts are feasible and no feasibility evaluation is required to fulfill the concept selection problem.

The concepts are compared based on the evaluation of the given criteria, and the one that best satisfies the criteria is suggested as the PAV concept. This decision making scheme indicates that the final solution depends highly on what criteria are used for concept selection. In addition, the preferences of the criteria also have strong impact on the final solution since one design concept often is better at some aspects but worse at others than another concept. The preference information of this problem is represented by the relative weight, thus each criterion is given a weight showing its importance when the concepts are evaluated. The weighting factor can be directly assigned by the DM or can be obtained by performing the QFD analysis [Dieter, 2000].

The PAV concepts are envisioned to perform a door step-to-destination mission depicted in Figure 44. In order to be a successful concept, the customers' requirements should be met, as shown in Table 7. The requirements are often in the form of constraint and serve as the criteria base on which the concept is selected. For example, the total travel time should be less than 3.5 hours and the flyover noise should less than 79 dB. One design solution will be infeasible if it violate any of the criterion constraint.

The available information for this concept selection problem is a set of RSEs used to facilitate the parametric assessment while providing a simple, easily manipulated approximation of complex model [Kleijnen, 1987]. These second order polynomial equations enable the quick tradeoff studies attempting to maximize the probability of success of the design solution.

Since the value of the evaluation criterion is expressed as a probability distribution, the decision rule used to determine the best concept should be a function of the PDF of the criterion to capture the uncertainty effects and to provide suitable confidence on the results obtained. Probability of Success (POS) is a plausible objective that can be employed as a criterion to evaluate the concept, and thus the decision rule will become the maximization of the POS of the concept.

The characteristics of the PAV concept selection problem outlined above are summarized in Table 11. The complete understanding of the decision making problem is the foundation that supports the sequent design decisions. From this point, the multi-criteria decision making advisor will be employed to facilitate the decision making process.

Table 11: Characteristics of the PAV Concept Selection Problem

Problem Characteristics	PAV Concept Selection Problem
Problem Type	Concept Selection
Alternative Characteristics	Existing, and Feasible
Attribute Characteristics	Constrained
Preference Representation	Relative Weight
Preference Information	Given/Assigned
Key Characteristics	Uncertainty
Available Information	Response Surface Equations
Decision Rule	Maximize the Probability of Success

The most appropriate method for selecting the PAV concept selection problem is needed first. The user sends the query to the advisor system to request the method selection task. Based on the task, the advisor will present the user a set of questions which are related to the characteristics of the problem. The user is allowed to choose the answer to each question from the options provided by the advisor (Figure 49). For the

PAV concept selection problem, the answers were selected depending upon the problem characteristics listed in Table 11.

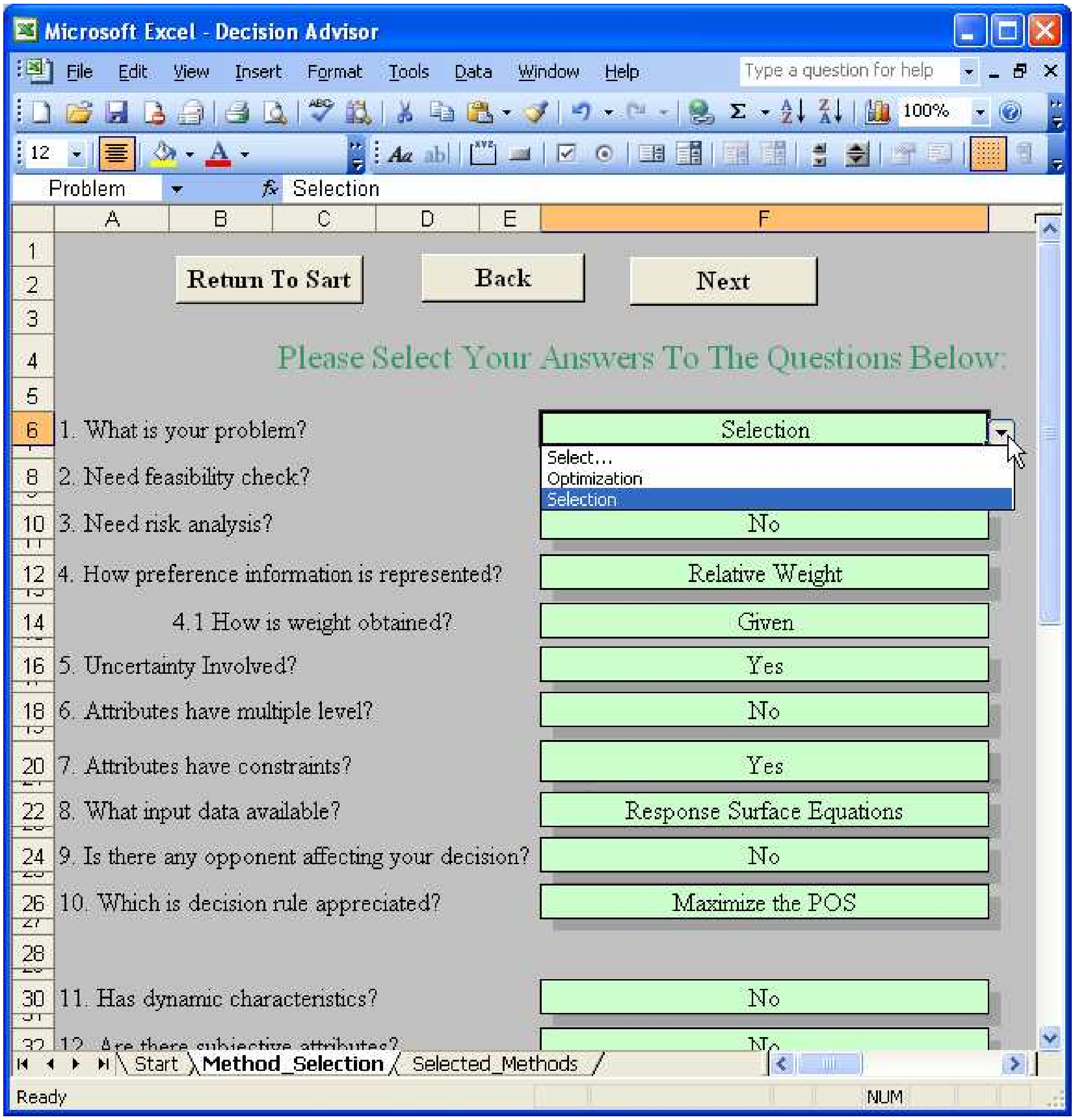


Figure 49: Questions Provided by MIDAS Used for Selecting Decision Method

The advisor then analyzes the answers using the information in the knowledge base and sequentially calculates the appropriateness score for each method in the method base. In this example, the questions are assigned the same weight except that the question 5 is assigned a higher importance because the uncertainty is a key characteristic of the PAV concept selection problem. The final result of the method selection is illustrated in Figure 50. It can be seen that the Joint Probability Decision Making Technique (JPDM) is evaluated as the best method to handle the problem under consideration.

The obtained result can be explained by comparing the methods with the given problem. As described in Section 2.6.2.5, JPDM is a technique which is capable of dealing with the product selection and optimization problems this indicate that JPDM can handle both MADM and MODM problem. This technique incorporates a multi-criteria and a probabilistic approach to system design, thus it can capture the uncertainty existing in the problem. In addition, JPDM uses joint probability of success, which assesses the probability of satisfying the criteria concurrently, as the objective function to make design decision. The joint probability of success is calculated over the area of interest which is defined by the constrained criteria. The joint probability of success can also be used to evaluate the feasibility of the alternative since if an alternative is not feasible, its joint POS will be zero. The preference information of the JPDM technique is represented by a weighting factor and each criterion is assigned a default weight of $1/N$ which can be adjusted if the criteria have unequal importance, where N is the number of the criteria. Furthermore, JPDM has five implementation schemes, as shown in Figure 51, and each of them employs different techniques to generate the input data, such as metamodel or Fast Probability Integration (FPI). In summary, the characteristics of JPDM technique are summarized in Table 12.

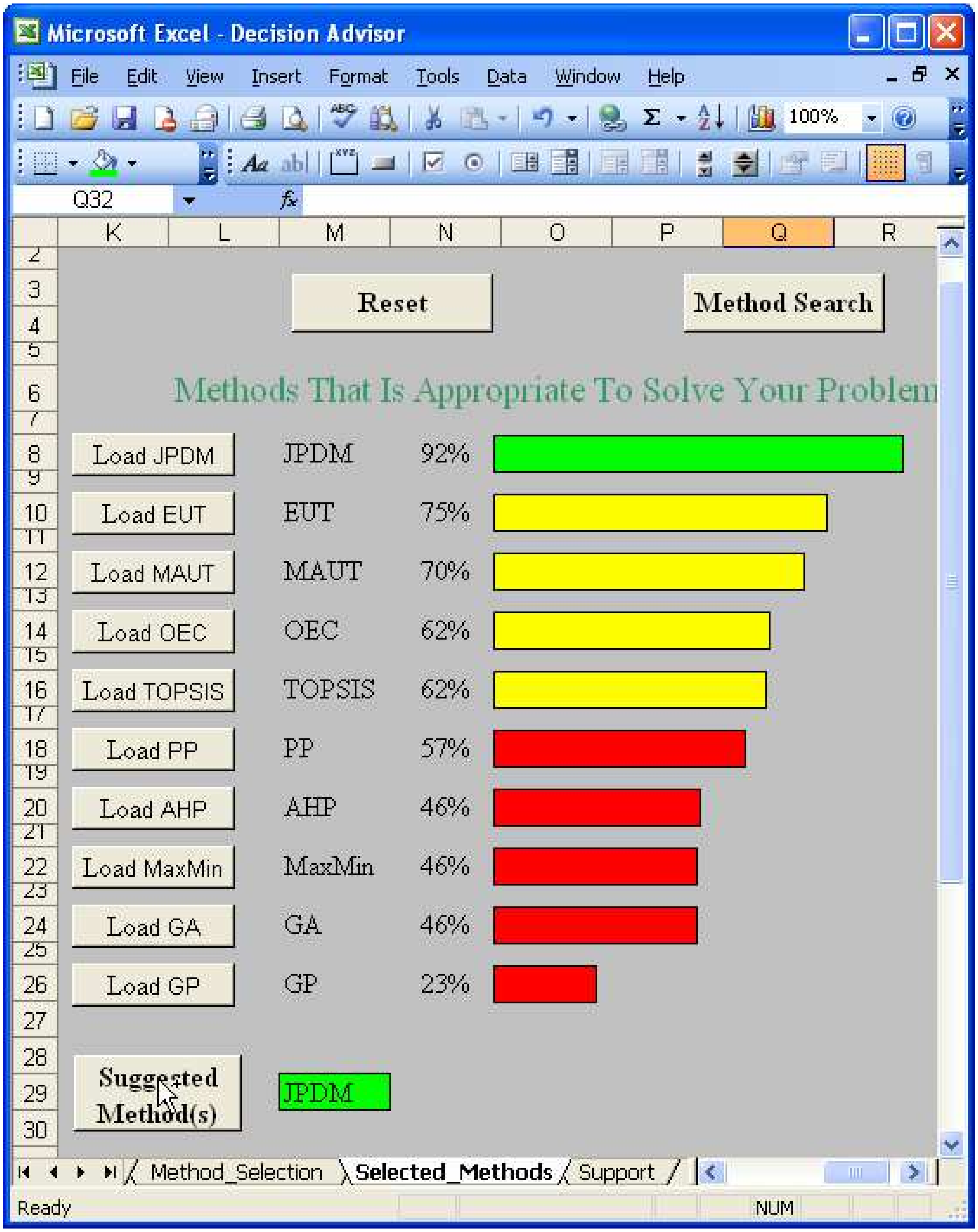


Figure 50: Method Selection Results for PAV Concept Selection Problem

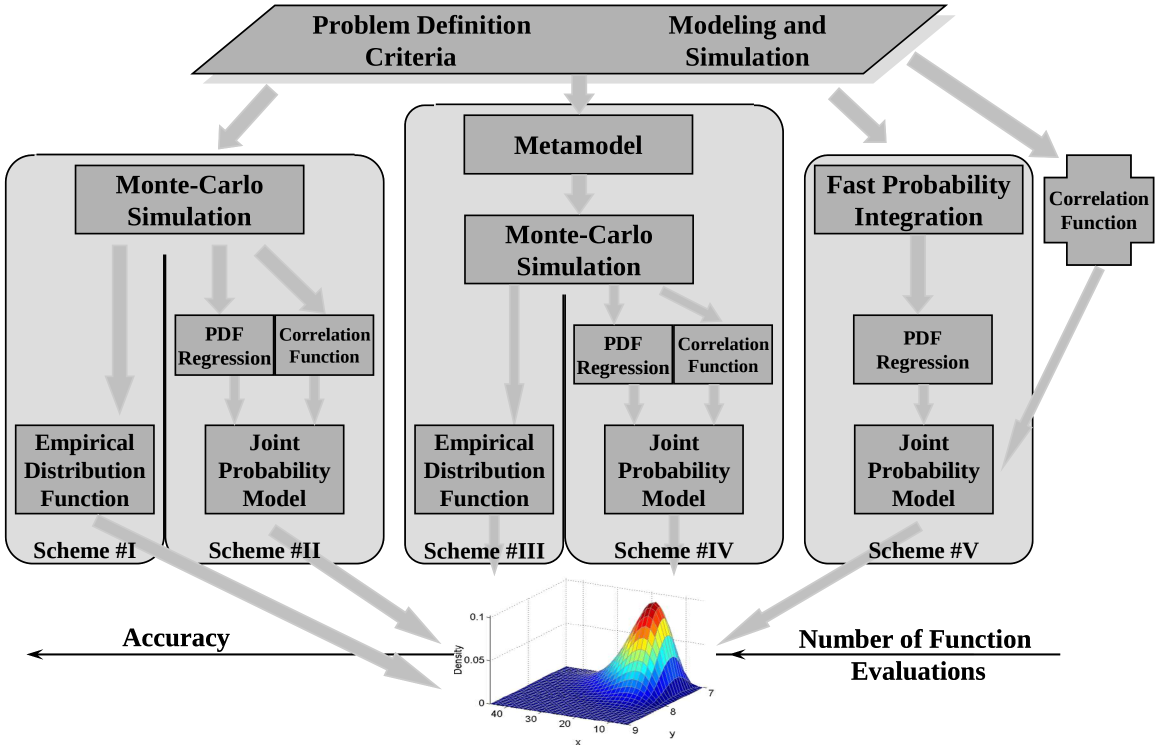


Figure 51: Schemes for Evaluation of the Joint Probability Distribution [Bandte, 2000]

Table 12: Characteristics of JPDM Technique

Method-Related Characteristics	JPDM Technique
Problems Handled	MADM
	MODM
Can perform feasibility evaluation?	Yes
Attribute Characteristics	Constrained attributes define the area of interest
Preference Representation	Weighting Vector
How is preference obtained?	Assigned
Key Capabilities	Capture Uncertainty
	Handle Multiple Criteria
Inputs Accepted	PDF of criteria (JPF model)
	Empirical Distribution Function of criteria (EDF model)
Objective Function	Probability of Success
Decision Rule	Maximize the Probability of Success

Comparing Table 11 and Table 12, one can see that the characteristics of JPDM technique match well with the characteristics of the PAV concept selection problem. This indicates that the JPDM technique possesses the capabilities required to solve the PAV concept selection problem. This result is consistent with the selection the MIDAS made for the problem.

The method with the second highest appropriateness score is the Expected Utility Theory (EUT) technique. This is due to the fact that the EUT technique is able to utilize the probability distribution function to capture the uncertainty of the problem. This capability helps it to obtain a higher score than the rest of the methods. However, the EUT technique requires a utility function of the criteria to complete the assessment while the PAV concept selection problem does not provide this input information. In addition, EUT does not consider the condition that the criteria have constraints, which will lead to an infeasible alternative being selected as the best solution. These observations explain why EUT got a score less than the threshold and imply that it is not an appropriate method to be capable of handling the current problem.

The low scores obtained by the rest of the methods indicate that they are far from being an appropriate method to solve the PAV concept selection problem since they are not able to manipulate the uncertainty existing in the problem and most of them don't perform feasibility evaluation. Moreover, none of the method can utilize the available information, RSEs, to evaluate their own objective function.

Therefore, the JPDM technique appears to be the most appropriate method to handle the PAV concept selection problem, and the utilization of this method is expected to produce a desired solution for the given problem.

The decision validation process is similar to the method selection process, except that the decision was already made using another method. To validate the decision made, the method selection process should be worked through. If the selected method is the same as the method used, the decision should be valid, otherwise the decision need to be remade using the selected method suggested by the MIDAS.

4.1.2.2 PAV Concept Selection Using JPDM

The JPDM technique is a powerful method that is able to assess the probability of satisfying the multiple criteria concurrently while keeping the infeasible alternatives from being selected. As described in Section 2.6.2.5, this technique can uses an EDF for a joint probabilistic formulation to calculate the joint and marginal probability of success. The EDF model for calculating the joint probability of success is given in Algorithm 1.

Though the JPDM is a good method for making decision under uncertainty, the decision makers may not know to use this method to solve the problem under uncertainty. It is a way that the DMs can learn the method by themselves and then apply it to the problem to get the problem solved. However, this is time consuming especially when the learning curve is steep.

The MIDAS can help the DMs to use the method that they are not familiar with to facilitate the decision making process. When a technique is selected as the most appropriate method to solve the current problem, the advisor can invoke the method which has a rigorous step by step problem solving procedure. This procedure is presented to the DMs through a user interface by providing an explicit guidance. By following the guidance, the DMs are allowed to use the method without know how it works. The only actions expected from the DMs are inputting some necessary data such as the number of the alternatives, the number of criteria. The rest of the assessments can be completed by clicking the corresponding buttons under the guidance.

Algorithm 1: EDF Model of JPDM Technique

Inputs: Number of alternatives n; number of criteria m; data sample of each alternative; number of data sample of each alternative N_i, $i=1,2,\cdots,n$; area of interest (constraints of each criterion) X_{ij}^{u}, X_{ij}^{l}, $i=1,2,\cdots,n$, $j=1,2,\cdots,m$.

Outputs: Joint probability of success for each alternative, Joint POS; univariate (marginal) probability of success of each criterion for each alternative POS_{ij}, joint probability distributions (plots)

Calculate univariate POS of each criterion for each alternative

for $i=1$ to n **do**
  for $j=1$ to m **do**
    for $l=1$ to N_i **do**
      if $X_{ij}^{l} < X_{ij} < X_{ij}^{u}$ **then**
        $I_{ijl}=1$
      else $I_{ijl}=0$
      end if
      $POS_{ij}=\frac{1}{N_i}\sum_{l=1}^{N_i} I_{ijl}$
    end for
  end for
end for

Calculate joint POS for each alternative

for $i=1$ to n **do**
  $Joint\ POS=\frac{1}{N_i}\sum_{l=1}^{N_i}\prod_{j=1}^{m} I_{ijl}$
end for

In this example, the JPDM technique was selected as the most appropriate method. To use the JPDM technique, one can simply click the "Load JPDM" button shown in Figure 50, and thus the JPDM technique will be loaded. Figure 52 illustrates the problem solving procedure of JPDM technique provided by the MIDAS. It can be seen that the advisor supplies an explicit instruction that can be followed by the DMs. The only

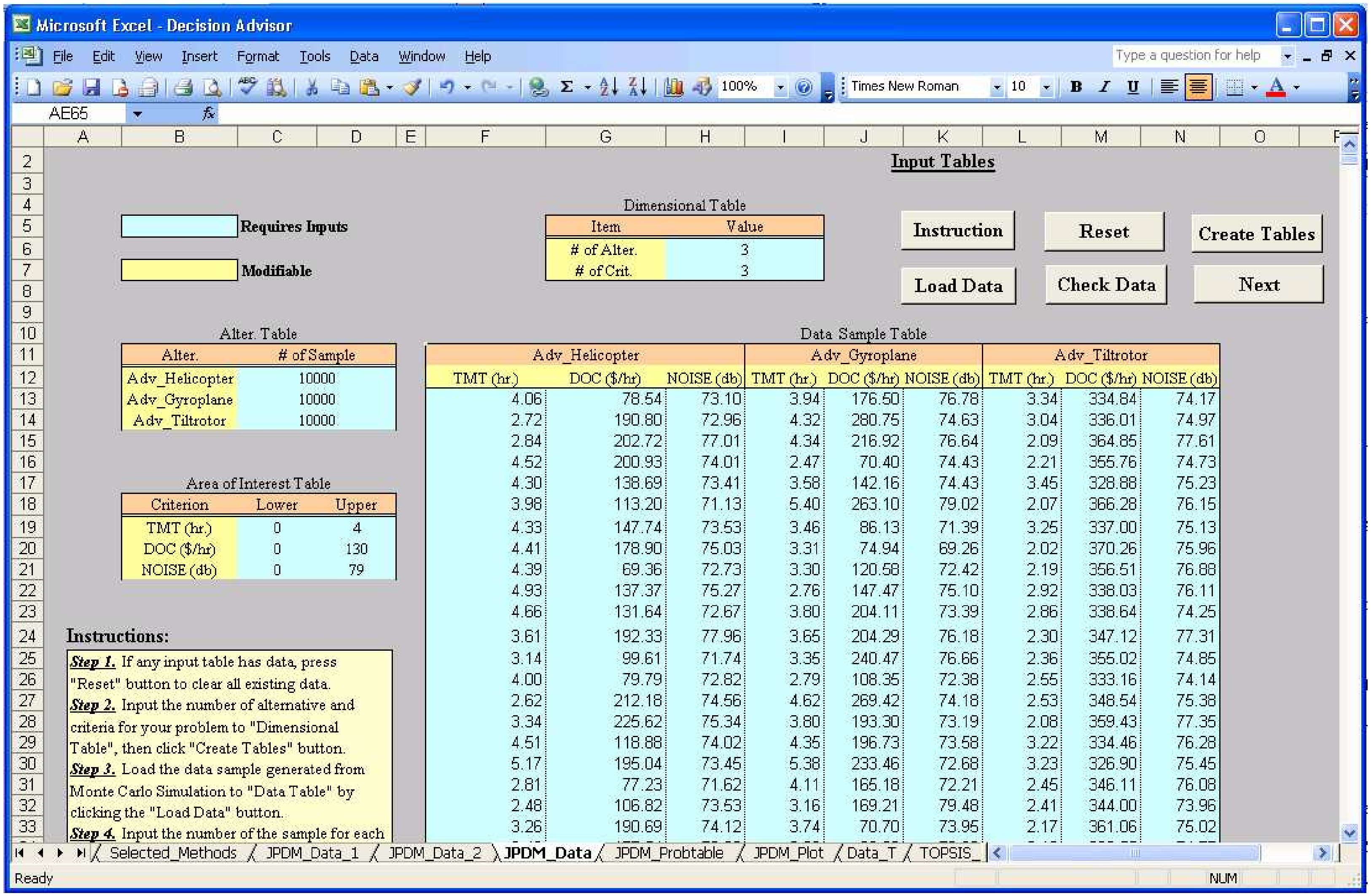


Figure 52: Problem Solving Procedure of JPDM Technique

information needs to be inputted by the DMs is the data highlighted in blue. The data in the data sample table can be obtained by some sampling technique such as the Monte Carlo Simulation using the available response surface equations. Uncertainty is propagated to the system level by defining appropriate probability distributions to uncertain mission requirements, vehicle attributes and infused technologies. The area of interest, defined by the upper limits and lower limits of the criteria need to be determined by the DMs. This area is the region that the desired solutions should be located in and outside of which any solution is excluded from analysis. In this study, three criteria were determined as the selection criteria: Door to Destination (D-D) time, Direct Operating Cost (DOC) and noise. One can clearly see that all the criteria are desired to be as small as possible, therefore zero as a lower bound was assigned to all the criteria. On the other hand, the maximum acceptable values are 4 hrs for D-D time, and 130 $/hr and 79dB for DOC and noise respectively.

Based on the data input, the advisor can automatically produce the joint probability distribution of the criteria. Figure 53 and Figure 54 show the joint probability distributions of D-D time vs. noise and DOC vs. noise. In addition the advisor can calculate the joint probability of success for each concept and univariate probability of success for each criterion. The respective probabilities of success are listed in Table 13. The steps to produce the joint probability distribution and calculate the probabilities of success can be simply completed by following the guidance provided by the advisor through a friendly user inter face. This simple operation allows the DMs to make their through a friendly user inter face. This simple operation allows the DMs to make their decisions using JPDM without knowing how the technique works.

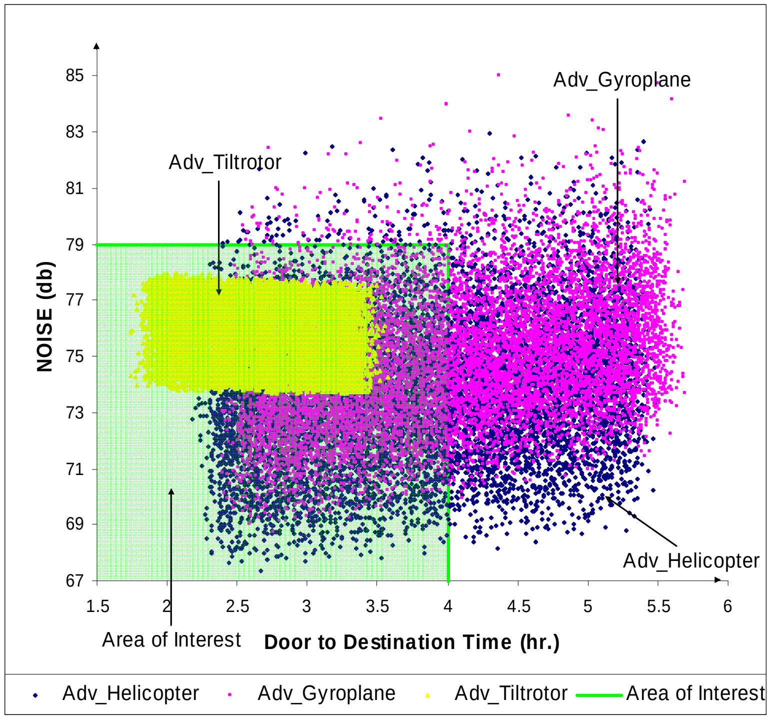


Figure 53: Joint Probability Distribution (D-D vs. Noise)

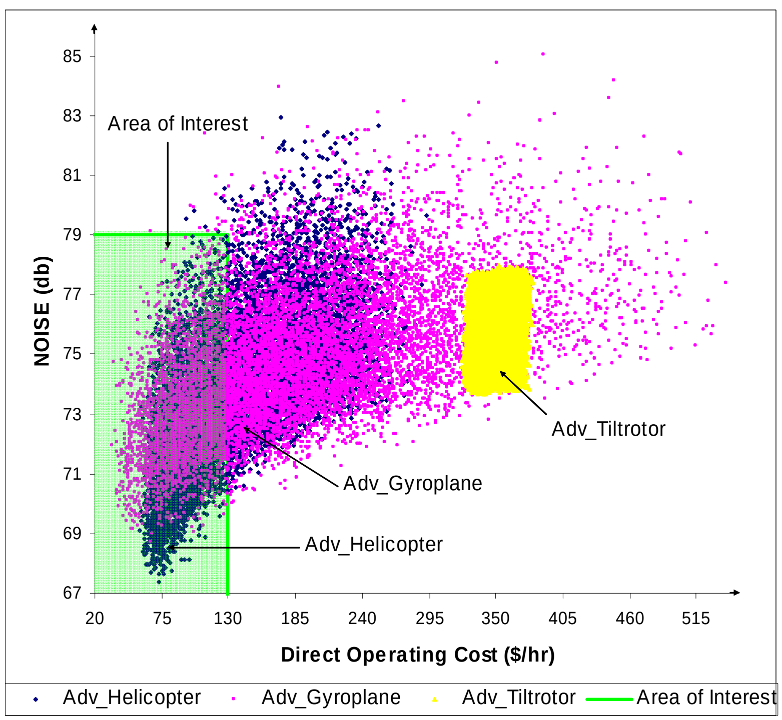


Figure 54: Joint Probability Distribution (DOC vs. Noise)

Table 13: Probability of Success

Alternatives	Joint POS	P(*D-D*<*4* hr)	P(*DOC* <130 $/hr)	P(*Noise* < 79 db)
Adv_Helicopter	0.2708	0.5572	0.4736	0.9759
Adv_Gyroplane	0.2004	0.481	0.2855	0.955
Adv_Tiltrotor	0	1	0	1

From Table 13, it can be seen that the highest POS was obtained with the advanced helicopter concept, indicating that this concept has more viability than the other alternatives as measured by the criteria of DOC, doorstep-to-destination time and noise. The advanced gyroplane has relatively high probability of meeting the requirements, while the advanced tiltrotor has zero probability of satisfying the criteria.

The same rank can also be obtained from the physical explanation. With similar cruise speed as the gyroplane, the advanced helicopter has VTOL capability, providing more access time savings compared to the advanced gyroplane concept with its ESTOL capability. This makes the advanced helicopter win over the advanced gyroplane when they are evaluated by the D-D time. In addition, compared with the advanced gyroplane concept, the advanced helicopter concept has an advantage for the DOC requirement. This is driven by the fact that the helicopter requires maintenance only on a rotor system while a rotor and wing system must be supported on the gyroplane. The noise levels of these two concepts are very similar. However, it is worth noting that the gyroplane concept is the safest concept among the three concepts because it is in autorotation at all times. The advanced tiltrotor predominated in the D-D time and noise because of its combination of the vertical take-off and landing with the speed and range of a turboprop However, this concept have no probability of satisfying the given criteria. The high DOC, due to its complexity, violates the constraint and causes it to have no chance to be a PAV concept. This can be clearly seen from Table 13 in which the PoS for satisfying DOC is zero.

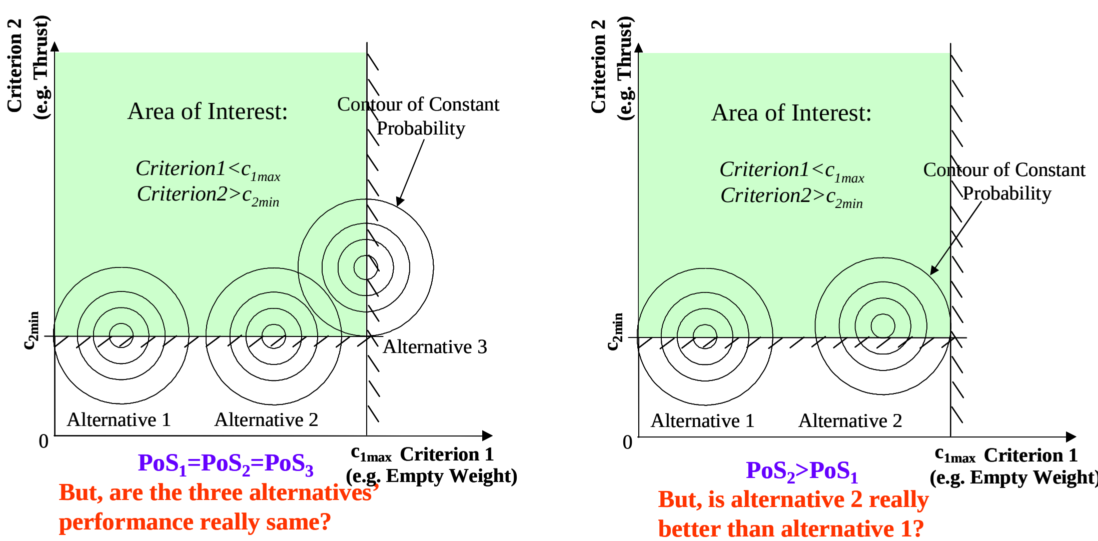
The result of the PAV concept selection problem demonstrated that the MIDAS is able to facilitate the decision making process by providing an explicit problem solving procedure of the selected method. This procedure allows the DMs to utilize the selected method through a friendly user interface without knowing the mathematical model of the method.

4.1.2.3 New Method Generation

The JPDM technique appears to be an effective multi-criteria decision making method which can measure the goodness of the alternative by producing the probability of the alternative satisfying the given requirements which is in the form of probability of sucess. The POS is a single metric that enables a comparison of all alternative solutions on an equal basis. Hence, POS allows for the use of any standard single-objective optimization technique available and simplifies a complex multi-criteria selection problem into a simple ordering problem, where the solution with the highest PoS is the best.

The advantages of the JPDM due to the use of POS does not cover its own underlying limitations. In the JPDM, the POS is obtained by integrating the joint probability density function over the area of criterion values that are of interest to the customer for the JPM model, or by counting the number of the occurrences of the alternative solutions within the area of interest for the EDF model. Obviously, the calculation of PoS does not take the absolute location of the Joint Probability Distribution Function (JPDF) into account, which leads the JPDM to become awkward for concept selection when the calculated

POS' of the alternatives are very similar but their JPDF locations are very different, as illustrated in Figure 55. In Figure 55 (a), assuming the criteria C_1 and C_2 will be minimized and maximized respectively, the POS of those three alternatives are totally equal (0.5), but one cannot say that the three alternatives have the same goodness. Alternative 1 is apparently better than the other two because it has less deviation from the target values (0 for criterion 1 and infinity for criterion 2). In the case shown in Figure 55 b), the POS of alternative 2 is greater than alternative 1, however, it is not prudent to say that alternative 2 is more advanced than the other. On the other hand, alternative 1 is much better than alternative 2 with respect to the given criteria due to the same reason given in the previous case. The weakness illustrated here indicates that the 'best' solution selected based on the value of the PoS by JPDM is not necessarily the actual best solution, which makes the JPDM become awkward for handling these kinds of concept selection problems. Furthermore, one can clearly see that for the general case the value of the POS cannot accurately represent the concept performance since it does not take the deviation into account.



(a) (b)

Figure 55: Limitations of the JPDM Technique

The probability of success is calculated using a joint probability function or empirical distribute function over a weighted area of interest. This discovers the fact that JPDM employs the relative weight to represent the decision maker's preference information. It is well known that the most serious drawback of the weighting method is that it cannot generate proper members of the Pareto-optimal front when this front is not convex. That is, there may not exist a weight vector that will yield a given Pareto point. The weighting method also suffers from the high computational costs as the number of optimization runs increases exponentially with the number of objectives.

In addition to the drawbacks described above, the JPDM utilizes the weight adjusted target values to adjust the weight. This technique narrows the target range of interest for the criteria with high preference weights and widens its range of interest for the ones with low weights [Bandte, 2000]. This concept is mathematically expressed as Equation (33) and (34) However, this treatment may upgrade an infeasible design to a feasible design when widening the area of interest and, similarly, downgrade a feasible design to an infeasible one when narrowing the area.

$$t_{\min} = (w \cdot N) \cdot z_{\min} \tag{33}$$

$$t_{\max} = \frac{z_{\max}}{(w \cdot N)} \tag{34}$$

where N is the number of criteria, $t_{\min}$ and $t_{\max}$ are the new lower and upper limits of the criteria defining the adjusted area of interest, $z_{\min}$ and $z_{\max}$ are the constraints on the criteria and w is the weighting vector representing the customer's preference.

The outlined observations indicate that the JPDM technique only considers where the weighted boundaries of the area of interest are and treats all the solutions in the area of interest the same. It is clear that the POS calculation can not fully capture the performance of the alternatives, thus, produces biased estimation of goodness. Therefore, the JPDM technique needs to be improved in order to be able to make high quality design decisions.

The improvement can be completed by revising an existing method or developing a brand new method, as a result, producing a hybrid method or a new method capable of fulfilling the capabilities which are required to make better decisions. The MIDAS is able to help the DM to generate the methods with improved performance in the process of selecting the most appropriate method for the problem under consideration. In other words, the MIDAS can provide hints to generate a new method to handle the given problem.

Assume that a decision maker wants to find a desirable method to solve the PAV advanced concept selection problem. The DM concerns about his or her preference and wishes that the preference information can be represented by a more sophisticated model rather than the relative weight. It is also assumed that the DM understands the other characteristics of the problem as listed in Table 14.

After the characteristics of the PAV concept selection problem are entered to the advisor system, the advisor will analyze the inputs and present the result to the DM through the user interface. Figure 56 depicts the method selection results. As one can see, there is no appropriate method which is capable of dealing with this problem. In this case, the advisor is able to provide hints that may be used to create a hybrid or new method.

Three hints are provided by the advisor, and they are: combining the JPDM with EUT, physical programming, and loss function. The JPDM technique still gets the highest score though it does not exceed the threshold. The JPDM is followed by EUT because EUT represents the DM's preference information using a utility function which is a good model for preference representation. In addition, the physical programming utilizes a class function to physically define the customer's preference, which is proven to be a successful model. Furthermore, loss function also provides a mathematical way to calculate the DM's preference. Therefore, the hints provided by the advisor are appropriate and can be used to develop the new method.

Table 14: PAV Problem Characteristics with revised Preference Information

Problem Characteristics	PAV Concept Selection Problem
Problem Type	Concept Selection
Alternative Characteristics	Existing, and Feasible
Attribute Characteristics	Constrained
Preference Representation	Sophisticated Model
Preference Information	Calculated
Key Characteristics	Uncertainty
Available Information	Response Surface Equations
Decision Rule	Maximize the Probability of Success

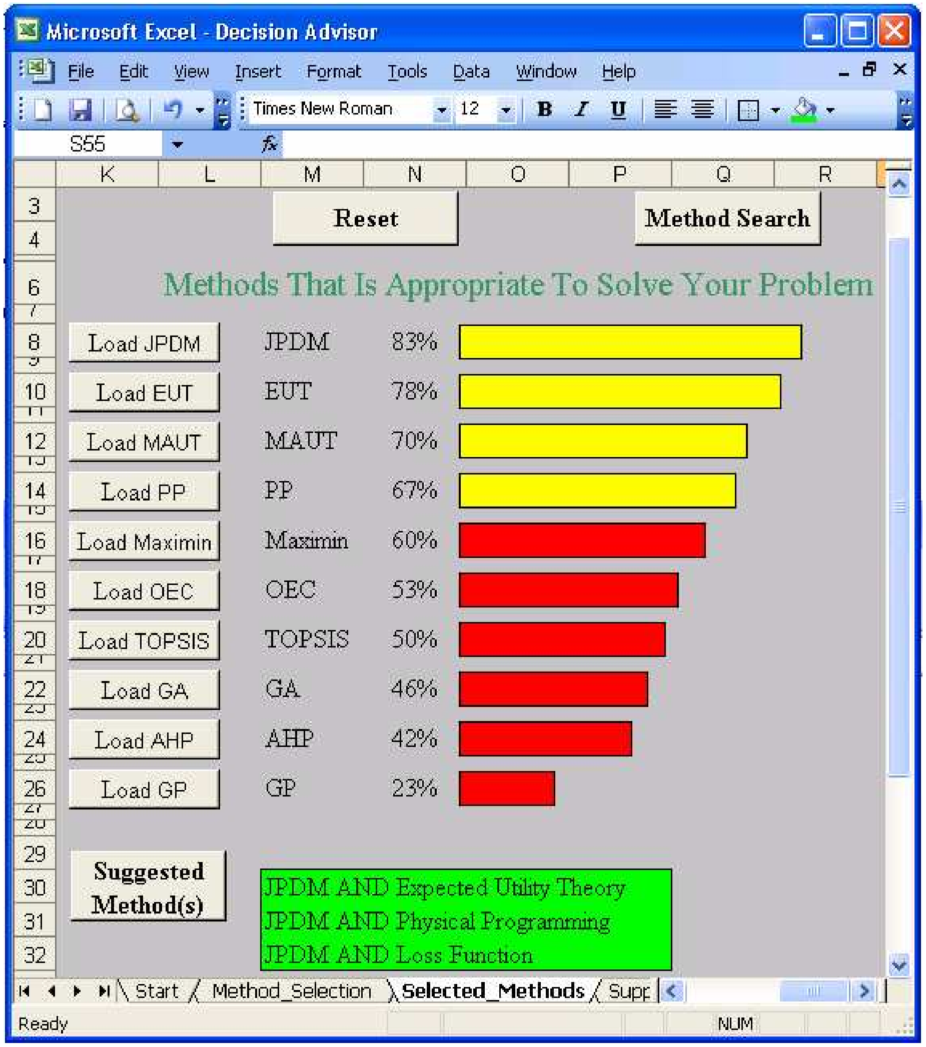


Figure 56: Hints Provided by the MIDAS for Generating New Method

Since utility has the capability of representing a decision maker's preference information by measuring the "goodness" of the decision making criteria, the first hint provided by the advisor is selected for the new method generation. As the JPDM technique still has highest appropriateness score, the new method will be developed based on this technique. The utility function used by EUT technique can improve the calculation of the POS of JPDM technique, thus it is used in the JPDM to represent the preference information.

Three types of utility functions are constructed and assigned to the corresponding attributes depending on their characteristics – smaller is better, larger is better or nominal is best. The utility of an attribute depends on its variation from target value which is the desired value that the attribute is expected to be. That is, utility is decreasing with the variation from target value. The attribute with “smaller is better” properties is an attribute that decision makers want to minimize, such as Direct Operating Cost (DOC). On the contrary, the “larger is better” attribute, such as Net Present Value (NPV), is to be maximized by decision makers. Attribute with “nominal is best” characteristics is an attribute that has highest utility at one specific value. These three utility functions are assumed in quadratic forms, given by Equation (35), (36) and (37) and visualized in Figure 57.

Larger is better:

$$u(x) = \begin{cases} 0 & (x \le x_l) \\ ax^2 + bx + c & (x_l \le x \le x_u) \\ k & (x_u \le x) \end{cases} \tag{35}$$

where: $a = k/(x_u - x_l)^2, b = 2kx_u/(x_u - x_l)^2,\ c = -kx_{l(}(2x_u - x_l)/(x_u - x_l)^2, 0 \le k \le 1$

Smaller is better:

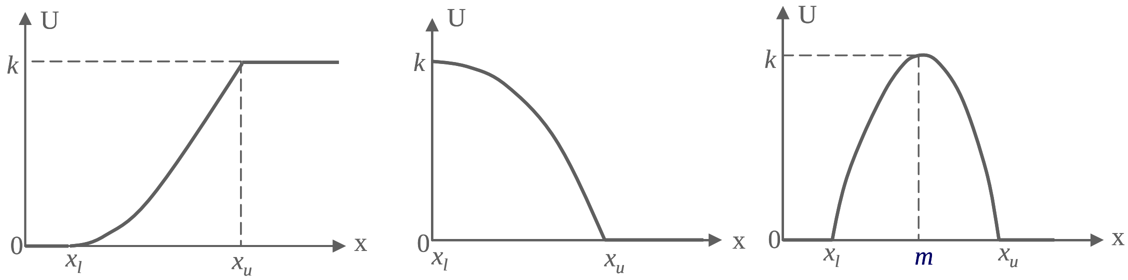
$$u(x) = \begin{cases} -ax^2 + k & (0 \le x \le x_u) \\ 0 & (\text{else}) \end{cases} \tag{36}$$

where $a = k/x_u^2, 0 \le k \le 1$

Nominal is best:

$$u(x) = \begin{cases} ax^2 + bx + c & (x_l \le x \le x_u) \\ 0 & (else) \end{cases} \tag{37}$$

where $a = -4k/(x_u - x_l)^2, b = 4k(x_u + x_l)/(x_u - x_l)^2,\ c = -4kx_u x_l/(x_u - x_l)^2, 0 \le k \le 1$



(a) Larger is Better (b) Smaller is Better (c) Nominal is the Best

Figure 57: Three Types of Utility Functions in Quadratic Form

This representation of preference information refines and improves the JPDM technique. In this study, Joint Utility (JU), with physical meaning, is assigned as the objective function. The joint utility is defined as an addition of marginal utilities contributed by all the attributes, given in Equation (38). This calculation is based on an assumption that the attributes are independent and their utilities are additive, and the design alternative is feasible.

$$U(X) = \sum_{i=1}^{n} u_i(x_i) \tag{38}$$

where $u_i(x_i)$ is marginal utility function

In a multiple decision making problem, the JU can be computed by integrating the Joint Utility Function (JUF) over the design space for JPM model, or dividing the total utility by the number of the solution in the concept sample for EDF model.

$$JUF = \iint_{\Omega} \cdots \int U(X) f(X) dX \tag{39}$$

$$JUF = \frac{1}{N} \sum_{i=1}^{n} U(X_i) \tag{40}$$

It is worth noting that there are two underlying assumptions here. One assumption is that the utility contributed by the other objectives in which the decision maker is not interested are not taken into account when calculating the JU, that is, the calculated JU is only valid under the given criteria. The other assumption is that the individual utilities can be added linearly when the joint utility is calculated. In other words, no correlation exists among the marginal utilities in the joint utility function.

The appropriate type of marginal utility function for each single criterion needs to be constructed using Equation (35 – 37) before the Joint Utility of an object can be computed. One can clearly see that given the quadratic form and interest of the area (x_l and x_u), the marginal utility function can be determined when the parameter *k* is known. The constant *k*, which defines the maximum utility of an attribute when it reaches the target value, may be the most difficult and important part of construction of the marginal utility function. In a single criterion decision making problem, most applications of the utility function can use a value of 1 for *k* since an objective is considered the best when its criterion has the target value. However, it is a different story for a multi-criteria decision problem. A realistic constant *k* should be defined prudently by the decision makers to represent their actual preference. The greater *k* is, the more important the corresponding criterion is. The joint utility can be calculated using Equation (39) or (40), and the concept with the maximum joint utility will be selected as the best solution.

The JUF given in Equation (40) is a linear combination of a set of quadratic functions so it is a smooth and continuous multivariate quadratic function. Apparently, the time and cost of computing the JUF will increase dramatically in proportion to the number of concepts, criteria or sample size. It is apparent that the customer desirability is based on some specific range, not a specific value, so without loss in accuracy, the use of a discrete UF can significantly simplify the computation of the joint utility. Figure 58 shows a utility function for smaller is better case, where the decision maker characterizes the degree of desirability of 6 ranges for each criterion [Messac, 1996]. A discrete UF can be established based on the desirability, with the constant value within each of the ranges. The constant value of each range is the value of the utility function at the middle point of this range. In range 6 (the infeasible range), the value of the loss function is assigned to be 0. The discrete UF for the smaller is better case is shown in Equation (41).

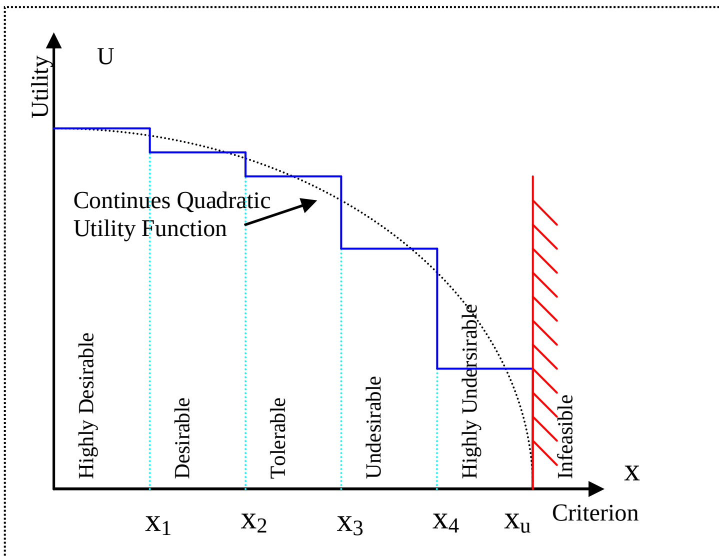


Figure 58: Utility Function for Smaller is Better

$$U_j(x_j) = \begin{cases} U_{j1} = -a_j(x_{j1}/2)^2 + k & \text{If } 0 < x_j < x_{j1} \\ U_{j2} = -a_j((x_{j1} + x_{j2})/2)^2 + k & \text{If } x_{j1} \le x_j < x_{j2} \\ U_{j3} = -a_j((x_{j2} + x_{j3})/2)^2 + k & \text{If } x_{j2} \le x_j < x_{j3} \\ U_{j4} = -a_j((x_{j3} + x_{j4})/2)^2 + k & \text{If } x_{j3} \le x_j < x_{j4} \\ U_{j5} = -a_j((x_{j4} + x_{ju})/2)^2 + k & \text{If } x_{j4} \le x_j < x_{ju} \\ U_{j6} = 0 & \text{If } x_{ju} \le x_j \end{cases} \tag{41}$$

Thus, the joint utility can be reduced to,

$$JU = \sum_{j=1}^{n}\{\sum_{t=1}^{5}[U_{jt}\int_{x_{j(t-1)}}^{x_{jt}} f_j(x_j)dx_j] + \int_{x_{j5}}^{x_{j\max}} f_j(x_j)dx_j\} \tag{42}$$

$$JU = \frac{1}{N}\sum_{j=1}^{n}\{\sum_{t=1}^{5}[U_{jt}I(x_{j(t-1)} \le x_j < x_{jt})] + I(c_{j5} \le x_j < x_{j\max})\} \tag{43}$$

where U_{jt} is the utility of j^{th} criterion in t^{th} range, $x_{j0}= x_{jl}$, $x_{j5}= x_{ju}$, $f_j(x_j)$ is the marginal PDF of j^{th} criterion. Figure 59 illustrates a joint utility function of two criteria with "smaller is better" utility.

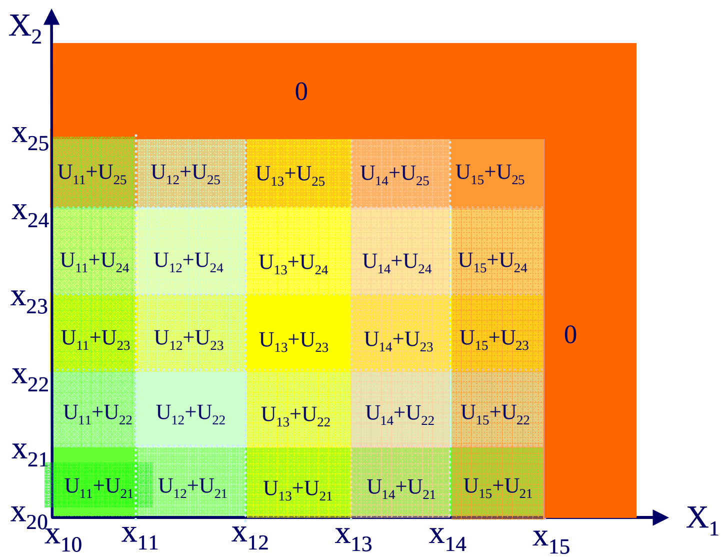


Figure 59: Joint Utility Function for Smaller is Better

To show the improvements that the proposed method achieves, the same PAV advanced concept selection problem stated in Section 4.1.2.2 is performed as an example of implementation. First of all, the utility function of each criterion requires to be constructed. To construct the discrete utility function, the parameters in Equation (41) are determined by the decision maker or designer based on their preference as listed in Table 15. Here m is the target value of the criterion. For the three given criteria, their target values are all zero since they are "cost" criteria and need to be minimized. x_j (j=1, 2,

…,5) are the limits of the 6 desirability ranges, from highly desirable to infeasible. The utility used within those ranges are calculated by Equation (41). The values of k is assigned to be 1 for all the criteria which indicates the decision maker consider the maximum utility of the criterion is one when it reaches its target value. In addition, this also means that the three criteria have the same importance.

Table 15: Discrete Utility Function

		D-D Time	DOC	Noise
Coefficient of UF	k	1	1	1
	m	0	0	0
	a	0.0625	5.92E-05	0.00016
Highly Desirable $m<x<x_1$	**Utility**	0.9844	0.9991	0.996
	x_1	1	8	10
Desirable $x_1<x<x_2$	**Utility**	0.8594	0.96	0.9359
	x_2	2	50	30
Tolerable $x_2<x<x_3$	**Utility**	0.6094	0.6672	0.7436
	x_3	3	100	50
Undesirable $x_3<x<x_4$	**Utility**	0.3398	0.3162	0.4702
	x_4	3.5	115	65
Highly Undesirable $x_4<x<x_5$	**Utility**	0.1211	0.1121	0.1694
	x_5	4	130	79
Infeasible $x_5<x<x_{max}$	**Utility**	0	0	0
	x_{max}	max (D-D Time)	max(DOC)	max(Noise)

In this study, the joint probability distribution is established using the EDF model, so the joint utility for each concept is estimated using the Equation (43). Since the tiltrotor concept is infeasible, it is eliminated before processing the selection problem. The results are shown in Table 16.

Comparing the results shown in Table 13 and Table 16, calculated using the original JPDM technique and the proposed method respectively, one can get the same goodness ranking for the PAV concept selection problem with respect to the given criteria. The

Table 16: Joint Utility and Univariate Utility of Each Concept and Criterion

Alternatives	JU	U(D-D)	U(*DOC*)	U(*Noise*)
Adv_Helicopter	0.5519	0.2095	0.2154	0.1270
Adv_Gyroplane	0.3653	0.1446	0.1170	0.1037

highest PoS and JU was obtained with the advanced helicopter concept, indicating that this concept has more viability than the other alternatives as measured by the criteria of DOC, doorstep-to-destination time and noise.

Though the results obtained from the original JPDM technique and the proposed method are the same, the accuracy offered by these two methods is different. The proposed method considers the deviation from the target value, while the original JPDM only looks at the probability distribution within the area of interest and does not care how it is distributed; in other words, no consideration is made on the variation and deviation of the distribution. This can be observed from the fact that there is a big difference in goodness for noise between the results obtained from the JPDM technique and the proposed method, shown in Table 13 and Table 16 respectively. Clearly, this difference is caused by the deviation issue. From the joint probability distribution shown in Figure 53 and Figure 54, one can see that the noises of all the solutions are greater than 65dB for all of the alternatives, which is far from target value. On the contrary, door-to-destination time and DOC are much closer to their target values. This explains the significant difference in utility values.

The relative goodness of the two competitive PAV concepts, the advanced helicopter and advanced gyroplane, obtained from the two methods is shown in Figure 60. For the result obtained from the original JPDM technique, the relative goodness of a concept is computed by normalizing its PoS by the summation of the PoS of these two concepts. Similarly, for the results obtained from the proposed method, the relative goodness of a concept is determined by normalizing its JU by the summation of these two concepts. The proposed method makes the two concepts more distinguishable: the 15% difference in goodness increases to 20% after the proposed method was applied. Figure 53 and Figure 54 show that the advanced helicopter and the advanced gyroplane have similar distributions, deviations from their respective targets. Even in this case, the proposed method still gives a more explicit result than JPDM in indicating which alternative is the best solution.

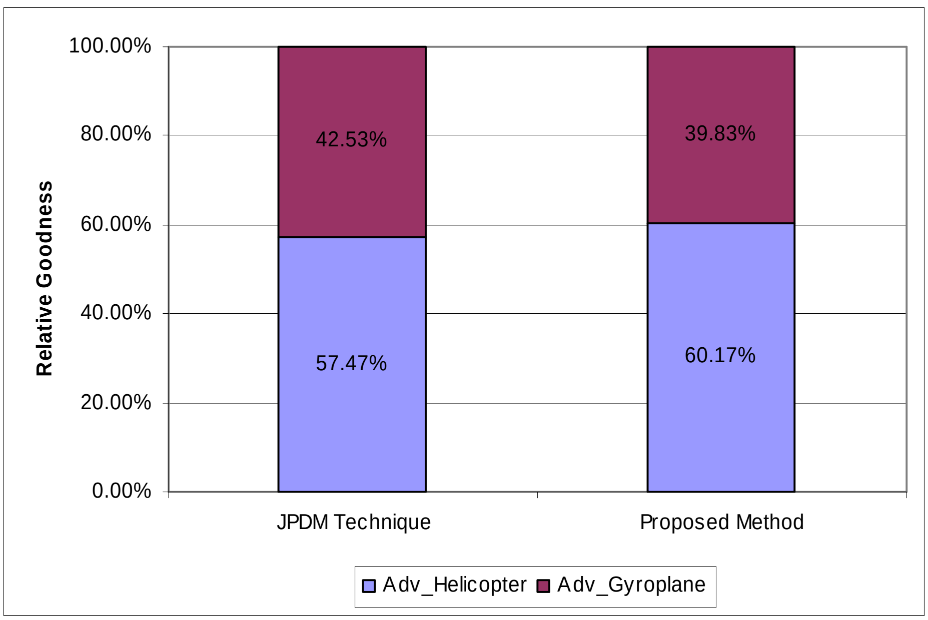


Figure 60: Comparison of JPDM and Proposed Method

When dealing with the cases described by Figure 55, the proposed approach will be much more competent than the original JPDM technique. In those scenarios, the alternative with the highest PoS but a large deviation from the target values will certainly be considered the "best" solution by the JPDM technique. This assertion is usually not consistent with the customer's preference, which is represented by a utility function. Therefore, the use of JPDM alone is not sufficient to make a wise decision, and improvement is necessary to overcome these limitations. The proposed approach not only maintains the ability to capture the system uncertainty and evaluate the multiple criteria concurrently, which are the highlights of JPDM, but also takes the deviation of the alternative's distribution into account. Thus, this method can provide more insight in a decision making process, and makes it an advanced method over the original JPDM technique. On the other hand, the proposed method relies on the accuracy of the utility function, which is always a difficult task for the decision maker and needs to be carefully determined. In this study, in particular, the parameters listed in Table 15 need to be determined prudently before the joint utility function is constructed.

4.2 *Findings and Observations*

The PAV advanced technology concept selection problem was fulfilled with the utilization of the multi-criteria decision making advisor. The most appropriate decision making method was first selected among a set of methods, and then the problem was solved using the selected method. With the intension of making better decision, the decision maker requires the preference information to be represented by a more sophisticated model. However, there is no a suitable method in the method base that can handle this revised problem. The decision making advisor provided several hints that can be used to develop a new method that has capabilities to deal with the problem. A method was proposed based on the JPDM technique using utility theory. The result shows that the improvement was achieved with the use of the proposed method.

In the method selection process, the most appropriate method is selected from the method library which store widely used decision making method. However, for a specific problem, there may be a suitable method existing out of the method library or just emerging, thus it is not possible for the advisor to suggest this method. Fortunately, the advisor system allows the new method is added to the method library, which will eventually increase the capability of the system.

The JPDM technique was identified as the most appropriate method to handle the concept selection problem. This method uses a metamodel defined by the response surface equations which captures the relationship between the design variables and attributes. The equations are quadratic polynomial function, and are only valid in the design space defined in Table 8 and Table 10. This is an important assumption that should be kept in mind. If the decision maker is willing to accept this assumption, the method can be used for solving the problem under consideration. Otherwise, another method should be identified.

By using the JPDM technique, the advanced helicopter concept was selected as the best solution to perform the PAV mission. This result was obtained based on the measurement of three criteria, door to destination time, direct operating cost, and noise. It is noteworthy that the result only holds true for the case that the concepts are evaluated by the three given criteria. If the criteria are different, additional analysis requires to be completed, and this often leads to a different result.

Several observations discover the limitations of the JPDM technique, and this technique is found not to be capable of dealing with the revised concept selection problem. A hybrid method was proposed based on the advice provided by the advisor.

The application of the proposed method yielded an improved decision for the revised problem and provided more insights to the decision making process.

The hybrid method essentially is a combination of traditional JPDM technique and utility function. It incorporates a probabilistic approach and utility theory to aircraft systems design and can accurately assess the POS of design concept. It eliminate the limitations of the traditional JPDM and offers improved performance so DMs can utilize it to make better decision with confidence. The hybrid JPDM technique is fitted in the decision tree for selecting MADM technique, as shown in Figure 61. A new question (Q3) is added to this tree diagram, which derives JPDM technique. And the hybrid JPDM technique emerges by combining the capabilities of the JPDM and overall utility function. Thus, using this adapted decision tree, a user can choose and take advantage of this advanced method to produce better result for his or her decision making problem.

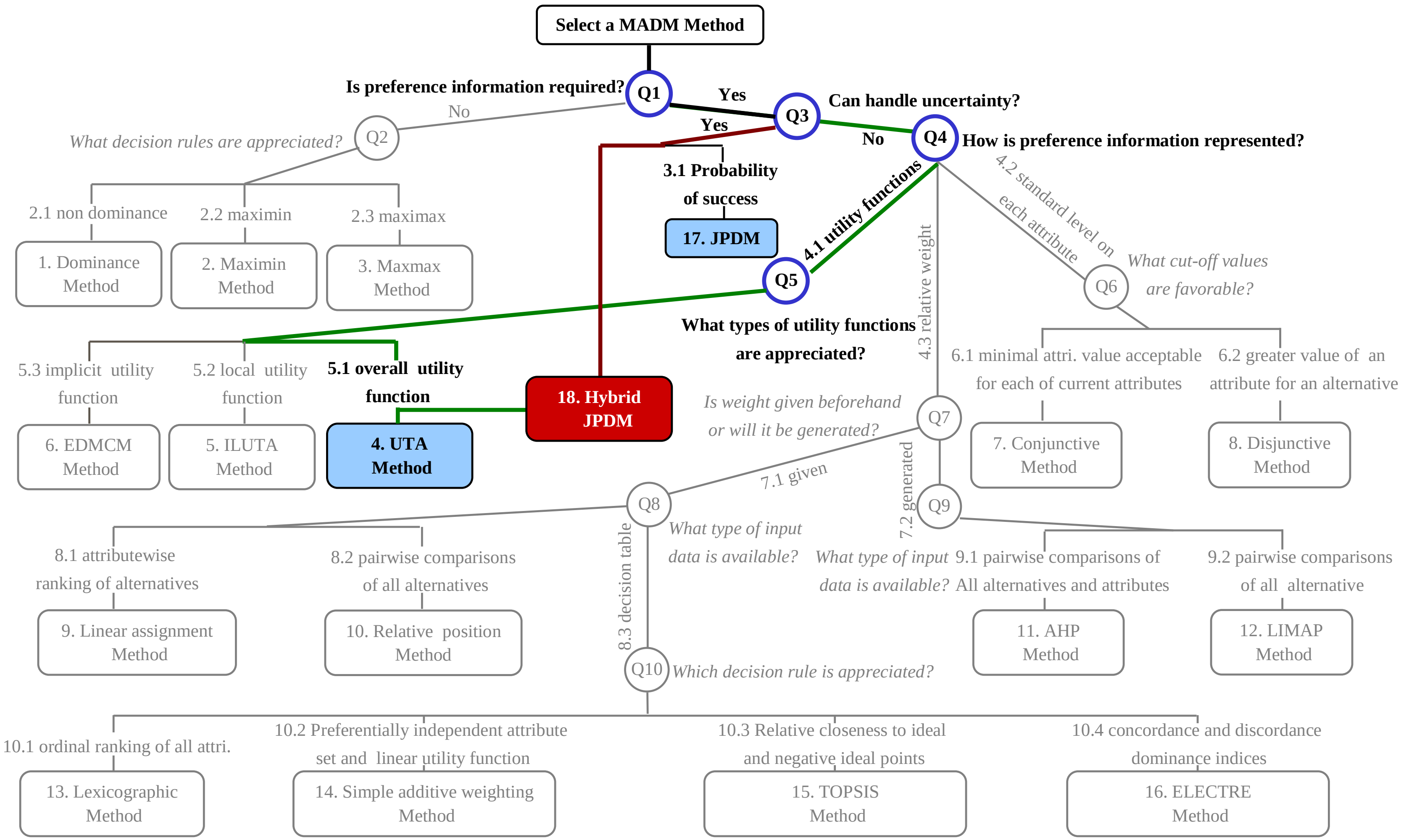


Figure 61: Decision Tree for MADM Technique Selection with Hybrid Method Fitting in

CHAPTER V

DYNAMIC DECISION MAKING UNDER UNCERTAINTY

The MIDAS process presented in Chapter III and Chapter IV provides an interactive approach that can effectively facilitate the decision making process in systems design. The most appropriate method is selected for the problem under consideration, and then is utilized to derive the solution to the decision problem by following its rigorous problem solving procedure. In this process, decisions are made based upon static information which is fixed all the time. For example, in order to capture the essence of the problem, the characteristics of the problem are explored and then used to form the basis upon which the method selection process is founded. The characteristics are the properties of the given problem and thus usually do not change. When the problem is given the information associated with the problem, such as the requirements, constraints and attribute values, will not change during the problem solving procedure. Therefore, the decision making in the MIDAS process is primarily under static conditions.

However, in many other domains, such as complex system operation, decisions are often made based on the assessment of the information which is changing over time. Under this circumstance, decision making is not any more a one-time action as it is under static conditions, but needs to be accomplished in a sequential manner. Obviously, this is a dynamic decision making process which usually requires decision maker to make multiple and interrelated decisions in a continuously changing environment [Gonzalez, 2005]. Due to the facts that uncertainties often exist in the operational environment and time pressure requires DM has real time decision making capability, these decisions are made mainly using the uncertain or incomplete information. As a result, the consequences of the decisions are hard to perfectly and deterministicly reason. This fact further exacerbates the complexity of the dynamic decision making process since uncertainty becomes an important factor that needs to be captured and analyzed in order to make proper decision.

As discussed in Section 1.1.2, an advanced approach needs to be developed to handle the problem of Dynamic Decision Making Under Uncertainty (DDMUU). This motivates the second part of this research.

5.1 DDMUU in Complex System Operation

The most difference between the dynamic and static decision making problems is the explicit reference to time. As illustrated in Figure 5, with time series being one of the properties of the decision problem, sequential decision making becomes a fundamental task faced by the decision maker. Complex system operation is one of such fields where time-dependent decisions are required to be made in a dynamic environment. In order to keep the system working functionally and effectively the decision maker, or operator, needs to make proper decisions based on the assessment of a large amount of information representing the system state. Consequently, decision makers must handle the real time data and information under time pressure. It has been already discussed on several occasions that the data and information used to make decision are usually uncertain or incomplete, thus, decision makers have to deal with uncertainties existing in the decision making process. The complexities of the decision making in complex system operation are always a challenge to human decision makers since it is usually difficult for human being to manage and organize the time-dependent information and make wise decisions based on the probabilistic assessment of the acquired information.

These issues associated with decision making in complex system operation lead to the following observation:

Observation 6: In complex systems operation, uncertainty and dynamic characteristics are two major factors that affect the decision making process, and it is usually complicating the decision process for humans.

This observation implies that decision making in complex system operation would benefit by employing an advanced approach to handle the time-dependent information and uncertain conditions.

As particularly stated in Section 1.1.2.1, in modern ship operation, more and more emphasis has been given to reducing cost and manning workload, and increasing ship survivability and mission effectiveness. This results in a requirement that the large amount of changing information needs to be rapidly processed and the decisions associated with ship operation should be made autonomously. The Integrated Reconfigurable Intelligent Systems framework is proposed as a possible solution to fulfill this requirement. With the reconfigurable systems, the IRIS designed ship will assess the incoming information and then configure itself into the mode most adequate to deal with the situation at hand. Moreover, the ship is able to be aware of its surroundings through the gathering of data from sensors onboard the vehicle and provide guidance to a human operator as to the best course of action. In general, the reactions are determined by the overall assessment which is a combination of the different assessments produced by the various systems for the same event in terms of urgency and priority. Figure 62 depicts that the IRIS designed ship is capable of self-monitoring, self-assessing and self-reacting.

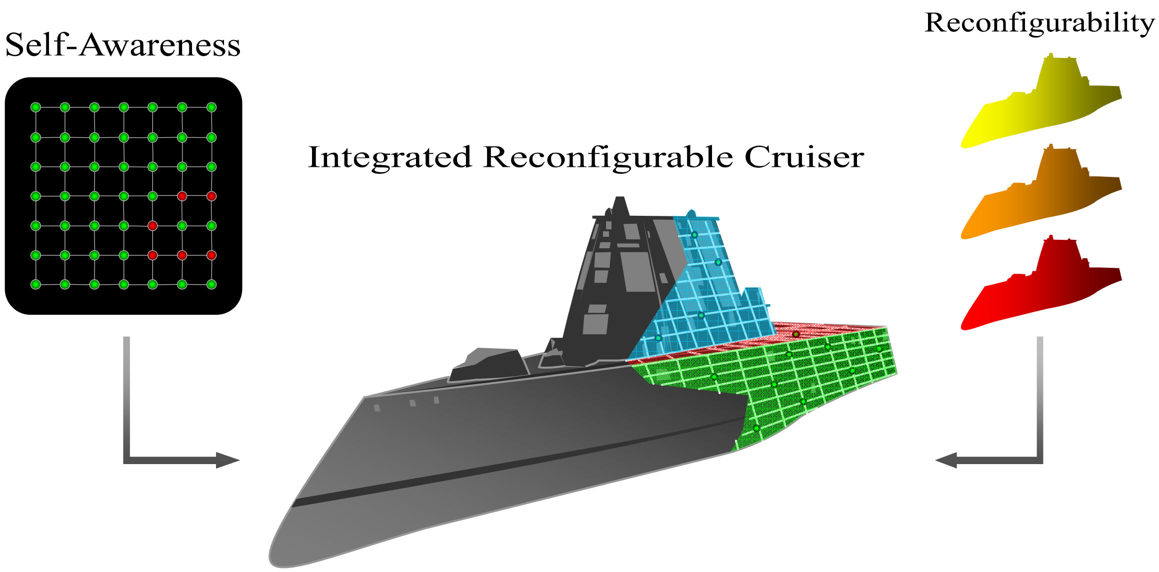


Figure 62: IRIS Concept

It has been stated in Observation 6 that a human decision maker has difficulties in manipulating the time-dependent and uncertain information. In order to increase the mission effectiveness and ship survivability and reduce operating cost, the selection of the best course of action should be automated so that fast reactions can be accomplished to deal with the situation at hand. This requires the system to possess the capability to make autonomous decisions based on the analysis of the incoming information which is uncertain and changing over time. Therefore, an approach is needed to handle the real time information and make autonomous decisions under uncertain conditions. This can be state as the research question below:

Question 6: Is there a decision making formulation that can effectively make real time decisions reacting to the current ship situation based on uncertain information? (Observation 6)

In order to develop a decision making formulation capable of making autonomous decisions for the ship operation problem, existing approaches to dynamic decision making under uncertainty will be investigated and their potentials for decision making in complex system operation will be examined in next section.

5.2 Existing Approaches to DDMUU

Dynamic decision making under uncertainty is an area where tradeoffs need to be done in an uncertain and real time domain. The complexity of this problem has attracted the attention of the researchers in both decision science and operations research. Many efforts have been made to facilitate the problem solving procedure of dynamic decision making under uncertainty. As a result, various approaches were proposed, and among these approaches three ones are widely used [Leong, 1998]. They are Dynamic Decision Analysis (DDA), Artificial Intelligence planning (AIP) and Markov Decision Process (MDP).

5.2.1 Dynamic Decision Analysis

Decision analysis, originating from the game theory and operations research [Raiffa, 1968; Keeney and Raiffa, 1976], allows the decision maker to make effective decision under risk and uncertainty. The decision analysis often employs a model which utilizes the probability theory and utility theory to obtain an expected return or cost, then decide the best course of action to be taken. The decision tree and influence diagrams are two typical analytical formalisms in decision analysis. A decision tree, also known as Classification and Regression Tree (CART) [Breiman et al., 1984], provides a graphical decision model that allows the DM to lay out options and investigate the possible consequences. Influence diagrams, also known as relevance diagrams, offer a graphical structure within which the influences among each essential element, including decisions, uncertainties and objectives are presented.

As noticed by Leong [1998], some decision analysis techniques, such as the Markov cycle tree [Beck and Pauker, 1983; Hollenberg, 1984] and stochastic trees [Hazen, 1992], were developed to deal with the dynamic decision problem. These techniques are based on the traditional decision analysis models such as decision trees and influence diagrams, and are capable of representing the stochastic process of the dynamic decision problem.

Dynamic decision analysis can provide insights into the complex decision situation and thus support the DM to select the best solution to the decision problem. The graphical model helps understand the rationale of the selection. On the other hand, however, the graphical structure does not allow the use of the admissible solution methods [Leong, 1993]. In addition, the dynamic decision analysis requires the DM to have enough knowledge to set up the model. For example, the DM needs to know the decisions and the corresponding consequences with probabilities. This causes the difficulties in applying the dynamic decision analysis methodology.

5.2.2 Artificial Intelligence Planning

Emerging in the 1960s from the works associated with the general problem solver [Newell and Simon, 1963], artificial intelligence planning is a key area in artificial intelligence. The AI planning is used to provide a plan that is a fixed sequence of actions to achieve the goals in a dynamic environment. Early research in AI planning was based on complete and deterministic information. Modern AI planning takes incomplete and uncertain information into account and is able to generate planning for a stochastic process. AI planning involves the representation of actions, reasoning about the effects of actions, and techniques for efficiently searching the space of possible plans. Significant changes have occurred in recent research: application of the methodology has become more empirical and heuristic or constrained-based search approaches become common [Blum and Furst, 1997; McDermott, 2000; Bacchus, 2001; Geffner, 2002].

AI planning is a plausible approach to the problem where dynamic decision requires to be made under uncertain conditions. It provides more flexible and expressive problem description when formulating the complex problem. However, this expressiveness may "significantly complicate search control for the optimal solutions" [Leong, 1998]. Moreover, the fixed planning is usually not suitable to handle the domain-dependent planning problems because the domain-specific information and knowledge varies with domains and the planning processes are significantly different. These facts stop AI planning from being applied smoothly in practice.

5.2.3 Markov Decision Processes

Markov Decision Processes (MDPs), also known as controlled Markov chains, were invented by Howard in 1960 [Howard, 1960]. This approach provides a mathematical framework characterized by a set of states that the system could be, a set of actions that the decision maker has to choose in each state and a transition matrix that represents the probabilities of one state transiting to other states if a certain action is executed in the original state. A reward is earned after a certain action is executed in a specific state. The solution to a MDP is an optimal policy defining which action should be taken for a given state, regardless of prior history. MDP is found to be surprisingly rich in capturing the essence of sequential decision making under uncertainty, and it was successfully applied in many areas, including operations research, control engineering, decision sciences, and so on.

The comparison of the three approaches shows that though the dynamic decision analysis and AI planning have their own advantages in handling the dynamic decision making problem with uncertainty, these two approaches have difficulties in practical application. On the other hand, the Markov decision process has been successfully implemented in many areas and appears to be a promising approach. This leads to another hypothesis:

Hypothesis 6: A well formulated Markov decision process is capable of automatically finding the best course of action to reconfigure the ship into the state adequate to deal with the situation at hand. (Question 6)

The Markov decision process is a powerful approach to the problem of dynamic decision making under uncertainty and it has the potential to facilitate the decision making analysis for the ship operation problem. Its theoretical foundation will be described in the next section.

5.3 Markov Decision Process Model

The Markov decision process is an extension of Markov chain, which is a discrete time stochastic process describing the states of a system at successive times. At these times, the system changes from one state to another or stays in the same state. The changes of state are called transitions. Markov chain must satisfy the Markov property which states that the transition of the system depends only on the current state, but not on the states in the past. Figure 63 illustrates an example of Markov chain with four states and possible transitions.

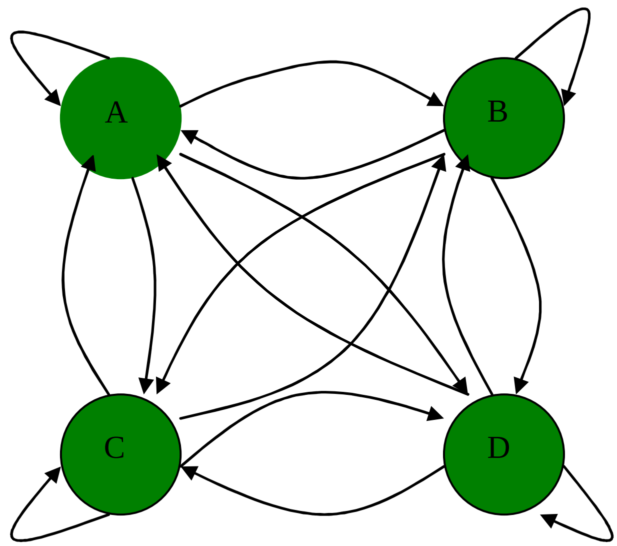


Figure 63: Example of a Markov Chain

A Markov decision process is a Markov chain with actions and rewards [Wiki, 2005]. The actions are the alternatives that have to be chosen in each state, and the execution of an action will cause the system transits to the next state. After an action is performed in a state, a reward will be earned for this state action pair. The reward of the action state pair plays a critical role in determining which action should be chosen in each state. Notice that in a MDP the best action taken in a state is not necessary the action resulting in the maximum reward in the state. This is because the action with maximum immediate reward may cause the system to transit to an undesired state in the future. Therefore, to choose the best action right tradeoffs should be made between the immediate rewards and the future gains to yield the best possible solution

5.3.1 Definition

In a MDP, a decision maker makes decisions at a set of time points, known as decision epochs. The decision epoch can be continuous or discrete. In this dissertation, the discrete decision epochs are considered and denoted by natural numbers $t \in \mathrm{N}$. Mathematically, a classical unconstrained, single-agent Markov decision process can be defined as a quadruple (4-tuple) (S, A, P, R)consisting of

- a state space $S = \{i\}$;
- a action space $A = \{a\}$, where the set of possible actions in state i is denoted by A_i, and $A = \bigcup_{i \in S} A_i$;
- a transition probability distribution function $P = [p_{iaj}]:\ S \times A \times S \mapsto P(S)$, where $P(S)$ defines the space of probability distribution over the state space S, and p_{iaj} is the probability of transiting to state $j \in S$ by executing action $a \in A_s \subseteq A$ in state $i \in S$; and
- a reward function $R = [r_{ia}]:\ S \times A \mapsto R$, where r_{ia} defines the immediate reward earned for executing action $a \in A_i \subseteq A$ in state $i \in S$.

The MDPs are classified into finite and infinite MDPs in term of the numbers of the states and actions. The study presented in this dissertation focuses on finite MDP in which the numbers of the states and actions are finite. This assumption implies that the state and action space is countable. The transition probability distribution function P is a function defined for all states $i \in S$ and the actions $a \in A_i \subseteq A$, and $\sum_j p_{iaj} = 1$ for every state i. The reward function R is defined for the transitions whose probabilities are positive, and such a transaction is referred to as a valid transition.

The Markov decision processes are typically divided into two categories, finite-horizon and infinite-horizon MDPs, based on the number of decision epochs. In the finite horizon MDPs problem, the total number of steps that the system goes through is finite and the last step is referred as T $(T \geq 1)$ and no decision is made at or after this epoch. In the infinite case, the agent stays in the system forever unless the desired goal is obtained.

A Markov decision process starts from an initial state $s_0 \in S_0 \subseteq S$ and, as an action $a \in A_s$ is taken, transits to the next state j with a probability of p_{0aj} defined in the transition probability function P. Then a new action is chosen and executed in current state, resulting in a new transition. In the process, at decision epoch t the state of the system is in i_t depending on the system's trajectory.

A MDP problem consists of a MDP model represented by a set of states, a set of actions, transition probability and reward functions. Figure 64 depicts an example of MDP problem. In this example, a startup company may be in four possible states: poor and unknown, poor and famous, rich and unknown, and rich and famous, which defines the state space of the MDP problem. In each state the decision maker of the company has to decide between saving money (S) or advertising (A), which constructs the action space of the MDP problem. The transition probabilities of one state changing to the other with a chosen action are listed in Figure 64. The rewards for each state are also listed in Figure 64. As can be seen, the rewards are specified as: 0 for poor and unknown, 0 for poor and famous, 10 for rich and unknown, and 10 for rich and famous. $\gamma \in (0,1]$ is the discount factor used to convert the future reward to present value, and in this example it is assigned as 0.9.

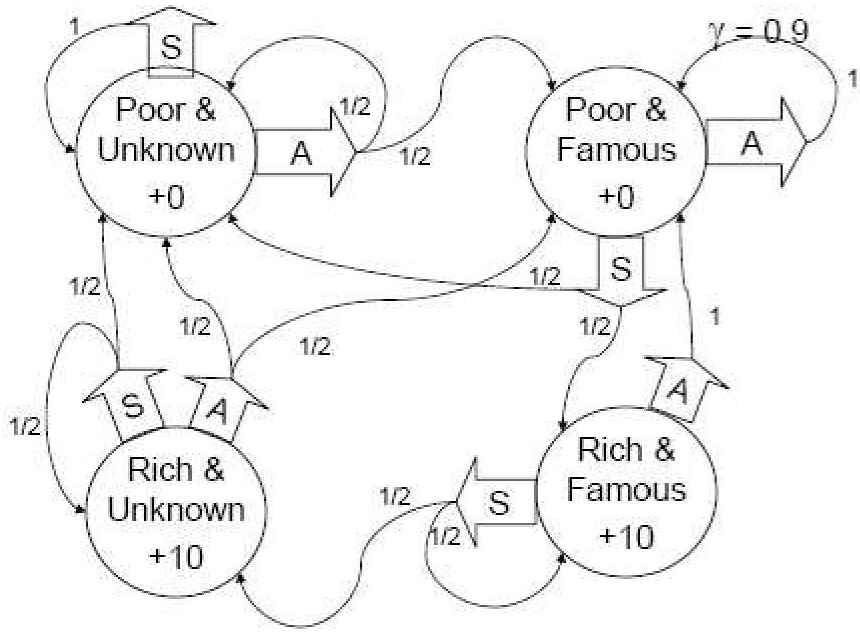


Figure 64: An Example of Markov Decision Process Problem [Moore, 2005]

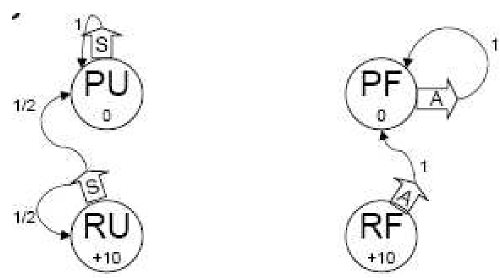
For this MDP problem, it is assumed that in each system state one action must be taken. This assumption leads to several questions that need to be answered: 1) how to determine the action to be taken in a specified state? 2) is the action that maximizes the immediate reward the best choice? In order to answer these questions a decision rule should be adopted to specify which action should be taken in each state.

5.3.2 Policy

The solution to the Markov decision processes is defined as policy. A policy, denoted as π, is a mapping from states to actions, which specifies the action to take for a given state, regardless of prior history. A stationary policy is defined as a policy that does not depend on time but only on the current state. It should be noted that almost all the work related to the Markov decision process is to find an optimal stationary policy. The stationary policy can be further classified into two categories: deterministic policy and randomized policy. A deterministic policy always takes the same action for a specific state while a randomized policy chooses an action a for a state i based on some probability distribution over a set of actions $a \in A_i \subseteq A$. A pure policy is referred to as a stationary deterministic policy, i.e., the action taken in each state is fixed. It is clear that a Markov decision process combined with a pure policy would reduce to a Markov chain.

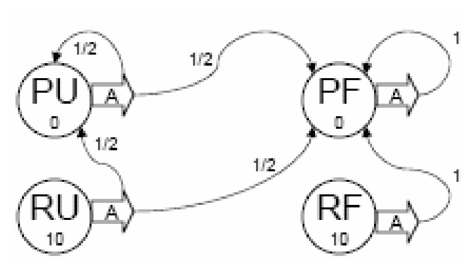
A randomized policy, denoted as $\pi = [\pi_{ia}]$, is a mapping of state-action pair to probability distribution, where π_{ia} defines the probability of choosing action a when the system is in state i. The randomized policy has such a property: $\sum_a \pi_{ia} = 1$, indicating that an action has to be chosen in each state [Dolgov and Durfee, 2004]. Obviously, a

STATE → ACTION	
PU	S
PF	A
RU	S
RF	A



(a) Policy 1

STATE → ACTION	
PU	A
PF	A
RU	A
RF	A



(b) Policy 2

Figure 65: Example Policies [Moore, 2005]

pure policy can be considered as a randomized policy that has only one action for each state and the probability of being chosen is 1.

Two different policies for the startup company example are illustrated in Figure 65. The table on the left describes a policy and the figures on the right depict, starting from states RU and RF, respectively, how the state transits from one to another by following the corresponding policies. It is apparent that the policies shown in Figure 65 are pure policies.

5.3.3 Techniques to Solve MDP Problem

Given a state in a Markov decision process, a decision maker often confronts the situation of which action to choose. Since the process is sequential, an action performed in a state not only has effect on immediate next state but also has effect on the following states. Figure 66 presents an example showing how an action affects the future states. Therefore, choosing the best action in each state should be based on the assessment of more than the immediate effects of the action and a tradeoff should be done between the immediate rewards and future gains [Cassandra, 2003].

A policy is preferred over the other if it obtains a better value of the evaluation criterion which often is some cumulative function of the rewards, such as the expected total rewards, the expected discounted rewards, or the average expected rewards. Assuming the expected discount rewards is employed as the criterion to evaluate the policy, if a Markov decision process starts from state i, the expected discounted sum of future rewards $V(i)$ is given by Equation (44)

$$V(i) = r_{ia} + \gamma \sum_{j=1}^{N} p_{iaj} V(j) \tag{44}$$

where r_{ia} is the immediate reward earned by executing action a in starting state i, γ is the discount factor which has property of $\gamma \in (0,1]$. The goal of the Markov decision processes is to find an optimal policy that maximizes the value function, as shown by Equation (45).

$$V(i) = \max_a \left[r_{ia} + \gamma \sum_{j=1}^{N} p_{iaj} V(j) \right] \tag{45}$$

Equation (45) is also known as Bellman optimality equations [Bellman, 1957]. From Equation (45) one can see that the value of a policy depends upon the initial state of the process. It has been proved that for an unconstrained MDP there exists an optimal policy such that for any initial state there is no better option than to follow the policy, i.e., $\forall$ policy π and initial state $i \in S$, $\exists$ optimal policy π^* that $V(\pi^*, i) \geq V(\pi, i)$.

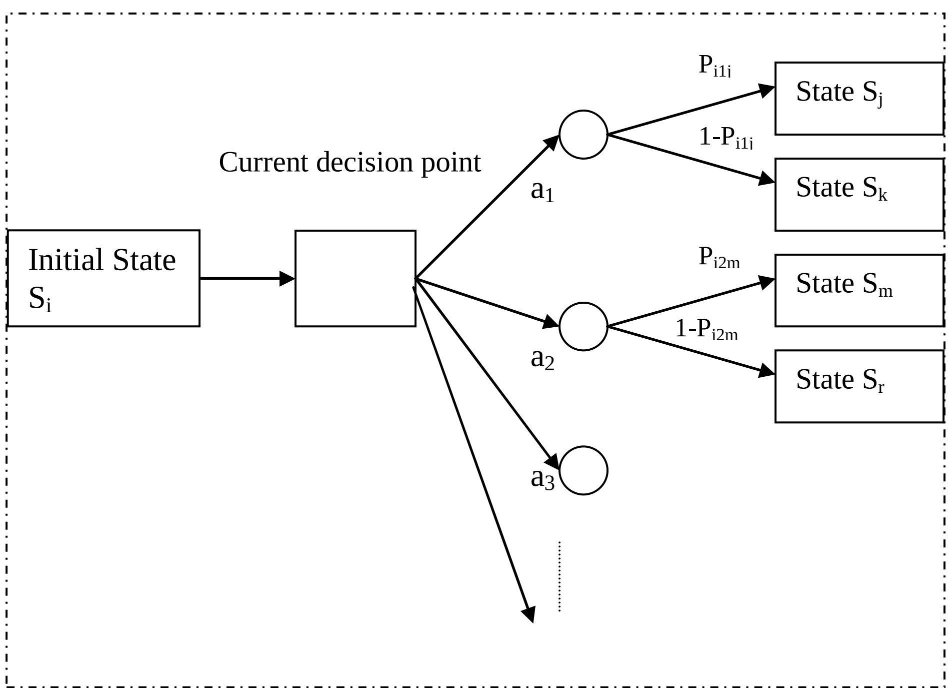


Figure 66: Effects of Actions on the Future States

There are three widely used algorithms for determining the optimal policy to a Markov decision process. They are value iteration, policy iteration and linear programming.

5.3.3.1 Value Iteration

The value iteration algorithm, based on the Bellman optimality equations, is a well-known algorithm for producing an optimal policy to a discounted Markov decision process. This algorithm calculates the value function, given by Equation (45), by finding a sequence of value functions, each one derived from the previous one.

The first step of the value iteration algorithm is to find the value function for a horizon length of 1 for each state. This is quite simple. Since the horizon length is 1 the immediate reward will be the value function for each state. The value function needs to be maximized therefore the action which incurs the highest immediate reward is selected as the decision.

The second step is to compute the value function for a horizon length of 2, which is the summation of the immediate rewards and the value of the action that will be chosen. Since the values of each state has been calculated for the horizon length 1, the value for horizon length 2 can be obtained by adding the immediate effects of each of the possible actions to the already computed value function to find the action with the best value. It is worth emphasizing that after the action is made in horizon length 1 the states out from the initial state is determined by the transition probability function which shows the probabilistic effects of the actions.

The algorithm then iterates again to compute the value function for horizon 3 using the horizon 2 value function. This iteration continues until we have found the value function for the desired horizon, or until the value function is converged. Finally, the optimal policy is derived from the maximum value function, given by Equation (46). The algorithm of value iteration is described below.

$$\pi^* = \arg\max_a \left[r_{ia} + \gamma \sum_{j=1}^{N} p_{iaj} V(j) \right] \tag{46}$$

Algorithm 2: Value Iteration

Inputs: Immediate rewards for each state r_{ia}, transaction probability distribution function $P = [p_{iaj}]$, initial state i, a small positive number ε

Outputs: Value function $V^*(i)$, optimal policy π^*

Calculate the immediate rewards for all states for horizon length of 1

$V^1(i) = \max_a (r_{ia})$ for all i

Calculate the value function for all states for horizon length of 2

$V^2(i) = \max_a \left[r_{ia} + \gamma \sum_{j=1}^{N} p_{iaj} V^1(j) \right]$ for all i

n=2

while $\max_a \left| V^n(i) - V^{n-1}(i) \right| > \varepsilon$ **do**

n=n+1

` $V^n(i) = \max_a \left[r_{ia} + \gamma \sum_{j=1}^{N} p_{iaj} V^1(j) \right]$ for all i

end while

$V^*(i) = V^n(i)$

$\pi^* = \arg\max_a \left[r_{ia} + \gamma \sum_{j=1}^{N} p_{iaj} V^*(j) \right]$

5.3.3.2 Policy Iteration

Policy iteration, proposed by Howard (Howard, 1960), is another effective algorithm to find the optimal policy for a Markov decision process. This algorithm manipulates the policy directly, rather than finding it indirectly via the optimal value function $V^*(i)$. The first step in policy iteration algorithm is to randomly choose a policy π^0 as the starting point. Then the expected rewards $V^0(i)$ for all states along the Markov process are calculated using π^0. After the value of each state $V^0(i)$ under current policy is known, it may possible improve the value by changing the first action taken. If this is the case, a new policy will be produced based on the value of the state calculated using previous policy. This step is given by Equation (47) and is guaranteed to strictly improve the performance of the policy. The above steps are repeated until the iteration is converged, and at this point the optimal policy π^* is reached. Figure 67 shows the process of the policy iteration algorithm [Sutton and Barto, 1998], where $\xrightarrow{E}$ denotes a policy evaluation and $\xrightarrow{I}$ denotes a policy improvement.

$$\pi^1(i) = \arg\max_a \left[r_{ia} + \gamma \sum_{j=1}^{N} p_{iaj} V^0(j) \right] \tag{47}$$

$$\boldsymbol{\pi_0 \xrightarrow{E} V^{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V^{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \cdots \xrightarrow{I} \pi^* \xrightarrow{E} V^*}$$

Figure 67: Policy Iteration Algorithm

The policy iteration algorithm is described in Algorithm 3. Notice that in each policy evaluation, the value function needs to be computed iteratively until it converges. Each policy improvement is evaluated once and the algorithm is complete when the policy obtained is equal to the previous one.

Algorithm 3: Policy Iteration

Inputs: Immediate rewards for each state r_{ia}, transaction probability distribution function $P = [p_{iaj}]$, initial policy π^0

Outputs: Value function $V^*(i)$, optimal policy π^*

Initialization

π^0, $V^0(i) = 0$

1 for m=1 **do**

Policy Evaluation

for n=1 **do**

$$V^n(i) = \max_a \left[r_{ia}^{\pi^m}{}_{ia} + \gamma \sum_{j=1}^{N} p_{iaj}^{\pi^m} V^1(j) \right] \text{ for all } i$$

until $\max_a \left| V^n(i) - V^{n-1}(i) \right| < \varepsilon$

end for

$V^*(i) = V^n(i)$

Policy Improvement

$$\pi^m = \arg\max_a \left[r_{ia}^{\pi^m}{}_{ia} + \gamma \sum_{j=1}^{N} p_{iaj}^{\pi^m} V^*(j) \right]$$

If $\pi^m = \pi^{m-1}$

stop

else

go to 1

end if

end for

5.3.3.3 Modified Policy Iteration

In practice, both value iteration and policy iteration have their advantages and disadvantages: value iteration is much faster per iteration but takes more iteration to complete, while policy iteration takes fewer iterations, but is relatively slower in the policy evaluation step. Puterman proposed an algorithm, referred to as modified policy iteration, which is a combination of the two algorithms and can speedup the calculation [Puterman, 1994]. This algorithm, instead of finding an exact value for $V^*(i)$, finds an approximation to $V^*(i)$ in policy evaluation step by performing a few steps of value iteration where the policy is held fixed over successive iterations. This has been shown to produce an optimal policy within shorter iteration time [Kaelbling et al., 1996].

5.3.3.4 Linear Programming

Dynamic programming, including the value iteration and policy iteration algorithms, offers effective approaches to solving the operational problems in a Markov decision process. The goal of these algorithms is to calculate the expected rewards that can be obtained by solving the Bellman equation, as shown in Equation (45). These algorithms have a rigorous process in which the value function needs to be calculated for all the states in each iteration. However, as the number of state variables increases, the size of the state space will typically grows exponentially, which is known as the curse of dimensionality. This causes the dynamic programming formulation to become intractable for solving this type of problems.

Linear programming, with the pioneering work of D'Epenoux [1963], was proposed as one of the approaches to deal with this difficulty. Linear programming is an area of linear algebra in which the goal is to maximize or minimize a linear function $f(\vec{x})$ of n variables $\vec{x} = (x_1, x_2, \cdots, x_n)$ on a region whose boundary is defined by linear inequalities and equations. An unconstrained single-agent Markov decision process can be formulated as a linear programming, given by Equation (48):

$$
\begin{aligned}
&\max \sum_i \sum_a x_{ia} r_{ia} \\
&s.t. \quad \sum_a x_{ja} - \sum_i \sum_a p_{iaj} x_{ia} = \alpha_j \\
&\qquad\ x_{ia} \geq 0
\end{aligned}
\tag{48}
$$

or the first constraint can be written as

$$\sum_i \sum_a (\delta_{ij} - p_{iaj}) x_{ia} = \alpha_j \text{, where } \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}.$$

The optimization variables $\mathbf{x} = \{x_{ia}\}$ corresponding to a policy π are referred as the occupation measure which can be interpreted as expected frequency that action a is chosen in state i. Therefore the occupation measure is essentially a probability measure over the set of state-action pairs (i,a) and it has the property that the expected total reward to that policy can be expressed as the expectation of the immediate reward with respect to this measure [Altman, 1999], as shown by Equation (48). The policy π can be obtained from the **x** as:

$$\pi_{ia} = \frac{x_{ia}}{\sum_a x_{ia}} \tag{49}$$

$\alpha = \{\alpha_j\}$ is a measure of initial probability distribution over the states. And thus the first constraint in Equation (48) can be considered as the conservation of probability and is not an external constraints imposed on the problem. This constraint can be interpreted as the expected frequency that state j is visited less the expected frequency that j is transmitted from all state-action pairs should be equal to the expected frequency of starting in state j [Dolgov and Durfee, 2004]. The second constraint clearly indicates that the probability of taking action a in state i is nonnegative.

CHAPTER VI

SOLUTION METHOD TO MULTI-AGENT MARKOV DECISION PROCESS AND ITS IMPLEMENTATION

In Chapter V, the approaches to deal with single-agent unconstrained Markov decision process are discussed. They provide various algorithms for efficiently achieving the goal of the standard MDP: finding an optimal policy which maximizes the expected total rewards. In complex system operation, the system often consists of multiple subsystems which provide necessary functionalities to the system. Either these subsystems work independently so that there are no correlations between their actions, or they work dependently, in which case the actions of one subsystem can influence the actions of the other ones. However, the subsystems will be considered coupled together regardless their dependencies when the resource allocation problem is taken into account

6.1 Multi-agent Resource Allocation Problem

6.1.1 Problem Description

A complex system, such as an aerospace vehicle or a naval ship, relies on various subsystems to provide the necessary functions in order to successfully perform the desired mission. To maintain their functionalities, all the subsystems need necessary resources, such as electrical power, chilled water, or fuel, to work properly. These resources are often limited and shared by all the subsystems. This can be summarized as an observation below:

Observation 7: In complex system operation, limited resources are often shared by various subsystems.

It has been stated in Section 5.1 that in order to increase mission effectiveness, a modern ship should be able to reconfigure itself into the state that is most suitable for the situation under consideration. The reconfiguration is accomplished by taking the best action in the current state based on the assessment of the incoming information. Thus, during the ship operation a best course of action needs to be identified and taken in order to increase the mission effectiveness.

The execution of the actions often consumes resources. Since different subsystems need to work together to realize various functions required to complete the desired actions, they require different amounts of resources to function properly. Therefore, there is a clear need for resource allocation among the subsystems in order to ensure their performance and satisfy the ultimate goal of the system operation. The completion of the resource allocation will reconfigure the ship to a new state most suitable to deal with the situation at hand. Hence, the realization of the best course of action and the resource allocation problem are closely coupled, that is, to find the best course of action a resource allocation problem needs to be solved.

Apparently, in the resource allocation problem, the subsystems are coupled regardless their work dependencies because their resource consumptions are constrained by the total available resources. In addition, the resources may be limited so that not all subsystems can obtain required resource. This implies that coordination must be done among the subsystems when the resources are allocated. Therefore, an approach is needed to facilitate the effective resource allocation analysis. This leads to the following research question:

Question 7: Is there a mathematical formulation for the resource allocation that can effectively distribute the shared resource to each subsystem? (Observation 7)

6.1.2 Problem Assumption

As discussed before, in the ship operation, various subsystems require different resources to perform the necessary actions. The resources are often limited and shared by all the subsystems. Thus, one of the primary tasks in ship operation is to allocate the resources to the subsystems so that they can provide functions required to fulfill the desired action.

In this document, subsystems are assumed to operate independently. This assumption implies that the action of one subsystem does not depend on or result in an action of the other subsystem. However, this assumption does not mean the subsystems can take any action without constraint. Since they require resources to execute the actions and the total resources are limited, the subsystems are coupled by the resources that they share. In other words, the subsystems are independent when they operate but coupled via the shared resources when the resource allocation is taken into account. Once the subsystems obtain the required resources, they operate completely independently. In this case, we say the subsystems are loosely coupled.

A subsystem is often considered as an agent in the system operation. In the actual operation, if one event occurs, the ship will sense and assess the situation and make decisions based on the mission, environment and ship status. The decisions will indicate what actions should be taken by each agent with the consideration of not overusing the available resources. To perform the desired actions, the shared resources need to be distributed to the corresponding agents. The execution of the action will lead the ship to a new state, and consequently further decisions need to be made in the new state in order to achieve the ultimate goal of the system operation. This state-action step will repeat at each decision epoch until the completion of the operation. Notice that whenever a decision needs to be made on what actions should be taken in a state, the resource allocation problem will be executed. As a result, the resource allocation problem will affect the selection of the action taken in each state by imposing the available resource constraint on the decision making process.

The goal of the system operation is to identify and take on the best course of action to maximize the objective of the operation. The best course of action can be obtained by solving a resource allocation problem. The accomplishment of resource allocation needs sequential decisions to be made in a stochastic process. Clearly, this problem is suitable to be formulated as a Markov decision process. The differences between this resource allocation problem and a classic Markov decision process are that multiple agents are involved and constraints are imposed on the problem. In order to solve the resource allocation problem, a hypothesis is made:

Hypothesis 7: The resource allocation problem can be formulated as a multi-agent Markov decision process subject to the resource available constraint. (Question 7)

6.2 Solution Method

A multi-agent Markov decision process is used to formulate the sequential decision making for multiple agents in a stochastic process. In this process, each agent has its own state space, action space and transition probability function. In addition, the immediate reward for each state action pair may vary with the agent. In this document, we assume there exist M agents and, without loss of generality, each agent has the same state space $S=\{i\}$ and action space $A=\{a\}$. To handle the resource allocation problem, the multi-agent MDP needs to be capable of dealing with constraints.

6.2.1 Related Work

The resource allocation problem discussed above has attracted some researchers' attentions and as a result several methods have been developed. A straightforward approach is to formulate this multi-agent MDP as a large MDP over the joint state space and action space of all agents [Boutilier, 1999]. However, this approach suffers from the "curse of dimensionality" since the number of joint state will be N^M and the number of joint action will increase to $(NK)^M$, where $N=|S|$ and $K=|A|$. The size of the joint state and action spaces will increase exponentially with the increase of the number of agents, states or actions, thus making it intractable to use the traditional techniques described in Section 5.3.3 for solving the multi-agent MDP problem.

Another approach for solving the multi-agent MDP is to decompose the global MDP into several independent or loosely coupled local MDP problems, and then local MDP problems are solved independently and their policies π_1, π_2, …, π_l are combined to produce a joint policy $\pi=(\pi_1,\pi_2,\cdots,\pi_l)$ to the global MDP problem. This problem decomposition approach has been adopted studied by a few researchers in their study [Boutilier et al., 1997; Meuleau et al., 1998; Singh and D., 1998; Xuan et al., 2000].

6.2.2 Loosely Coupled Markov Decision Process

Dolgov and Durfee noticed that the existing methods "either do not allow one to completely avoid the explicit enumeration of the joint states and actions or provide only approximate solutions to the global policy optimization problem" [Dolgov and Durfee, 2004]. They presented a new method that allows one to fully explore the structure of the global MDP problem and does not sacrifice the optimality. This method formulates a resource allocation problem as a loosely coupled MDP and utilizes linear programming to handle the external constraints representing the resource limitations.

The linear programming formulation for an unconstrained multi-agent MDP with total expected reward as the optimization criterion is given by Equation (50). Clearly, Equation (50) with the capability to deal with multi-agent MDP is an extension of Equation (48). It can be seen that the expected total reward of the global MDP is a linear combination of the ones of the local MDPs. This formulation is base upon the assumption that the agents operate independently, as discussed in Section 6.1.2.

$$\begin{aligned}&\max \sum_m \sum_i \sum_a x_{ia}^m r_{ia}^m \\ &s.t. \quad \sum_i \sum_a (\delta_{ij} - p_{iaj}^m) x_{ia} = \alpha_j^m \\ &\qquad x_{ia}^m \geq 0\end{aligned} \tag{50}$$

When the resource allocation is taken into account, external constraints are added to the equation to prevent the resource from being overused. Dolgov and Durfee formulated a multi-agent MDP with operationalization constraints [Dolgov and Durfee, 2004] based on Equation (50). An agent is said to exhibit operationalization constraints if a particular policy is not operational due to the resource limitation. Equation (51) [Dolgov and Durfee, 2004] presents the constrained linear programming algorithm for solving a constrained multi-agent MDP problem.

$$
\begin{aligned}
&\max \sum_m \sum_i \sum_a x_{ia}^m r_{ia}^m \\
&s.t \quad \sum_i \sum_a (\delta_{ij} - p_{iaj}^m) x_{ia}^m = \alpha_j^m \\
&\qquad \sum_m \theta(\sum_a c_{ak}^m \sum_i x_{ia}^m) \le \hat{c}_k \\
&\qquad \sum_k q_{kl} \theta(\sum_a c_{ak}^m \sum_i x_{ia}^m) \le \hat{q}_l^m \\
&\qquad x_{ia}^m \ge 0
\end{aligned}
\tag{51}
$$

where $\theta(z) = \begin{cases} 0 & z = 0 \\ 1 & z = 1 \end{cases}$, δ_{ij} is Kronecker delta, defined as $\delta_{ij} = \begin{cases} 0 & \mathrm{i} = \mathrm{j} \\ 1 & \mathrm{i} \ne j \end{cases}$ and

- p_{iaj}^m represents the probability that agent m transits to state j if action a is executed in state i.
- r_{ia}^m defines the reward agent m earns for executing action a in state i.
- c_{ak}^m defines the action resource requirements, that is, if agent m requires resource k in order to execute action a then c_{ak}^m =1, otherwise c_{ak}^m =0.
- $\hat{c}_k$ defines the total amount of resource k available to be shared by all the agents in the group
- q_{kl} defines the amount of cost in type l resulting from consuming a unit of resource k.
- $\hat{q}_l^m$ defines the upper bounds on how much cost l the agent m can incur.
- α_i^m is the initial probability distribution of the state i for agent m.

Equation (51) is not a linear programming problem since the step function θ is nonlinear. In order to solve this problem, Dolgov and Durfee reduced the above problem to a Mixed Integer Linear Program (MILP) by rewriting the step function θ and normalizing the occupation measure **x** [Dolgov and Durfee, 2004]. The MILP is given by Equation (52).

$$
\begin{aligned}
&\max \sum_m \sum_i \sum_a y_{ia}^m r_{ia}^m \\
&s.t \quad \sum_i \sum_a (\delta_{ij} - p_{iaj}^m) y_{ia}^m = \frac{\alpha_j^m}{X} \\
&\qquad \sum_m \Delta_k^m \le \widehat{c}_k \\
&\qquad \sum_k q_{kl} \Delta_k^m \le \widehat{q}_l^m \\
&\qquad \sum_a c_{ak}^m \sum_i y_{ia}^m \le \Delta_k^m \\
&\qquad \mathrm{y}_{ia}^m \ge 0, \quad \Delta_k^m \in \{0,1\}
\end{aligned}
\tag{52}
$$

where $y_{ia}^m = \dfrac{x_{ia}^m}{X}$, $X \ge \sup \sum_a c_{ak}^m \sum_i x_{ia}^m$, $\Delta_k^m = \theta(\sum_a c_{ak}^m \sum_i x_{ia}^m)$.

6.2.3 Recyclable Resource

The mixed integer linear program described in Section 6.2.2 formulates a loosely coupled multi-agent MDP process to solve a resource allocation problem with operationalization resources. The operationalization resources include tools, equipments and personnel, and are often represented as discrete variables. This type of resource is reusable. This implies that once an agent obtains such a resource the agent can keep it all the time and use it to perform multiple actions. The step function in second constraint of the Equation (51) indicates that if an agent expects to perform the actions which need an operationalization resource k, then the agent will get one unit of this resource regardless how often the agent use the resource.

The execution resources like time, fuel and money are consumable. If an agent uses one unit of this type of resource, it will be reduced from the total available resource. Thus, when the agent needs to perform another action that requires this resource, it has to request one more unit from the rest of the available resource. Hence, the execution resource depends on the frequency of its usage. This type of resources is modeled using the following linear constraint.

$$\sum_m \sum_i \sum_a h_{iau}^m x_{ia}^m \le \widehat{h}_u \tag{53}$$

where h_{iau}^m defines if agent m performs action a in state i, it will consume h_{iau}^m units of resource u, and $\widehat{h}_u$ represents the upper bound of the expected resource consumption.

However, the constrained multi-agent MDP approach explained in Section 6.2.2 did not model the recyclable resource, which is a common type of resource often needed in complex system operation. The recyclable resource such as chilled water of a chilled water system is neither reusable nor consumable, but recyclable. Typically, chilled water is produced by the chiller and distributed to the subsystems as a coolant fluid. The agents producing heat load transfer their heat through a heat exchanger to the chilled water and get cooled. Once a unit of chilled water completes the heat exchange with the agent, it absorbs the heat and its temperature increases. Hence, the water, which has become “hot” water, can not be reused to cool the system before it returns to the chiller and is reproduced as chilled water. Since the capacity of the chiller is determined, the chilled water produced by the chiller is limited per cycle. This resource limitation imposes a constraint on to the resource allocation problem and needs to be modeled in the global MDP.

$\hat{g}_w$ is defined as upper bound of a recyclable resource w, and g_{iaw}^m represents the amount of this recyclable resource consumed by agent m if action a is executed in state i. Therefore, the expected resource w that an agent m can consume at one decision epoch is given by

$$\sum_i p_i^m \left[\sum_a \pi_{ia}^m g_{iaw}^m \right]$$

where $\pi_{ia}^m = \dfrac{x_{ia}^m}{\sum_a x_{ia}^m}$ is a policy defining the probability that agent m takes action a in state i, and $p_i^m = \dfrac{\sum_a x_{ia}^m}{\sum_i \sum_a x_{ia}^m}$ defines the probability that agent m can be in state i.

To avoid overusing the limited recyclable resource, the total resource required by the agents should not greater than the available resource, which is modeled by a constraint given by Equation (54)

$$\sum_m \sum_i \left(\frac{\sum_a x_{ia}^m}{\sum_i \sum_a x_{ia}^m} \right) \left[\sum_a \left(\frac{x_{ia}^m}{\sum_a x_{ia}^m} \right) g_{iaw}^m \right] \le \hat{g}_w \tag{54}$$

Equation (54) can be reduced to

$$\sum_m \left(\frac{\sum_i \sum_a x_{ia}^m g_{iaw}^m}{\sum_i \sum_a x_{ia}^m} \right) \le \hat{g}_w \tag{55}$$

It is clear that the term $\sum_i \sum_a x_{ia}^m$ can be interpreted as the total expected number that the agent m is visited (i.e. total expected decision epoch). Since all the agents work together and at any decision epoch each agent has to be visited once, the total decision epoch for each agent should be equal. Thus, Equation (55) can be reduced as:

$$\sum_m \sum_i \sum_a x_{ia}^m g_{iaw}^m \leq \sum_i \sum_a x_{ia}^t \widehat{g}_w \tag{56}$$

where $t \in \{1, 2, \cdots, M\}$. Equation (56) can be further rewritten as:

$$\sum_m \sum_i \sum_a x_{ia}^m (g_{iaw}^m - \lambda_m \widehat{g}_w) \leq 0 \tag{57}$$

where $\lambda_m = \begin{cases} 1 & \text{if m} = rand\{1,2,...,M\} \\ 0 & \text{else} \end{cases}$

With the recyclable resource effectively modeled, the MILP problem described in Section 6.2.2 can be capable of dealing with three types of the resources: reusable, consumable and recyclable resources. Mathematically, the improved method is given by Equation (58).

$$\begin{aligned} &\max \sum_m \sum_i \sum_a x_{ia}^m r_{ia}^m \\ &s.t \quad \sum_i \sum_a (\delta_{ij} - p_{iaj}^m) x_{ia}^m = \alpha_j^m \\ &\qquad \sum_m \theta(\sum_a c_{ak}^m \sum_i x_{ia}^m) \leq \widehat{c}_k \\ &\qquad \sum_k q_{kl} \theta(\sum_a c_{ak}^m \sum_i x_{ia}^m) \leq \widehat{q}_l^m \\ &\qquad \sum_m \sum_i \sum_a h_{iau}^m x_{ia}^m \leq \widehat{h}_u \\ &\qquad \sum_m \sum_i \sum_a x_{ia}^m (g_{iaw}^m - \lambda_m \widehat{g}_w) \leq 0 \\ &\qquad x_{ia}^m \geq 0 \end{aligned} \tag{58}$$

Notice that if there is operationalization resource shared by the agents, Equation (58) can be transformed into the form similar with Equation (52). In this case, it can be solved by using the MILP technique. On the other hand, if there is no operationalization resources, the second and third constraints will be taken out. It is clear that the problem will become to a linear program and can be solved using linear programming technique.

6.2.4 Dependency

Equation (58) can be utilized to solve the resource allocation problem for a constrained multi-agent Markov decision process. The equation can either be reduced to a mixed integer linear program or linear program problem depending on the existence of operationalization resource. In this formulation, the objective function, total expected reward, is a linear combination of individual agent's expected total reward, which is based on the assumption that the agents operate independently. This formulation is well suited for the system whose agents operate independently. In the case that dependencies exist among some agents, the linear program can be constructed by abstracting these dependent agents as one independent agent. The states of the abstracted agent can be derived from the dependent agents, and the actions will be the joint actions over these states. With the utilization of abstraction, the system with dependent agents can also be modeled using Equation (58) and the linear programming technique can be employed to obtain the optimal policy for the abstracted multi-agent Markov decision process.

6.2.5 Resource Allocation Formulation

The resource allocation problem is formulated as a constrained multi-agent Markov decision process which can be solved using Equation (58). The solution to Equation (58) is an optimal policy that specifies the action to be taken by each agent in a specific state. Thus the resources consumed by the agent to execute the action are essentially the solution to the resource allocation problem. Therefore, the resource allocation problem and the policy optimization problem are closely coupled.

The resource allocation process is depicted in Figure 68. The process starts from recognizing the system state at time t by assessing the information of the mission, operational environment and ship status collected from the console and sensors. Based on the ship state, the resources required by each agent, such as propulsion system, weapon system and radar system, and their priorities can be obtained. The element R_{ijreq} in the resource required matrix represents the amount of resource j required by agent i for the current state. Similarly, the element r_{ij} in priority matrix defines the priority of agent i requiring resource j. Then the information is utilized as inputs by a multi-agent MDP

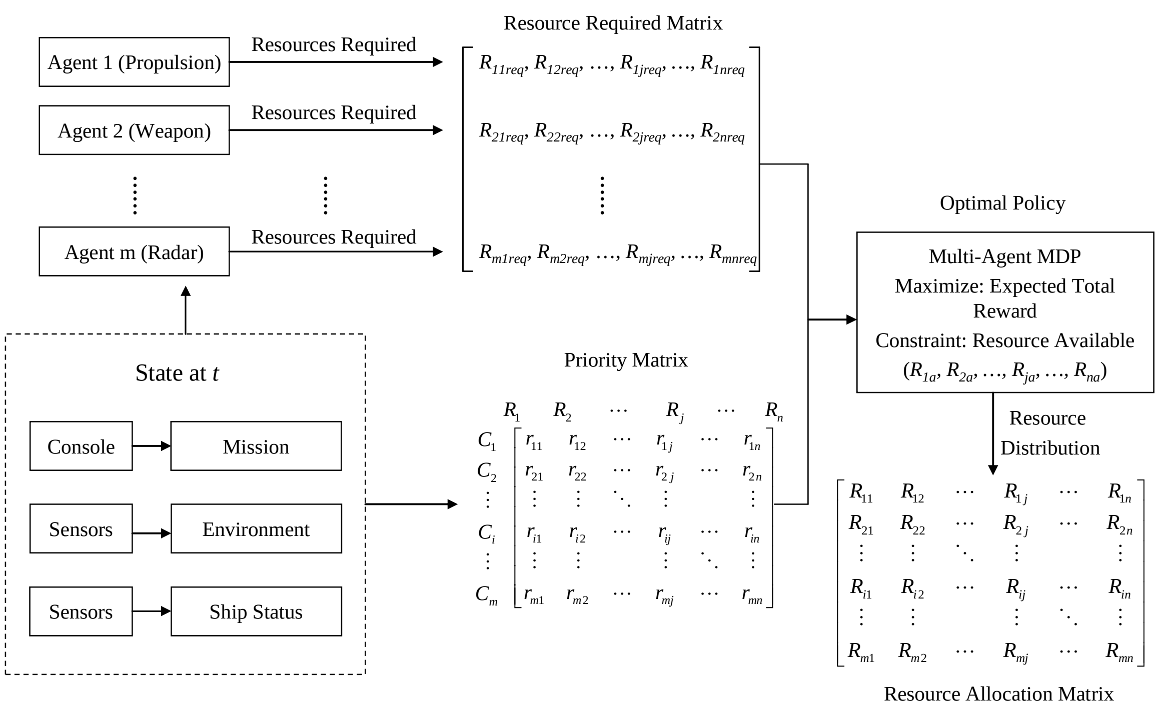


Figure 68: Resource Allocation Process

formulation to produce the optimal policy. The available resources impose the constraints on the MDP process when the optimal policy is calculated. Finally the resources will be distributed by supplying the required resource to the agents which perform corresponding actions determined by the optimal policy.

The resource allocation process can be detailed described as the step by step procedure below:

Step 1: Identify state and action spaces for each agent

Assume the system consists of M agents which operate independently. The state space S^m ($m = \{1, 2, \cdots, M\}$) and action spaces A^m ($m = \{1, 2, \cdots, M\}$) of the agents need to be identified. A state of an agent is defined by one or more state variables. In each state of agent m, there is a set of action A_i^m can be taken, and all A_i^m compose the action space A^m.

Step 2: Estimate transition probability function and define immediate rewards

For each agent m, the transition probability matrix $P^m = [p_{iaj}^m]$ ($m = \{1, 2, \cdots, M\}$) needs to be estimated. The transition probabilities are often estimated using the historical data or calculated based on the simulation results. The immediate reward r_{ia}^m of the agent for each state-action pair needs to be defined by decision maker based on the expected effect of the action.

Step 3: Identify the resource type, upper bound of each resource and resource required by each agent for each state-action pair

The resources required to carry out the actions should be identified and their types (reusable, consumable or recyclable) need to be recognized. The upper bounds of the resources are required to be known. In addition, the resources required by each agent for each state-action pair need to be identified.

<u>Step 4:</u> Find optimal policy

With the inputs well defined in step 1 to step 3, a constrained multi-agent Markov decision process can be formulated utilizing Equation (58). The optimal policy will be obtained by solving the equation employing the linear programming or mixed integer linear program technique.

<u>Step 5:</u> Resource allocation

After the optimal policy is produced, the resource allocation problem can be fulfilled in the system operation process. At a decision epoch, the optimal policy specifies which action should be taken in the current state, then the resources required to carry out the actions will be distributed to the agent to complete the resource allocation task.

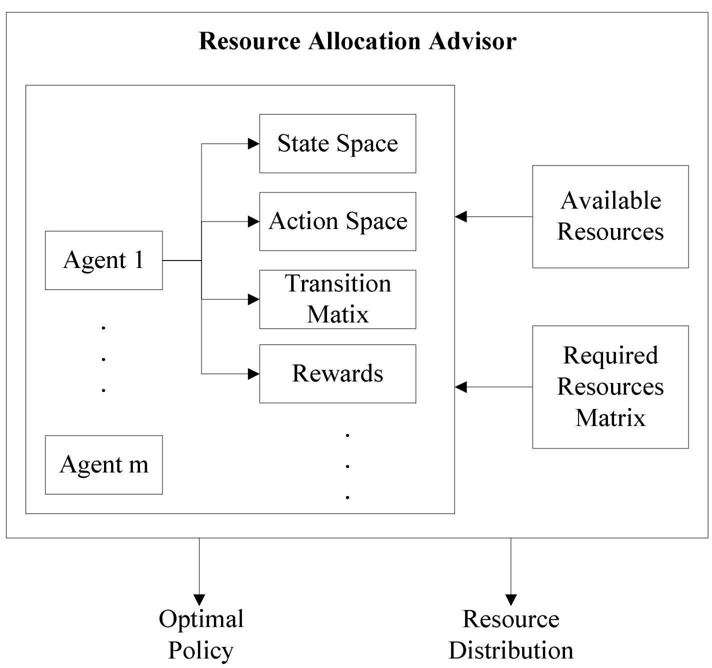


Figure 69: Resource Allocation Advisor

In the ship operation, in order to increase mission effectiveness and reduce cost, autonomous decisions need to be made. Thus, the decision making associated with resource allocation requires automation. A decision making advisor, shown in Figure 69, is proposed to realize the autonomous resource allocation. This advisor encompasses a constrained multi-agent MDP formulation which can generate the optimal policy used to allocate the resource.

It can be seen from Figure 69 that with all the inputs available, the advisor automates the step 4 and step 5 of the resource allocation process. In the system operation, some event may happen, such as damage occurrence or mission change. In this case, the associated inputs of the resource allocation advisor should be updated, and then the new optimal policy is calculated to direct the resource allocation process. This is illustrated in Figure 70.

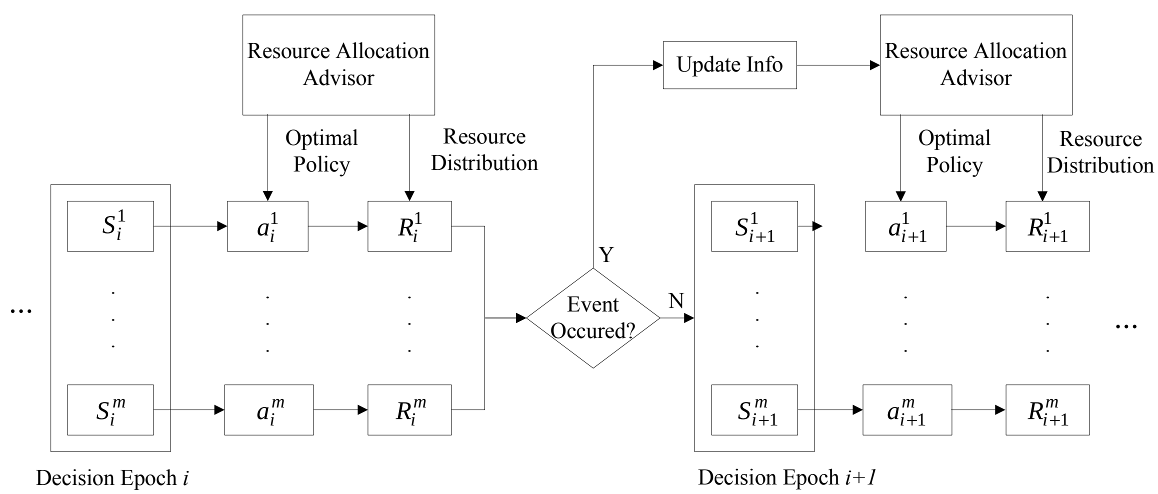


Figure 70: Resource Allocation When Event Occurs

6.3 *Implementation of Resource Allocation Formulation*

The IRIS framework provides a concept that integrates different ship systems to monitor and assess the ship state and then reacts to the current state by reconfiguring the ship to a new state which can best handle the situation at hand. Obviously, the ultimate objective of IRIS concept is to enable the ship to make autonomous decisions for determining the best action in each state to effectively perform the desired mission. To accomplish this objective, the problem can be modeled as a multi-agent Markov decision process and an optimal policy can be obtained to identify the best course of action. A resource allocation advisor is proposed to make autonomous decisions for the resource allocation process. To demonstrate the autonomous decision making and reconfiguration capabilities of the advisor, a resource allocation problem for the Chilled Water Reduced Scale Advanced Demonstrator (CW-RSAD) is chosen as a proof of concept.

6.3.1 CW-RSAD Model

In order to maximize the ship's performance, resources are required to be rapidly and effectively allocated to the subsystems. In addition, since an IRIS designed ship is envisioned to be able to reconfigure itself into a new state most suitable for the current situation, resources must be redistributed to support the reconfiguration. Therefore, a resource allocation problem needs to be solved in order to achieve the capability of reconfiguration.

The CW-RSAD is a reduced-scale model of two zones of the Arleigh Burke chilled water system (Figure 71) and is located at the Naval Surface Warfare Center in Philadelphia. The RSAD was originally constructed to investigate the component level intelligent distribution control system which is employed to achieve reliable unmanned control of shipboard auxiliary systems. It consist of 4 pumps, 2 chiller plants, and 16 service loads which are the units of equipment cooled by the chilled water system [Scheidt, 2002]. It also contains 2 expansion tanks with the capacity to deliver 40 gpm of chiller water. The RSAD utilize a vertically offset main loop to distribute chilled water to the 16 service loads [Fairmount Automation, 2006].

Figure 71: Chilled Water Reduced Scale Advanced Demonstrator [Scheidt, 2002]

In order to maintain their functions, the 16 service loads require cooling by the chilled water system to prevent them from being damaged due to overheated. Since the RSAD has limited capacity to provide the chilled water, usually not all of the service loads can obtain the required cooling resource. Therefore, the chilled water needs to be effectively distributed to the system and the distribution should maximize the performance of the RSAD, that is, the limited chilled water should be best used so that the utility of the system is maximized.

The service load can be in several states such as “overheated” or “working properly”, and actions associated with chilled water distribution will be taken in each state depending on the expected value of executing this action. This resource allocation problem is explored using the multi-agent Markov decision process resource allocation formulation described in Section 6.2.5.

6.3.2 Resource Allocation for CW-RSAD Model

To formulate the resource allocation problem using the multi-agent Markov decision process, a set of state, a set of action, transition probability function and immediate rewards of state-action pairs should be defined for each agent. And with this information optimal policy will be calculated and used to control the resource allocation.

6.3.2.1 Step 1: Identify state and action space for each agent

Agents

Obtained from Naval Surface Warfare Center, a FlowMaster model of RSAD is illustrated in Figure 72. The electrical architecture was developed to match the 16 service loads present in the RSAD. Each service load is considered as an agent and assumed to operate independently.

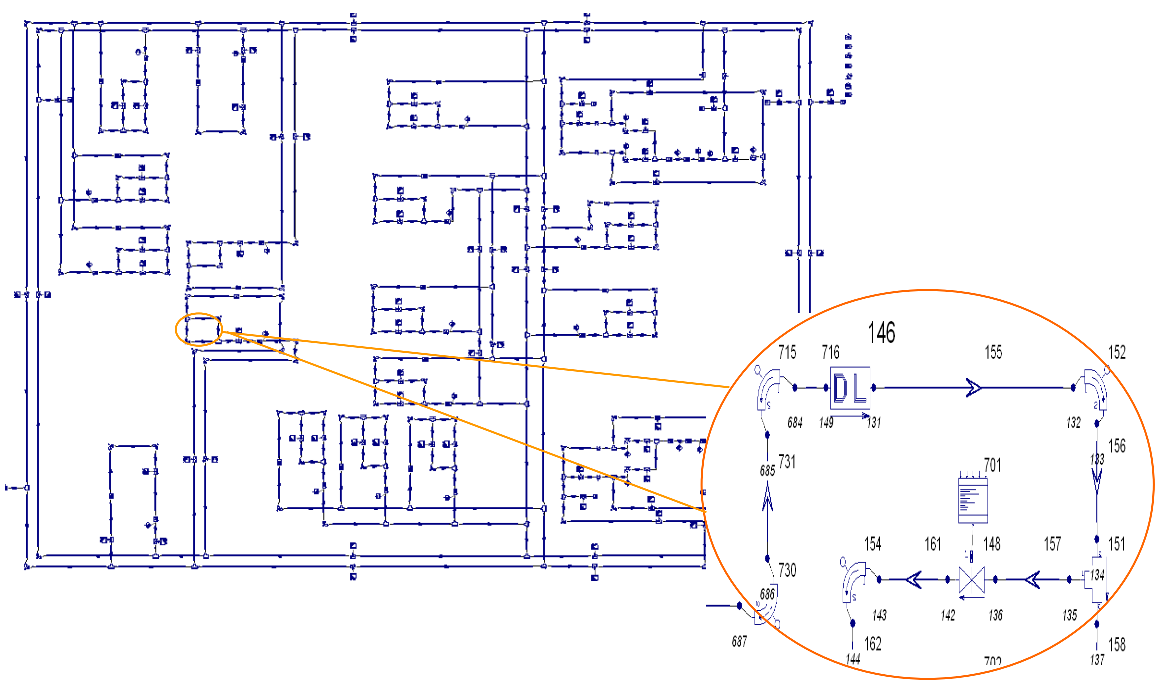


Figure 72: FlowMaster RSAD Model

Each service load represents a physical system, and the mapping between the service load and the physical systems are identified and listed in Table 17.

Table 17: Physical System of RSAD Model

Agent	Notation	Service Load	Modeled System
1	SVC01	AN/SLQ 32 Heat Exchanger	AN/SPY-1 Radar and Sonar System
2	SVC02	Aft Stbd Array Rm	Aft Starboard Array Room
3	SVC03	Director Eqpt Rm 1	Director System 1
4	SVC05	Aft Port Array Rm	Aft Port Array Room
5	SVC06	Fwd IC/Gyro	Forward IC/Gyro System
6	SVC08	Director Eqpt Rm 2	Director System 2
7	SVC10	Fwd Stbd Array Rm	Forward Starboard Array Room
8	SVC11	5"54 Gun Elex	Gun Weapon System
9	SVC12	HVAC CIC No.1	Combat Information Center 1
10	SVC13	HVAC CIC No.2	Combat Information Center 2
11	SVC14	HVAC CIWS wrkshp No. 1	Close-In Weapon System 1
12	SVC15	Fwd Port Array Rm	Forward Port Array Room
13	SVC16	HVAC Crew Living Space No. 2	Crew Living Space 2
14	SVC22S	C&D Heat Exchanger	C&D WTR CLR
15	SVC22P	C&D Heat Exchanger	C&D WTR CLR
16	SVC23	HVAC Crew/CPO Galley	Crew/CPO Galley Space

For simplicity and without loss of generality, the state space, action space, transition probability matrix and immediate rewards of all agents are assumed to be equal.

State Space

The states of each agent are described by the combination of two state variables, one representing the status of the agent and the other representing the priority assessment of the agent. The possible states of the agent are listed in Table 18.

The first state variable is used to describe the state of the agent itself. This state variable has three values: overheated, working properly and off. When the agent's temperature is higher than the threshold, it is considered "overheated". An agent is defined as “working properly” if it is working and its temperature is below the threshold. “Off” is a state that transits from a previous state. For example, if an agent is overheated, it may be turned off to prevent from being damaged, or if an agent is working properly but has low priority, it may be turned off to save the resources. In these cases, the agent state becomes “off”.

Table 18: State Space of Agent

State *i*	Description		
1	Overheated	&	High Priority
2	Overheated	&	Mid Priority
3	Overheated	&	Low Priority
4	Working Properly	&	High Priority
5	Working Properly	&	Mid Priority
6	Working Properly	&	Low Priority
7	Off	&	High Priority
8	Off	&	Mid Priority
9	Off	&	Low Priority

The other state variable is defined as priority which is a measure of emergency of an agent. The priority is assessed based on the states of mission being performed, the operational environment and agent status.

The mission being performed has a main contribution to the priority since the agents’ priorities vary significantly with the mission. A different mission requires different emphasis on certain functions, therefore, the agents which provide the required functions will have high priorities. For example, in a battle mission, in order to successfully accomplish the mission, weapon and radar systems should maintain proper functionalities and thus they have high priorities.

The operational environment, representing the surroundings of ship system, also affects the priorities of the agents. Since the same mission may be performed in different environments, the priorities of the agents may change with the environment. Under a cruise mission, for example, the propulsion system often has the highest priority if there is no enemy around. However, when the ship is in a hostile environment, the weapons system may have a higher priority than the propulsion system.

It is clear that the priority of an agent depends on its own status. For example, if an agent is turned off, its priority is certainly low (i.e. it is not going to be used and no resource will be provided to it). Or if an agent is overheating, it mostly has high priority to get the required resources.

Therefore, the overall priorities of the agents are determined by the combination of mission, environment and status, given by Equation (59).

$$pr = \sum_{i=1}^{3} w_i * pr_i \tag{59}$$

where $pr_i = (pr_{i,1}, pr_{i,2}, \cdots, pr_{i,16}), i = 1, 2, 3$ is the priority vector contributed by mission, environment and status respectively. w_i, $i = 1, 2, 3$ is the corresponding relative importance of the three contributors.

As mentioned before, state variables can be obtained from the console, sensors or other agents which are able to directly provide the variables or supply the information that can be used to derive the values of the variables. A model was constructed to simulate the resource allocation process. This model is part of the integrated simulation environment developed by the IRIS team in Aerospace Systems Design Laboratory (ASDL) at Georgia Institute of Technology. The environment provides designers with an integrated modeling and simulation environment to evaluate design information using Model Center, as shown in Figure 73. It integrates the models of a simplified electrical power distribution network with a chilled water system and a hierarchical control system. The integrated design environment enables the fast execution of each model and can track the interface variables of the models. The models developed for this environment are based on the RSAD FlowMaster model as shown in Figure 72.

The model labeled as HLCtrl is a top level control system that assesses the events from a system point of view and makes autonomous decisions on what plans/actions should be performed to reallocate the available resources to the agents in order to reconfigure the system into the state which is adequate suitable for the mission, operational environment and agent status. Thus, the resource allocation task is accomplished by this model, as shown in Figure 74. From this figure, it can be seen that the state variables can be obtained from the other models: the first state variable can be obtained from the FlowMaster model and electrical model (labeled as ChilledWater and PowerModel in Figure 73, respectively) where the temperature of the service load can be calculated. The priority of each service load can be computed using the information from agent based control system, Human Machine Interface (HMI) model and external inputs. The agent based control system, labeled as ABCtrl in Figure 73, can indicate the status of the service load and send the information to the top level control system. The HMI model labeled as HMI serves as an interface that allows human operator to supervise the performance of the system and send the mission requirements to high level control system. There is no model that simulates the operational environment, thus the priority contributed by environment is modeled using the predefined data that are directly input to the resource allocation model. Therefore, with the information about the mission, environment and status, the overall priority can be calculated using Equation (59).

After the state variables are obtained, the resource allocation model will formulate a multi-agent Markov decision process and then find an optimal policy to allocate the resources to the service loads.

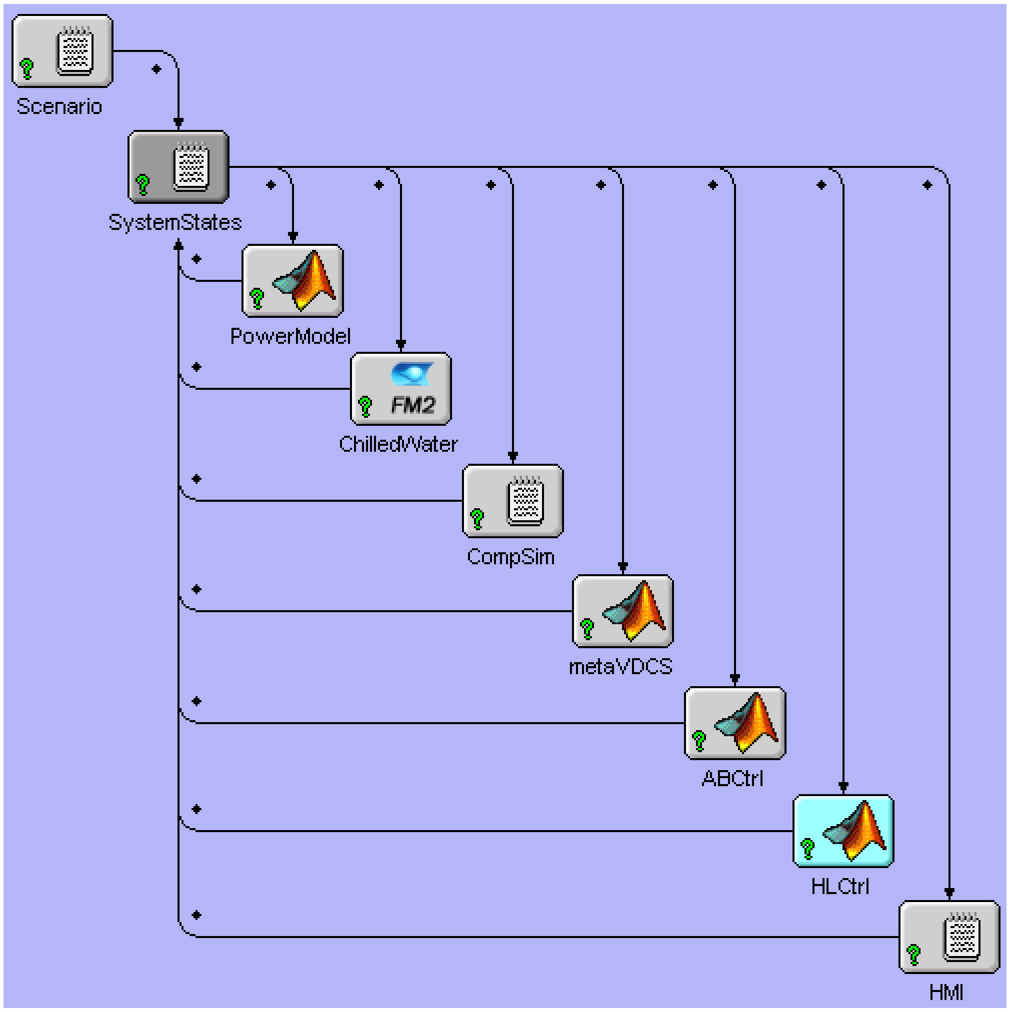


Figure 73: IRIS Integrated Environment

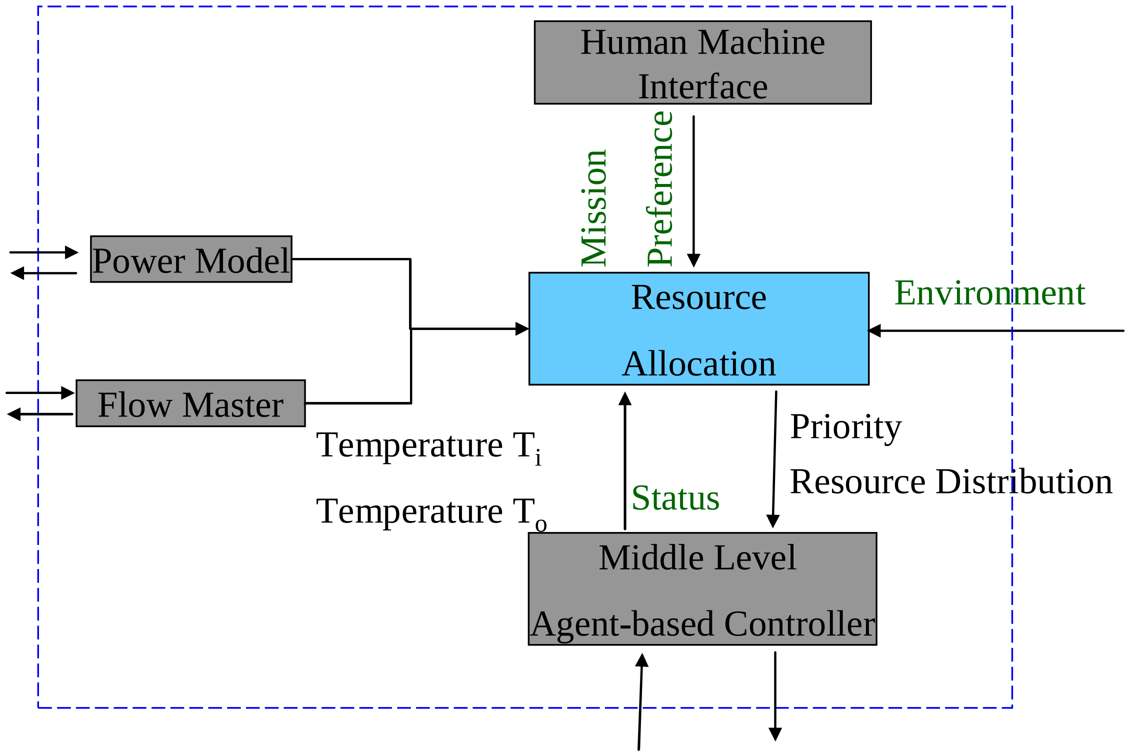


Figure 74: Resource Allocation Model for RSAD

Action Space

Three actions can be taken for each agent depending on its state. The actions are: supply agent the required cooling fluid, turn the agent off, turn the agent on and supply the required cooling fluid, as listed in Table 19.

Table 19: Action space

Action a_i	Description
a_1	Supply agent the required cooling fluid
a_2	Turn the agent off
a_3	Turn the agent on and supply the required cooling fluid

As stated in Table 18, each agent may be in one of 9 states at a given time. Notice that not all actions can be performed in each state since some actions are not appropriate to be taken in certain states. Action a_1 indicates that the required cooling fluid (chilled water) will be supplied to an agent, thus, this action can be performed in all states. Action a_2 can be taken in all the states except for the states that the agent's state is already "off" (i.e states 7, 8, 9) while a_3 can only be performed in such states. Table 19 lists the action spaces A_i for each state.

Table 20: Action Space for Each State

State i	Action Space A_i
1	$\{a_1, a_2\}$
2	$\{a_1, a_2\}$
3	$\{a_1, a_2\}$
4	$\{a_1, a_2\}$
5	$\{a_1, a_2\}$
6	$\{a_1, a_2\}$
7	$\{a_1, a_3\}$
8	$\{a_1, a_3\}$
9	$\{a_1, a_3\}$

6.3.2.2 Step 2: Estimate transition matrix and define immediate rewards

The transition probability matrix $P = [p_{iaj}]$ represents the probabilities of changing to state j if action a is executed in state i. It is clear that P is a $|S| \times |A| \times |S|$ matrix. The immediate return $R = [r_{ia}]$ defines the expected immediate reward by executing action a in state i. The transition matrix and expected immediate rewards of the three actions for RSAD model are given by Table 21, Table 22 and Table 23, respectively. In this study, without loss of generality, it is assumed that the transition probability matrixes and the corresponding rewards of the 16 service loads are the same.

Since the goal of the system operation is to work on the best course of action to gain the maximum desirability of its potential effect, the best course of action needs to be identified first. The optimal policy defines what action is the best to be taken in a system state, thus by following the optimal policy one can obtain the best course of action. The action taken in a state is considered as the “best” action because its potential effect is expected to best achieve the objective of the operation, that is, effectiveness. Therefore, the total rewards obtained by executing the best course of action represent the system’s effectiveness. In other words, it can be stated that reward earned by the execution of an action for a state-action pair represents its potential effectiveness.

Table 21: Transition Probability Matrix and Rewards for Action a_1

State i	p_{i11}	p_{i12}	p_{i13}	p_{i14}	p_{i15}	p_{i16}	p_{i17}	p_{i18}	p_{i19}	r_{i1}
1	0.1	0.07	0.03	0.55	0.15	0.1	0	0	0	20
2	0.08	0.1	0.02	0.1	0.5	0.2	0	0	0	10
3	0.01	0.01	0.08	0.05	0.1	0.75	0	0	0	-5
4	0.05	0.03	0.02	0.6	0.2	0.1	0	0	0	15
5	0.02	0.05	0.03	0.03	0.7	0.17	0	0	0	10
6	0.01	0.04	0.05	0.1	0.15	0.65	0	0	0	5
7	0	0	0	0	0	0	1	0	0	-5
8	0	0	0	0	0	0	0	1	0	-2
9	0	0	0	0	0	0	0	0	1	3

Table 22: Transition Probability Matrix and Rewards for Action a_2

State i	p_{i21}	p_{i22}	p_{i23}	p_{i24}	p_{i25}	p_{i26}	p_{i27}	p_{i28}	p_{i29}	r_{i2}
1	0	0	0	0	0	0	0.7	0.2	0.1	-10
2	0	0	0	0	0	0	0.2	0.7	0.1	-5
3	0	0	0	0	0	0	0.1	0.2	0.7	5
4	0	0	0	0	0	0	0.8	0.15	0.05	-20
5	0	0	0	0	0	0	0.6	0.3	0.1	-10
6	0	0	0	0	0	0	0.4	0.4	0.2	-5
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0

Table 23: Transition Probability Matrix and Rewards for Action a_3

State i	p_{i31}	p_{i32}	p_{i33}	p_{i34}	p_{i35}	p_{i36}	p_{i37}	p_{i38}	p_{i39}	r_{i3}
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0.7	0.2	0.1	0	0	0	15
8	0	0	0	0.1	0.6	0.3	0	0	0	10
9	0	0	0	0.05	0.15	0.8	0	0	0	-5

6.3.2.3 Step 3: Available Resource and Required Resource

In this example, the resource that needs to be allocated is chilled water which is a recyclable resource. As mentioned in Section 6.3.1, RSAD has the capacity to deliver 40 gpm chilled water, therefore, this is the cooling resource available for all the 16 service loads.

The resource required by each agent depends on their current state and action executed in this state. If action a_1 is performed in a state, the resource required by that state will be supplied to the agent. Typically, the resource required by the agent in the overheated state is greater than in the state of working properly, and the required resource is zero if an agent is in off state. If action a_2 is performed in any state, the resource required is zero since the agent is turned off and will not consume any resource. Action a_3 can only be taken in state 7, 8, and 9, and if it is executed the resource required by the agents in these states should equal the required resource that ensures they work properly. Table 24 lists the resource required to execute different actions in each state. The element g_{ia} $(i = 1,2,3,\cdots,9; a = 1,2,3)$ in the table defines the resource consumed by taking action a in state i. Without loss of generality, the 16 agents are assumed to have the same resource consumption for each state-action pair. From this table, one can see that when all agents work properly and are supplied the required cooling resources (i.e. the agents are in state 4, 5 or 6, and action a_1 is taken), the total required resource equals the total available resource.

Table 24: Resource Required by Each State-Action Pair

State i	g_{i1}	g_{i2}	g_{i3}
1	3	0	0
2	3	0	0
3	3	0	0
4	2.5	0	0
5	2.5	0	0
6	2.5	0	0
7	0	0	2.5
8	0	0	2.5
9	0	0	2.5

6.3.2.4 Step 4: Find optimal policy

In RSAD model, the only resource needs to be allocated is cooling fluid – chilled water which is a recyclable resource that can be reused by being chilled by the chiller of chilled water system. Therefore Equation (58) can be reduced to Equation (60) which can be solved by utilizing the linear programming technique.

$$\begin{aligned}
&\max \sum_m \sum_i \sum_a x_{ia}^m r_{ia}^m \\
&s.t \quad \sum_i \sum_a (\delta_{ij} - p_{iaj}^m) x_{ia}^m = \alpha_j^m \\
&\qquad \sum_m \sum_i \sum_a x_{ia}^m (g_{iaw}^m - \lambda_m \hat{g}_w) \le 0 \\
&\qquad x_{ia}^m \ge 0
\end{aligned} \tag{60}$$

The initial condition α_j^m ($j = 1,2,\cdots,9$; $m = 1,2,\cdots,16$) is given as 1 indicating that the number of the times that the agent *m* starts in each state *j*. At this point, all the necessary information required to compute the optimal policy is obtained. By using the linear programming technique, the optimal policy of the agent is calculated and shown in Table 25. Notice that since the transition probability matrix, immediate reward and resource required by all agents are assumed the same, the optimal policies for the agents are also the same.

Table 25: Optimal Policy

State i	π_{i1}	π_{i2}	π_{i3}
1	0.96	0.04	0
2	0.69	0.31	0
3	0.58	0.42	0
4	0.80	0.20	0
5	0.96	0.04	0
6	0.94	0.06	0
7	0.31	0	0.69
8	0.44	0	0.56
9	0.56	0	0.44

The optimal policy shown in Table 25 is a randomized policy. The element π_{ia} in this table represents the probability of taking action a in state i. When the policy is executed, an action will be chosen based on the probability distribution over the state space which is defined by the optimal policy. The optimal policy presented in Table 25 provides some insights about the best action to be taken in each state. It can be seen that if an agent is in any state of 1 to 6 the probability of being supplied the required cooling resource is much higher than the probability of turning the agent off. In addition, if an agent is in off state (i.e. state 7, 8 or 9) the execution of the optimal policy will tend to turn the agent on and supplying it the required resource except it is in state 9. It can be explained as state 9 has low priority so keeping it off can save some resource that could be used by the high priority states.

<u>Step 5: Resource allocation Process</u>

In the system operation, at time t, also considered as a decision epoch, an agent is in the state i. An action a is chosen from a set of allowable actions and then executed with the objective of maximizing the expected total reward. The proper action to take can be identified by following the optimal policy as shown in Table 25. That is, in a certain state which action is selected to be executed is determined by its probability over this state. For example, if an agent is in state 1, action a_1 has a probability of 96% to be executed while action a_2 has a probability of 4% to be taken. After the action is performed in the state, the agent will change to a state with a probability defined by Table 21, Table 22 or Table 23 based on the selected state-action pair, and at the same time the agent receives a reward. In this example, if action a_2 is selected, the agent will transit to state 7, 8 or 9 with probability of 70%, 20%, 10% respectively, and meanwhile receive a reward of -10. In addition, the execution of action a_2 in state 1 consumes 0 unit of resource, which can be found from Table 24. At next decision epoch, the agent will go through the same process and then move to another new state. This stochastic process is illustrated in Figure 75. Notice that at each decision epoch, all the 16 agents need to take one action based on the optimal policy, and the executions of all the actions should not overuse the total available resource.

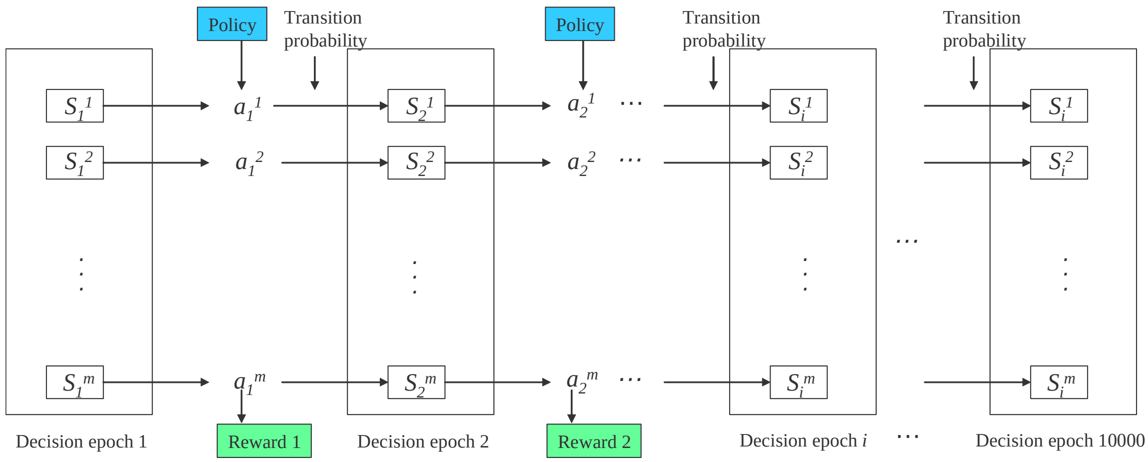


Figure 75: Action Selection and Resource Allocation Process

6.3.3 Simulation Studies

The objective of the simulation study is to investigate the effects of the course of action defined by the optimal policy and gain insights into its performance. Since the optimal policy is the solution to the constrained multi-agent MDP problem, eventually, the constrained multi-agent MDP formulation will be examined. This formulation is encompassed in the resource allocation model, labeled as HLCtrl in Figure 73. Therefore, the resource allocation model will be tested in the simulation study process. To test the resource allocation model, instead of using the integrated simulation environment presented in Figure 73, a stand-alone MATLAB program is used to perform the simulation. In a simulation, at each decision epoch the optimal policy calculated in Section 6.3.2.4 determines an action to take based upon the probabilities of the actions over the state. After an action is taken in the current state, the system earns a reward and then transits to a new state depending on the transition probabilities defined in Table 21, Table 22 and Table 23. This process is repeated at each decision epoch until it reaches the maximum number of the decision epoch. Thus, the optimal policy can be investigated using this simulation program without needing to invoke any other model of the integrated simulation environment.

To explore the performance of the optimal policy, four other policies, as given by Table 26, are constructed to compare with the it. The policy given by Table 26 (a) is a deterministic policy which always chooses the action that will maximize the immediate reward in each state. The other three randomized policies are arbitrary, valid policy.

Table 26: Four Policies

(a) Maximum Immediate Reward Policy

State i	π_{i1}	π_{i2}	π_{i3}
1	1	0	0
2	1	0	0
3	0	1	0
4	1	0	0
5	1	0	0
6	1	0	0
7	0	0	1
8	0	0	1
9	1	0	0

(b) Arbitrary Policy 1

State i	π_{i1}	π_{i2}	π_{i3}
1	0.8	0.2	0
2	0.7	0.3	0
3	0.4	0.6	0
4	0.9	0.1	0
5	0.6	0.4	0
6	0.4	0.6	0
7	0.4	0	0.6
8	0.5	0	0.5
9	0.8	0	0.2

(c) Arbitrary Policy 2

State i	π_{i1}	π_{i2}	π_{i3}
1	0.4	0.6	0
2	0.1	0.9	0
3	0.7	0.3	0
4	0.5	0.5	0
5	0.3	0.7	0
6	0.8	0.2	0
7	0.5	0	0.5
8	0.1	0	0.9
9	0.4	0	0.6

(d) Arbitrary Policy 3

State i	π_{i1}	π_{i2}	π_{i3}
1	0.9	0.1	0
2	0.8	0.2	0
3	0.6	0.4	0
4	0.2	0.8	0
5	0.3	0.7	0
6	0.4	0.6	0
7	0.7	0	0.3
8	0.9	0	0.1
9	0.2	0	0.8

6.3.3.1 Policy Comparison

Average Reward for One Simulation

The comparison starts by running a simulation for each of the policies. In the simulation, the 16 agents begin with an initial state and go through 10000 decision epochs by following the policies. Since the 16 agents are assumed to operate independently, their initial states are also independent. The initial states of the 16 agents compose the initial state of the simulation which is shown in Table 27.

Table 27: Starting States for Policy Simulation

Agent	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
State	3	8	7	8	3	6	3	3	4	7	9	3	4	7	8	4

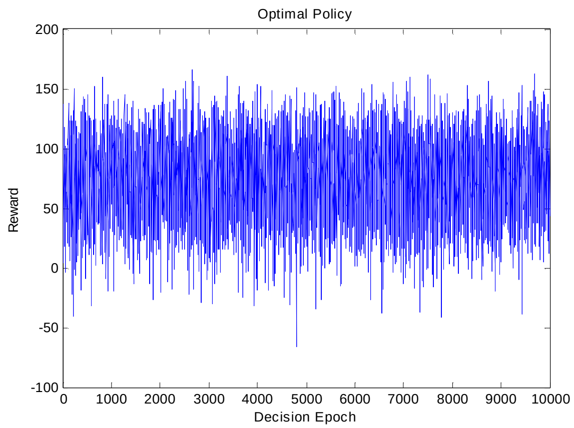
In the simulation, a policy is utilized to make the decision and select the action in each state. The operation of the system starts from the initial state $S_0 = \{s_0^1, s_0^2, \cdots, s_0^{16}\}$ (where $s_0^m = \{1,2,\cdots,9\}, m = 1,2,\cdots,16$) at decision epoch t_0. At this epoch, decisions need to be made upon choosing one action a_0^m (where $a_0^m = \{1,2,3\}$, $m = 1,2,\cdots,16$) for each agent in its initial state based on the probability $\pi_{ia,0}^m$ (where $i = 1,2,\cdots,9; a = 1,2,3;\ m = 1,2,\cdots,16$) defined by the policy. The effect of the action will result in a transition to a new state s_1^m (where $m = 1,2,\cdots,16$) and a reward $r_{ia,0}^m$ (where $i = 1,2,\cdots,9;\quad a = 1,2,3;\quad m = 1,2,\cdots,16$) is earned by the agent. The transition is manipulated by the transition probability matrix shown by Table 21, Table 22, or Table 23 depending on what the state action pair is. Consequently, the system enters to the next decision epoch t_1 and the same process at decision epoch t_0 will be repeated. The system then moves to the next decision epoch until the maximum number of the decision epoch (i.e. 10000) is reached. This process can be clearly viewed in Figure 75.

The rewards gained by the agents at each decision epoch are calculated for each policy. Figure 76 shows the reward trajectories for the five policies in a 10000 decision epoch simulation. The statistic results of average reward for the five policies are listed in Table 28.

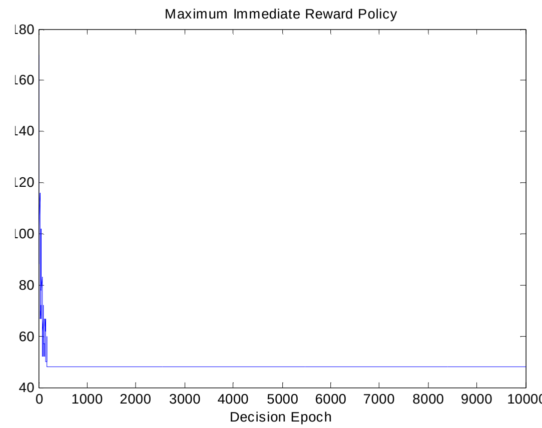
Table 28: Statistic Results of Average Reward for Five Policies

Policy	μ	σ
Optimal	98.0275	30.6818
Max Immediate Reward	48.0439	2.8519
Arbitrary 1	60.5912	36.6398
Arbitrary 2	17.5475	45.3764
Arbitrary 3	-27.8321	34.6259

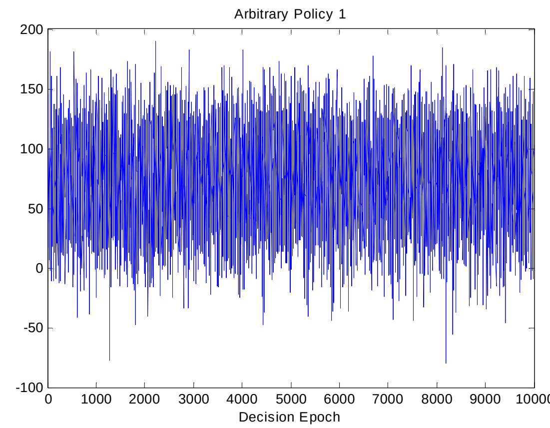
From Table 28 and Figure 76, one can see that maximum effectiveness (maximum average reward) is obtained when the system operates by following the optimal policy. This indicates that the best course of action is executed during the system operation process, and the optimal policy does have better performance than any other policy. The maximum immediate reward does not perform well. This implies that the decisions must not be made myopically, but must anticipate the opportunities and rewards associated with future system states. Different policies have different performance, and poor policy may lead to cost (negative reward) to the system and, in turn, makes the resource allocation ineffective. Notice that after around 100 decision epochs, the reward keeps as a constant for maximum immediate policy. This is due to the fact that once a state transits to state 9 it will stay in this state forever because action a_1, which generates the maximum immediate reward for state 9, will be taken in this state and the probability of changing back to state 9 is 1.



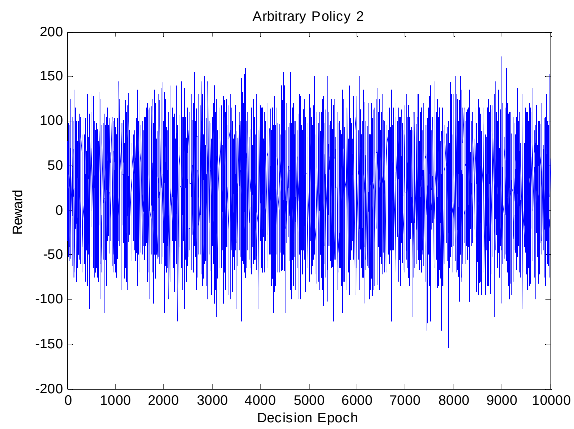
(a) Optimal Policy



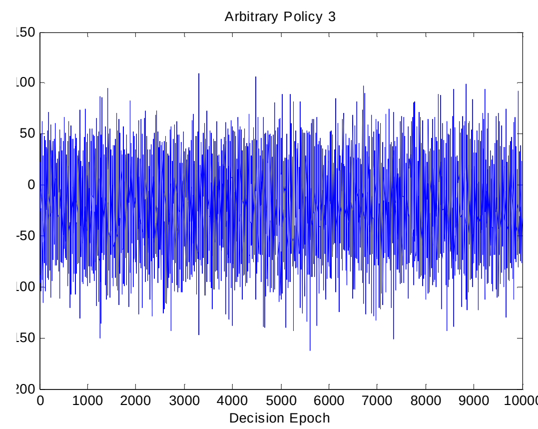
(b) Maximum Immediate Reward Policy



(c) Arbitrary Policy 1



(d) Arbitrary Policy 2



(e) Arbitrary Policy 3

Figure 76: Reward Trajectory for Five Policies

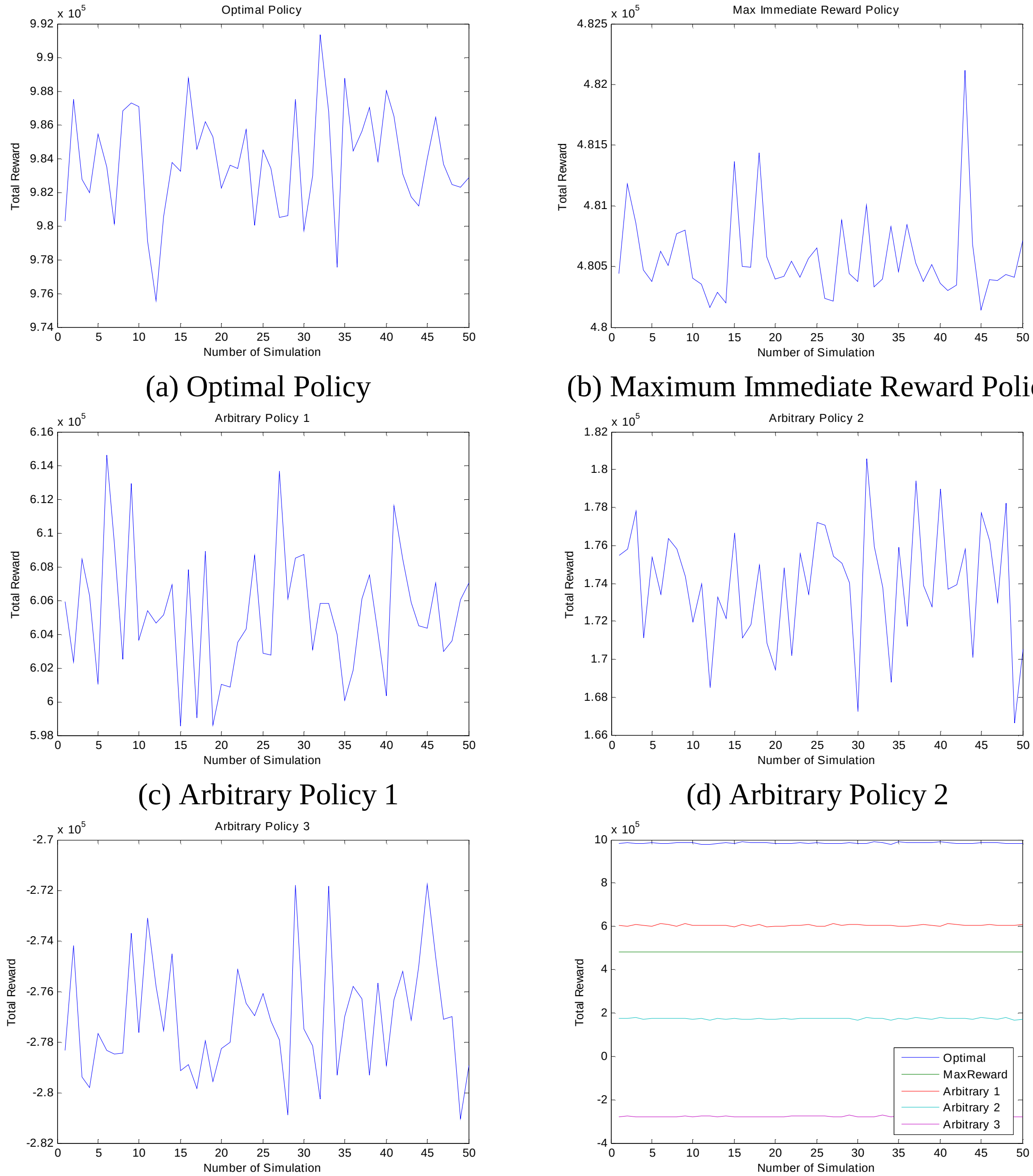
Total Rewards for 50 Simulations

To further investigate the performance of the policies, 50 simulations are performed for each policy. In each simulation, the 16 agents start from an initial state and go through 10000 decision epochs by following one of the policies. The initial state of each simulation for all the policies are the same so that the policies can be compared based on the same basis. The total reward in one simulation can be obtained by summing the reward at each decision epoch. The simulation runs 50 times for each policy and the total rewards of each run is computed. (a) to (e) of Figure 77 show the total rewards for the five policies in 50 simulations. Figure 77 (f) depicts the total rewards obtained by following each policy in the operation for 50 simulations. The statistic results of the total rewards for the five policies are listed in Table 29.

Table 29: Statistic Results of Total Rewards for Five Policies

	Optimal	Max Imme Rwd	Arbitrary 1	Arbitrary 2	Arbitrary 3
μ	9.84E+05	4.81E+05	6.05E+05	1.74E+05	-2.77E+05
σ	3.13E+03	358.623	3.68E+03	3.12E+03	2.28E+03

It can be seen from Figure 77 and Table 29 that the total rewards (effectiveness) vary with the simulations starting from different points. For different starting points, maximum total rewards are always obtained if the system operates under the optimal policy. This indicates that the optimal policy given by the MDP formulation does offer the best performance for system operation. Again, the maximum immediate reward policy is not a good policy, which implies that the actions with maximum immediate reward may produce side-effect on the future system states.



(a) Optimal Policy (b) Maximum Immediate Reward Policy

(c) Arbitrary Policy 1 (d) Arbitrary Policy 2

(e) Arbitrary Policy 3 (f) Total Reward Comparison

Figure 77: Total Rewards for Five Policies

6.3.3.2 Scenario Analysis

It is obvious that the optimal policy is determined by the transition probability and reward function. The reward function represents the decision maker's preference since the rewards for different state action pairs indicate their importance to the decision maker. Typically, decision maker's preference strongly depends on the priorities of the agents. In other words, if an agent is important for completing a certain mission, efforts will be made to maintain it in the state which is required to maximize the mission effectiveness. Therefore all the actions that try to make the agent transit to the desired state will earn high reward. To examine the different preference information, another three scenarios are analyzed. In these scenarios, the state space, action space, available resource, and required resource are the same as before, which are defined in Table 18, Table 19, and Table 24 respectively. The transition probability matrix and immediate reward function need to be updated to reflect the decision maker's preference information. Thus, new optimal policy will be calculated and the maximum immediate reward policy needs to be reconstructed based on the updated information.

Scenario 1 – Battle Mission

In this scenario, the system is envisioned to perform battle mission during the operation process. Therefore, the agents which have strong impact on this mission effectiveness will have high priority to obtain the required resource. These agents include weapon system, radar system, combat information system and so on. Figure 78 lists all the 16 agents which are classified in term of priority with respect to the battle mission. Since the agents have different priorities, their transition probability matrixes and reward functions also vary, however, the agents with same priority share the same transition probability matrixes and reward functions Table 30, Table 31 and Table 32 respectively show the transition probability and immediate reward of each state-action pair for the agents with different priorities.

The decision strategy in the battle mission is always to try to keep the agents with high priority working properly and not to turn them off. This implies that the action a_2 should not be taken when the agents are in the states 1-6, and action a_1 should not be taken in the states of 7-9 while a_3 should be performed in these states. In addition, since to keep these agents working is critical to the mission effectiveness, after some actions are performed these agents should still have high probability of transiting to the desired state (i.e. state 4). This preference information is reflected in the transition probability and immediate reward matrixes defined by Table 30. As we can see from this table that whatever the state-action pair is, the transition probabilities of state 4 are much higher than the other probabilities. Furthermore, rewards assigned to the state action pairs (i.e. action a_1 for states 1 – 6; action a_3 for states 7 – 9) whose effects are to keep the agents working and obtaining required resource are always positive, while the rewards given to the state action pairs that result in turning the agent off (i.e. action a_2 for states 1 – 6) or keeping the agent off (i.e. action a_1 for states 7 – 9) are negative.

To employ this decision strategy, the agents with high priority will have high probability of being in the desired state. This implies these agents will have high probabilities of working properly and obtaining the required resource. In this case, we can say that the system is reconfigured to the state most suitable to handle the situation at hand. To investigate the reconfiguration for this scenario, availability of each agent is calculated. Availability is defined as the proportion of time when an agent is in a functioning condition. A functioning condition is referred to as the states in which an agent is functional due to obtain the required resource. The availability for the 16 agents are calculated and listed in Table 33 to Table 37, and additional results can be found in Appendix D.

Agent	Notation	Service Load	Modeled System
1	SVC01	AN/SLQ 32 Heat Exchanger	AN/SPY-1 Radar and Sonar System
2	SVC02	Aft Stbd Array Rm	Aft Starboard Array Room
3	SVC03	Director Eqpt Rm 1	Director System 1
4	SVC05	Aft Port Array Rm	Aft Port Array Room
5	SVC06	Fwd IC/Gyro	Forward IC/Gyro System
6	SVC08	Director Eqpt Rm 2	Director System 2
7	SVC10	Fwd Stbd Array Rm	Forward Starboard Array Room
8	SVC11	5"54 Gun Elex	Gun Weapon System
9	SVC12	HVAC CIC No.1	Combat Information Center 1
10	SVC13	HVAC CIC No.2	Combat Information Center 2
11	SVC14	HVAC CIWS wrkshp No. 1	Close-In Weapon System 1
12	SVC15	Fwd Port Array Rm	Forward Port Array Room
13	SVC16	HVAC Crew Living Space No. 2	Crew Living Space 2
14	SVC22S	C&D Heat Exchanger	C&D WTR CLR
15	SVC22P	C&D Heat Exchanger	C&D WTR CLR
16	SVC23	HVAC Crew/CPO Galley	Crew/CPO Galley Space

High Priority   Mid Priority   Low Priority

Figure 78: Prioritized System in Battle Mission

Table 30: Transition Matrix and Reward Function for High Priority Agents in Scenario 1

State i	p_{i11}	p_{i12}	p_{i13}	p_{i14}	p_{i15}	p_{i16}	p_{i17}	p_{i18}	p_{i19}	r_{i1}
1	0.15	0.02	0.005	0.8	0.02	0.005	0	0	0	20
2	0.1	0.03	0.01	0.78	0.07	0.01	0	0	0	18
3	0.12	0.04	0.02	0.75	0.06	0.01	0	0	0	16
4	0.1	0.01	0.004	0.85	0.03	0.006	0	0	0	20
5	0.08	0.03	0.005	0.83	0.05	0.005	0	0	0	14
6	0.06	0.04	0.02	0.8	0.05	0.03	0	0	0	9
7	0	0	0	0	0	0	0.9	0.07	0.03	-18
8	0	0	0	0	0	0	0.8	0.1	0.1	-15
9	0	0	0	0	0	0	0.75	0.15	0.1	-12

State i	p_{i21}	p_{i22}	p_{i23}	p_{i24}	p_{i25}	p_{i26}	p_{i27}	p_{i28}	p_{i29}	r_{i2}
1	0	0	0	0	0	0	0.95	0.03	0.02	-20
2	0	0	0	0	0	0	0.85	0.09	0.06	-19
3	0	0	0	0	0	0	0.8	0.12	0.08	-18
4	0	0	0	0	0	0	0.97	0.02	0.01	-20
5	0	0	0	0	0	0	0.9	0.07	0.03	-19
6	0	0	0	0	0	0	0.85	0.1	0.05	-18
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0

State i	p_{i31}	p_{i32}	p_{i33}	p_{i34}	p_{i35}	p_{i36}	p_{i37}	p_{i38}	p_{i39}	r_{i3}
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0.95	0.03	0.02	0	0	0	20
8	0	0	0	0.9	0.07	0.03	0	0	0	18
9	0	0	0	0.85	0.09	0.06	0	0	0	16

Table 31: Transition Matrix and Reward Function for Mid Priority Agents in Scenario 1

State i	p_{i11}	p_{i12}	p_{i13}	p_{i14}	p_{i15}	p_{i16}	p_{i17}	p_{i18}	p_{i19}	r_{i1}
1	0.08	0.1	0.01	0.1	0.7	0.01	0	0	0	18
2	0.05	0.15	0.005	0.03	0.75	0.015	0	0	0	12
3	0.03	0.12	0.008	0.02	0.78	0.042	0	0	0	9
4	0.05	0.1	0.02	0.1	0.72	0.01	0	0	0	16
5	0.02	0.11	0.01	0.055	0.8	0.005	0	0	0	12
6	0.01	0.03	0.05	0.08	0.82	0.01	0	0	0	7
7	0	0	0	0	0	0	0.15	0.75	0.1	-16
8	0	0	0	0	0	0	0.08	0.85	0.07	-12
9	0	0	0	0	0	0	0.08	0.8	0.12	-10
State i	p_{i21}	p_{i22}	p_{i23}	p_{i24}	p_{i25}	p_{i26}	p_{i27}	p_{i28}	p_{i29}	r_{i2}
1	0	0	0	0	0	0	0.15	0.82	0.03	-18
2	0	0	0	0	0	0	0.1	0.88	0.02	-16
3	0	0	0	0	0	0	0.08	0.85	0.07	-12
4	0	0	0	0	0	0	0.15	0.83	0.02	-18
5	0	0	0	0	0	0	0.1	0.88	0.02	-16
6	0	0	0	0	0	0	0.08	0.86	0.06	-12
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
State i	p_{i31}	p_{i32}	p_{i33}	p_{i34}	p_{i35}	p_{i36}	p_{i37}	p_{i38}	p_{i39}	r_{i3}
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0.1	0.85	0.05	0	0	0	18
8	0	0	0	0.08	0.88	0.04	0	0	0	15
9	0	0	0	0.06	0.86	0.08	0	0	0	12

Table 32: Transition Matrix and Reward Function for Low Priority Agents in Scenario 1

State i	p_{i11}	p_{i12}	p_{i13}	p_{i14}	p_{i15}	p_{i16}	p_{i17}	p_{i18}	p_{i19}	r_{i1}
1	0.06	0.08	0.1	0.06	0.1	0.6	0	0	0	16
2	0.03	0.04	0.12	0.04	0.12	0.65	0	0	0	10
3	0.02	0.03	0.09	0.03	0.13	0.7	0	0	0	7
4	0.04	0.03	0.1	0.07	0.06	0.7	0	0	0	14
5	0.03	0.05	0.11	0.05	0.04	0.72	0	0	0	10
6	0.02	0.04	0.09	0.04	0.08	0.73	0	0	0	5
7	0	0	0	0	0	0	0.1	0.2	0.7	-14
8	0	0	0	0	0	0	0.08	0.19	0.73	-10
9	0	0	0	0	0	0	0.07	0.18	0.75	-8

State i	p_{i21}	p_{i22}	p_{i23}	p_{i24}	p_{i25}	p_{i26}	p_{i27}	p_{i28}	p_{i29}	r_{i2}
1	0	0	0	0	0	0	0.1	0.12	0.78	-16
2	0	0	0	0	0	0	0.08	0.11	0.81	-14
3	0	0	0	0	0	0	0.07	0.1	0.83	-10
4	0	0	0	0	0	0	0.12	0.06	0.82	-16
5	0	0	0	0	0	0	0.1	0.05	0.85	-14
6	0	0	0	0	0	0	0.08	0.04	0.88	-10
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0

State i	p_{i31}	p_{i32}	p_{i33}	p_{i34}	p_{i35}	p_{i36}	p_{i37}	p_{i38}	p_{i39}	r_{i3}
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0.08	0.1	0.82	0	0	0	16
8	0	0	0	0.06	0.09	0.85	0	0	0	12
9	0	0	0	0.05	0.08	0.87	0	0	0	10

Table 33: Agent Availability for Optimal Policy in Scenario 1

Agent	1	2	3	4	5	6	7	8
Availability	0.9999	0.9222	0.9225	0.9257	0.9235	0.9233	0.9227	0.9998
Agent	9	10	11	12	13	14	15	16
Availability	1	0.9997	0.9998	0.9251	0.8695	0.9998	1	0.8687

Table 34: Agent Availability for Maximum Immediate Policy in Scenario 1

Agent	1	2	3	4	5	6	7	8
Availability	0.9939	0.8123	0.8163	0.8142	0.8097	0.8141	0.8114	0.9942
Agent	9	10	11	12	13	14	15	16
Availability	0.9936	0.9945	0.9937	0.8115	0.6533	0.994	0.9939	0.6494

Table 35: Agent Availability for Arbitrary Policy 1 in Scenario 1

Agent	1	2	3	4	5	6	7	8
Availability	0.81	0.5376	0.5386	0.5378	0.5309	0.5296	0.5349	0.808
Agent	9	10	11	12	13	14	15	16
Availability	0.8079	0.8078	0.8052	0.5391	0.5292	0.808	0.808	0.5282

Table 36: Agent Availability for Arbitrary Policy 2 in Scenario 1

Agent	1	2	3	4	5	6	7	8
Availability	0.498	0.6215	0.6253	0.6246	0.623	0.6245	0.6245	0.4971
Agent	9	10	11	12	13	14	15	16
Availability	0.4989	0.5009	0.4964	0.6193	0.5908	0.499	0.4973	0.5905

Table 37: Agent Availability for Arbitrary Policy 3 in Scenario 1

Agent	1	2	3	4	5	6	7	8
Availability	0.2853	0.3098	0.309	0.3154	0.3096	0.3104	0.3124	0.2855
Agent	9	10	11	12	13	14	15	16
Availability	0.2826	0.2795	0.2837	0.3135	0.3619	0.2867	0.2867	0.3602

Scenario 2 – Cruise Mission

In this scenario, the system is envisioned to perform a cruise mission. Similar with scenario 1, the agents which highly affect the mission effectiveness are considered to have high priority. Figure 79 lists all the 16 agents which are classified in term of priority with respect to the cruise mission. In this scenario, agent 13 and 16 are considered to have high priority. The transition probability matrixes and rewards functions for this scenario are listed in Table 38, Table 39 and Table 40.

The decision strategy used in the cruise mission is similar to the one in the battle mission. That is, the agents with high priority should be kept in the desired state – the agents are in the "working" states with high priority because of obtaining the required resource. The availability of the 16 agents are calculated and listed in Table 41 to Table 45, and additional results can be found in Appendix D.

Agent	Notation	Service Load	Modeled System
1	SVC01	AN/SLQ 32 Heat Exchanger	AN/SPY-1 Radar and Sonar System
2	SVC02	Aft Stbd Array Rm	Aft Starboard Array Room
3	SVC03	Director Eqpt Rm 1	Director System 1
4	SVC05	Aft Port Array Rm	Aft Port Array Room
5	SVC06	Fwd IC/Gyro	Forward IC/Gyro System
6	SVC08	Director Eqpt Rm 2	Director System 2
7	SVC10	Fwd Stbd Array Rm	Forward Starboard Array Room
8	SVC11	5"54 Gun Elex	Gun Weapon System
9	SVC12	HVAC CIC No.1	Combat Information Center 1
10	SVC13	HVAC CIC No.2	Combat Information Center 2
11	SVC14	HVAC CIWS wrkshp No. 1	Close-In Weapon System 1
12	SVC15	Fwd Port Array Rm	Forward Port Array Room
13	SVC16	HVAC Crew Living Space No. 2	Crew Living Space 2
14	SVC22S	C&D Heat Exchanger	C&D WTR CLR
15	SVC22P	C&D Heat Exchanger	C&D WTR CLR
16	SVC23	HVAC Crew/CPO Galley	Crew/CPO Galley Space

High Priority Mid Priority Low Priority

Figure 79: Prioritized System in Cruise Mission

Table 38: Transition Matrix and Reward Function for High Priority Agents in Scenario 2

State i	p_{i11}	p_{i12}	p_{i13}	p_{i14}	p_{i15}	p_{i16}	p_{i17}	p_{i18}	p_{i19}	r_{i1}
1	0.6	0.02	0.005	0.35	0.02	0.005	0	0	0	20
2	0.1	0.73	0.01	0.08	0.075	0.005	0	0	0	18
3	0.12	0.04	0.72	0.05	0.065	0.005	0	0	0	16
4	0.1	0.01	0.004	0.85	0.03	0.006	0	0	0	16
5	0.08	0.03	0.005	0.03	0.85	0.005	0	0	0	14
6	0.16	0.14	0.02	0.6	0.075	0.005	0	0	0	10
7	0	0	0	0	0	0	0.86	0.03	0.11	-18
8	0	0	0	0	0	0	0.88	0.01	0.11	-16
9	0	0	0	0	0	0	0.749	0.15	0.101	-10

State i	p_{i21}	p_{i22}	p_{i23}	p_{i24}	p_{i25}	p_{i26}	p_{i27}	p_{i28}	p_{i29}	r_{i2}
1	0	0	0	0	0	0	0.95	0.04	0.01	-20
2	0	0	0	0	0	0	0.85	0.14	0.01	-18
3	0	0	0	0	0	0	0.8	0.19	0.01	-16
4	0	0	0	0	0	0	0.99	0.005	0.005	-18
5	0	0	0	0	0	0	0.98	0.015	0.005	-16
6	0	0	0	0	0	0	0.96	0.035	0.005	15
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0

State i	p_{i31}	p_{i32}	p_{i33}	p_{i34}	p_{i35}	p_{i36}	p_{i37}	p_{i38}	p_{i39}	r_{i3}
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0.98	0.015	0.005	0	0	0	18
8	0	0	0	0.97	0.025	0.005	0	0	0	15
9	0	0	0	0.95	0.04	0.01	0	0	0	-16

Table 39: Transition Matrix and Reward Function for Mid Priority Agents in Scenario 2

State *i*	p_{i11}	p_{i12}	p_{i13}	p_{i14}	p_{i15}	p_{i16}	p_{i17}	p_{i18}	p_{i19}	r_{i1}
1	0.18	0.2	0.01	0.1	0.5	0.01	0	0	0	18
2	0.15	0.15	0.1	0.03	0.55	0.02	0	0	0	12
3	0.13	0.12	0.1	0.02	0.58	0.05	0	0	0	9
4	0.15	0.2	0.02	0.1	0.52	0.01	0	0	0	16
5	0.02	0.11	0.01	0.06	0.6	0.2	0	0	0	12
6	0.01	0.03	0.05	0.08	0.32	0.51	0	0	0	-18
7	0	0	0	0	0	0	0.15	0.55	0.3	-16
8	0	0	0	0	0	0	0.18	0.35	0.47	-12
9	0	0	0	0	0	0	0.18	0.3	0.52	-8

State *i*	p_{i21}	p_{i22}	p_{i23}	p_{i24}	p_{i25}	p_{i26}	p_{i27}	p_{i28}	p_{i29}	r_{i2}
1	0	0	0	0	0	0	0.15	0.82	0.03	-18
2	0	0	0	0	0	0	0.1	0.88	0.02	-16
3	0	0	0	0	0	0	0.08	0.85	0.07	2
4	0	0	0	0	0	0	0.15	0.43	0.42	-18
5	0	0	0	0	0	0	0.1	0.28	0.62	-16
6	0	0	0	0	0	0	0.08	0.16	0.76	-15
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0

State *i*	p_{i31}	p_{i32}	p_{i33}	p_{i34}	p_{i35}	p_{i36}	p_{i37}	p_{i38}	p_{i39}	r_{i3}
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0.3	0.45	0.25	0	0	0	-10
8	0	0	0	0.28	0.38	0.34	0	0	0	-9
9	0	0	0	0.16	0.26	0.58	0	0	0	-10

Table 40: Transition Matrix and Reward Function for Low Priority Agents in Scenario 2

State i	p_{i11}	p_{i12}	p_{i13}	p_{i14}	p_{i15}	p_{i16}	p_{i17}	p_{i18}	p_{i19}	r_{i1}
1	0.16	0.18	0.155	0.005	0.1	0.4	0	0	0	16
2	0.13	0.24	0.155	0.005	0.22	0.25	0	0	0	14
3	0.12	0.25	0.19	0.01	0.23	0.2	0	0	0	-18
4	0.14	0.295	0.1	0.005	0.16	0.3	0	0	0	14
5	0.13	0.195	0.21	0.005	0.24	0.22	0	0	0	12
6	0.12	0.27	0.19	0.01	0.28	0.13	0	0	0	-18
7	0	0	0	0	0	0	0.3	0.4	0.3	-10
8	0	0	0	0	0	0	0.38	0.39	0.23	-12
9	0	0	0	0	0	0	0.37	0.38	0.25	-14

State i	p_{i21}	p_{i22}	p_{i23}	p_{i24}	p_{i25}	p_{i26}	p_{i27}	p_{i28}	p_{i29}	r_{i2}
1	0	0	0	0	0	0	0.2	0.62	0.18	-15
2	0	0	0	0	0	0	0.28	0.61	0.11	-14
3	0	0	0	0	0	0	0.27	0.6	0.13	-10
4	0	0	0	0	0	0	0.22	0.36	0.32	-14
5	0	0	0	0	0	0	0.2	0.25	0.55	-8
6	0	0	0	0	0	0	0.18	0.14	0.68	6
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0

State i	p_{i31}	p_{i32}	p_{i33}	p_{i34}	p_{i35}	p_{i36}	p_{i37}	p_{i38}	p_{i39}	r_{i3}
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0.28	0.4	0.32	0	0	0	-8
8	0	0	0	0.26	0.29	0.45	0	0	0	-5
9	0	0	0	0.15	0.28	0.57	0	0	0	-16

Table 41: Agent Availability for Optimal Policy in Scenario 2

Agent	1	2	3	4	5	6	7	8
Availability	0.9581	0.8996	0.8988	0.8975	0.8961	0.8993	0.8955	0.9587
Agent	9	10	11	12	13	14	15	16
Availability	0.9588	0.9572	0.9568	0.8983	0.9997	0.9578	0.9589	0.9969

Table 42: Agent Availability for Maximum Immediate Reward Policy in Scenario 2

Agent	1	2	3	4	5	6	7	8
Availability	0.7018	0.5811	0.5799	0.5832	0.5818	0.5829	0.581	0.7024
Agent	9	10	11	12	13	14	15	16
Availability	0.7039	0.7025	0.6976	0.579	0.9939	0.7017	0.7006	0.9948

Table 43: Agent Availability for Arbitrary Policy 1 in Scenario 2

Agent	1	2	3	4	5	6	7	8
Availability	0.4124	0.5021	0.5024	0.502	0.5034	0.5007	0.4997	0.4668
Agent	9	10	11	12	13	14	15	16
Availability	0.4697	0.4701	0.5354	0.5043	0.8024	0.4738	0.474	0.8046

Table 44: Agent Availability for Arbitrary Policy 2 in Scenario 2

Agent	1	2	3	4	5	6	7	8
Availability	0.5938	0.5855	0.5819	0.5802	0.5798	0.581	0.5805	0.5966
Agent	9	10	11	12	13	14	15	16
Availability	0.5974	0.5958	0.596	0.5805	0.5036	0.5953	0.5944	0.5031

Table 45: Agent Availability for Arbitrary Policy 3 in Scenario 2

Agent	1	2	3	4	5	6	7	8
Availability	0.4454	0.4204	0.4264	0.4255	0.4294	0.4234	0.4276	0.4399
Agent	9	10	11	12	13	14	15	16
Availability	0.4425	0.4427	0.4395	0.4304	0.304	0.4422	0.441	0.3057

From the results present in both scenarios, one can see that if the system operates under the optimal policy, it will obtain the maximum effectiveness over the other policies. In both scenarios, the agent availabilities highly depend on their priorities. In both scenarios, agents with high priorities always have much greater availabilities than the agents with low priorities if the system operates under the optimal policy. Poor policy does not guarantee the agents with high priority have high availabilities, which highly reduces the system effectiveness.

Scenario 3 – Events Occur

In some situations, events, such as damage occur, mission changes or environment varies, may happen in the system operation process. These events often significantly affect the ship state, which results in dramatic changes in the best course of action. Thus the control strategy needs to be modified to adapt this effect of the events. Since the states of the agents will vary dramatically, the same action taken in the same state of an agent will lead to different effects. Therefore, the transition probability matrixes need to be updated to reflect this effect. In addition, the DM's preference information will be different after the events occurred. For example, if damage happens to a high priority agent, the preference associated with this agent may change from supplying the agent required resource to turning the agent off in order to save the resource. Therefore, the reward function for each state-action pair needs to be updated too. After the transition probability matrices and reward functions are updated, they are input to the resource allocation advisor so that the new optimal policy will be calculated to identify the best course of action and reconfigure the system to the appropriate state. This process can be best interpreted by Figure 70.

In scenario 3, we consider a case that mission change happens at decision epoch 50. In detail, this case is described as: during the first 50 decision epochs, the system operates under cruise mission, and a hostile appears at decision epoch 50, so the system begins to operate under battle mission. Therefore at this point, the transition probability matrices and reward functions are updated in order to reflect and adapt to the current event. With these updates, the system will recalculate the optimal policy and use it to accomplish the reconfiguration. Thus, the resource will be reallocated to different agents forming the new system configuration most suitable to deal with the current situation. Figure 80 illustrates the reconfiguration map after the mission is changed. The numbers with colors represent the states of the agents. From this figure we can see that, during the first 50 decision epochs, the agents 13 and 16 are in desired states. After the mission change happens, the ship is reconfigured to a new mode so agents 1, 8 – 11, 14 and 15 obtain the required resource and are in working state. From scenario 1, we know these agents have high priority under a battle mission, thus their availabilities highly increase the mission effectiveness. This indicates that the system has capability to reconfigure itself into the state most suitable for the current situation.

In summation, the three scenarios show that the resource allocation advisor is able to obtain the optimal policy which can enable the ship operate on the best course of action, and allocate the resource to achieve the reconfigurability for current situation. Instead of using intuition or expert opinion, the ship is capable of making autonomous decisions which highly increase the mission effectiveness and reduce operating cost due to reducing the manpower for the system operation under the dynamic environment with the existence of uncertainty.

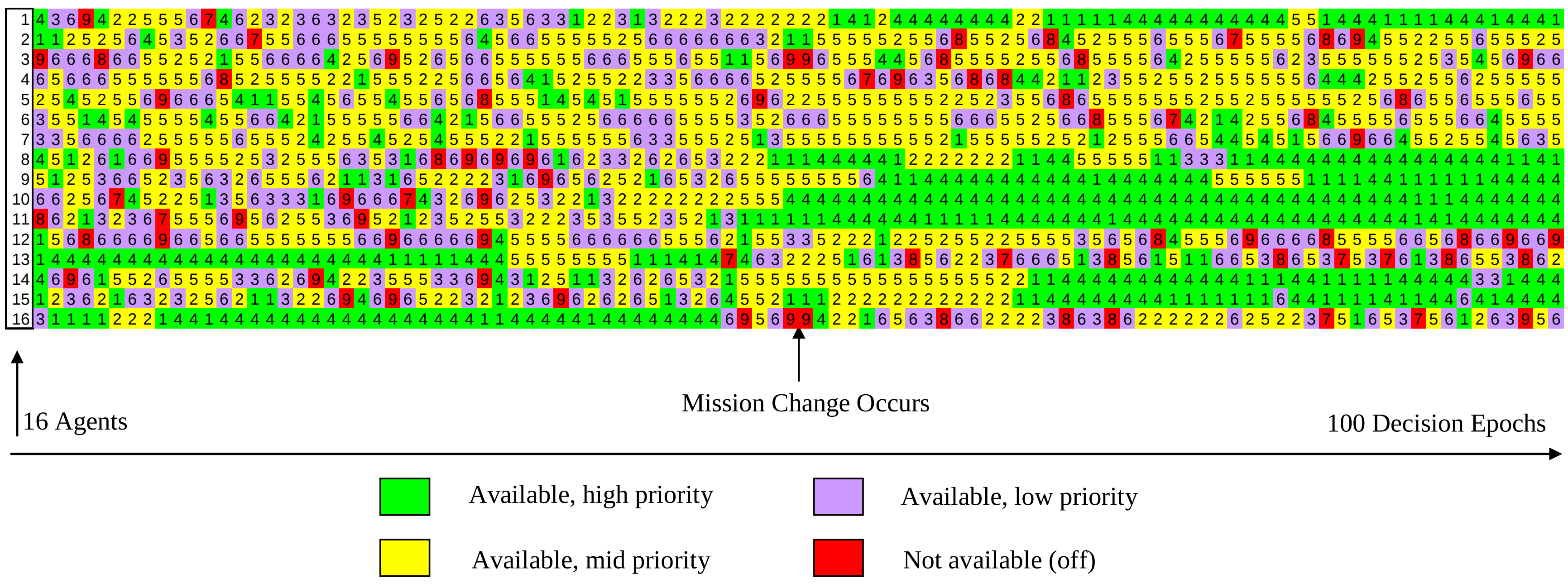


Figure 80: Reconfiguration Map after the Mission Change Occurs

6.3.4 Result Discussion

Complex systems like aerospace vehicle and naval ship mainly operate in a dynamic environment where decision making is required to be performed under uncertain conditions. In order to increase mission effectiveness and reduce operating cost, decisions need to be made autonomously to perform the desired mission or handle emergent events. This requires the systems have reconfigurability in response to dynamic changes in theirs environment. The proposed resource allocation advisor can derive the optimal policy which manipulates the system, advises on the best course of action, and reallocates the required resource to the system to reconfigure it into the best mode for reaching the objectives of the operation.

The resource allocation advisor employs a constrained multi-agent Markov decision process formulation to carry out the resource allocation and reconfiguration. MDP provides an effective approach to formulate the problem of sequential decision making in stochastic process. The other approaches discussed, such as dynamic decision analysis and AI planning, are alternative methods to deal with the dynamic decision making under uncertainty, but they are difficult for practical application. Dynamic decision analysis requires the decision maker have knowledge about the consequences of the decision, such as the effect of an action. This information is often uncertain and needs much effort to be discovered. These difficulties prevent dynamic decision analysis from being a widely used method for decision making under uncertainty. AI planning can handle the uncertain conditions in the dynamic decision making process, but the use of AI planning requires the developer to identify and handle complex numerical and logical relations. This disadvantage makes AI planning difficult to apply.

The MDP also has its own limitations. In order to get an optimal policy, the accurate transition probability function is needed. To formulate this transition probability function is always time-consuming and requires much effort. The transition probability function is typically estimated using historical data or calculated based on simulation program.

Since the system operation involves multiple agents and they share limited resources when desired mission is performed, the MDP formulation must be able to handle multi-agents and constraints. Dynamic programming including value iteration and policy iteration provides an effective way to solve the MDP, but it has disadvantages when applied to constrained multi-agent MDP. This is because as the number of agent increases, the state and action space of the MDP increases exponentially, making it intractable for dynamic programming techniques. The linear programming technique provides an explicit and easy way to deal with the constrained multi-agent MDP and it is used as the technique to calculate optimal policy for the resource allocation formulation. Results show that linear programming can efficiently handle the constrained multi-agent MDP.

CHAPTER VII

CONCLUSIONS AND RECOMMENDATIONS

Decision making in system design plays a critical role in determining the success of a design solution. Dynamic decision making under uncertainty is pervasive in the complex system operation and often has profound impact on the mission effectiveness. The first part of this dissertation addressed a multi-criteria decision making problem in which a framework was established to select the most appropriate decision making method for a given problem and provide advice for new method development. A personal air vehicle concept selection problem was performed as a proof of concept. The second part of this dissertation examined a multi-agent resource allocation problem and formulated a constrained Markov decision process to realize autonomous decision making under uncertain conditions for resource distribution. The method was applied to a reduced-scale advanced demonstrator with 16 agents as a proof of implementation.

7.1 Research Questions Answered

> ***Question 1:*** *How to represent different methods in order to capture their essence for method selection?*

The answer to this question forms the basis of the method selection approach. Literature search shows there are various decision making method available, thus it is important to use the one which is the most appropriate for the problem under consideration. In this dissertation, a decision making method is decomposed into decision maker related, method related, problem related and solution related characteristics

(Section 3.2.1.4), as proposed by **Hypothesis 1**. Different methods have different values of the associated characteristics which will be evaluated in the process of selecting the most appropriate method.

> ***Question 2:*** *How to evaluate the appropriateness of the methods for the problem under consideration?*

Since different methods have their own advantages and disadvantages, it is necessary to find a way to evaluate the method with respect to the problem under consideration. **Hypothesis 2** proposed to develop an algorithm to rank the decision making methods based on the given problem. The study presented in this thesis proposed an evaluation criterion called appropriateness index, as described in Section 3.2.2.1, which was derived from the concept that the most suitable method selected for the given problem should possess the capabilities to address the problem related, decision maker related and solution related characteristics of the decision making problem. Thus, the method with the highest appropriateness index will be considered as the most appropriate method to solve the given problem. Notice that the calculated appropriateness index of a method is only valid for the problem under consideration, that is, the appropriateness of a method varies with the given problem.

> ***Question 3:*** *In the case that DMs have limited knowledge about other methods*
>
> *(a) how does one to determine the validity of the decision made by the DMs using the method they are familiar with*
>
> *(b) is there a decision making formulation that allows DMs to select and utilize the most appropriate method to solve their decision problems?*

It is not necessary for a decision maker to know the methods that he or she does not have to know. The limited knowledge about other methods often makes it difficult for the decision maker in finding and using the most appropriate method. As a result, the decision maker often tends to use the method which he or she is familiar with to solve different problems. However, the use of an inappropriate method often leads to a misleading solution. Therefore, one must validate the decision made by using one specific method. **Hypothesis 3** proposed to allow the decision maker to select the most suitable method and then use it without being familiar with the selected method. This hypothesis leads to two main tasks for this thesis (Section 3.2.2.1, 3.2.2.2 and 3.2.2.3) and motivates the development of the Multi-Criteria Interactive Decision-Making Advisor and Synthesis process (MIDAS). In MIDAS, widely used methods are nominated as the candidate methods and stored in a database called method library. In addition, necessary knowledge associated with the methods are acquired and stored in a knowledge base to support the method selection in an interactive way.

> ***Question 4:*** *Can advice be given if no method in the method pool is suggested for the given problem?*

Since the number of the methods in the method library is limited and new advanced methods are emerging over time, it is not possible to include all the advanced methods in the MIDAS. As a result, the suitable method for the given problem may not be found in the method base. However, the characteristics of the problem can be recognized in the process of method selection, so the capabilities required to solve this problem can be derived based on the problem's characteristics. Thus, by analyzing the required capabilities the MIDAS can produce some advice which can be used as the hints to generate a new method or hybrid method, as proposed in **Hypothesis 4**. The detailed explanation of new method generation was described in Section 3.2.2.4.

Question 5: *Can the method selection be handled in an efficient manner?*

In order to find out the most suitable method, the decision maker has to manipulate a large amount of knowledge associated with the characteristics of the problem and methods. This is often time consuming and a source of frustration. **Hypothesis 5** proposed to use an advisor system to effectively facilitate the method selection process which directly resulted in the development of the MIDAS. MIDAS is designed as a knowledge based advisor system that allows the user to obtain the most appropriate method interactively and efficiently. The capabilities of the MIDAS can be found in Section 3.2.2.

Question 6: *Is there a decision making formulation that can effectively make the real time decisions reacting to the current ship situation based on uncertain information?*

The IRIS concept is envisioned to be able to make dynamic decisions under uncertain conditions to rapidly react to the current ship state. **Hypothesis 6** proposed to formulate the dynamic decision making problem using a Markov decision process which can find an optimal policy and then use it to realize autonomous decision making. This leads to a task to develop an autonomous decision making approach formulated as a Markov decision process to produce optimal policy. The actions taken in each state are manipulated by the policy and maximize the expected total reward. Thus, the best course of action can be found by following the optimal policy and the execution of the best course of action will lead to a plausible reconfiguration of the ship.

> ***Question 7:*** *Is there a mathematical formulation of the resource allocation that can effectively distribute the shared resource to each agent?*

The resource allocation problem involves multiple agents and resource available constraints. At each decision epoch, the agents require the resources in order to work properly. Since the resources are shared and limited, decisions need to be made on whether an agent can get the required resource. **Hypothesis 7** proposed to employ a constrained multi-agent Markov decision process to formulate the resource allocation problem. This directly motivated the development of the resource allocation advisor which encompasses a constrained multi-agent MDP formulation. The constrained multi-agent MDP can be solved using the linear programming technique, as shown in Section 6.2.2. And a more sophisticated constrained multi-agent MDP problem is formulated in Equation (58) to handle the operational, consumable and recyclable resources, and the problem can be reduced to a standard linear program or mixed linear integer program.

7.2 Summary of Contributions

One of the objectives of this research is to develop an approach to select the most appropriate decision making method for a problem in system design, and then use the selected method to solve this decision making problem. A Multi-Criteria Interactive Decision-Making Advisor and Synthesis process (MIDAS) was proposed as the solution approach. The other objective of the research is to establish a formulation for making autonomous decisions under uncertain conditions in complex system operation. A resource allocation advisor encompassing a constrained multi-agent Markov decision process was formulated as the solution to perform the dynamic decision making under uncertainty.

The method selection approach: It is important to select the most appropriate decision making method since the use of inappropriate method usually create misleading solution. The current method solution approaches have their own disadvantages and limitations. All the approaches can not provide guidance to reach the final solution to the decision making problem, however, which is often the ultimate goal of the decision maker.

To fill this gap, a knowledge based decision making method selection approach was developed in this dissertation. The approach starts to decompose the problem into different characteristics which can capture the essence of the problem. The characteristics have the relative weight determined by the decision maker's preference. A set of decision making methods are nominated as the candidate methods among which the most appropriate method will be selected. Based on the problem's characteristics and the corresponding relative weights, the methods are ranked in order by an appropriateness index which was proposed as the evaluation criterion. The method with the highest ranking will be selected as the best method and can be used to solve the given problem.

In the case that no method in the method pool is suggested, a new method may be developed. Through the problem decomposition, the characteristics of the problem are recognized, and thus, the capabilities needed to be fulfilled by a method can be obtained. The exploration of the capabilities often creates some hints that can be used to create a new method that is capable of dealing with the problem under consideration. This new method can be a brand new or hybrid method which combines certain capabilities of two or more existing methods.

The method selection approach is realized by a Multi-Criteria Interactive Decision-Making Advisor and Synthesis process (MIDAS). MIDAS is a knowledge based advisor which incorporates a knowledge base and method base to support the method selection process. MIDAS provides an interactive way to let the user to select the method and then direct him or her to use the selected method to reach the final design decision. MIDAS allows the decision maker to select and use the most appropriate method even when the decision maker does not know how the method works. It can also produce the hints for new method development if there is no method is suggested for the given problem. In general, MIDAS provides an interactive way to effectively fulfill the method selection task.

New method developed: As the solution approach was applied to the Personal Air Vehicle concept selection problem, the limitations of the JPDM technique were discovered. With the use of the MIDAS, several suggestions were provided serving as the hints to develop new methods. A hybrid decision making method was developed based on one of the hints by combining the JPDM technique with the utility function. Study in Section 4.1.2.3 shows that the developed hybrid method has improved performance over the traditional JPDM technique for the concept selection problem.

Autonomous decision making formulation: In order to increase the mission effectiveness and reduce operating cost, the decision making in the complex system operation should be performed autonomously. However, currently many critical ship systems, such as chilled water system, are operated manually, which produces a need for an advanced approach to facilitate the decision making tasks in the ship operation. A resource allocation advisor is proposed as a solution to perform autonomous decision making on distributing the resource for a chilled water reduced scale advanced demonstrator. This advisor encompasses a constrained multi-agent MDP formulation which is able to model the recyclable resources that the existing approaches do not consider. The formulation proposed to represent the system state as the combination of the mission state, environment state and status state. The resource allocation problem can be solved by finding an optimal policy to the Markov decision process. Then the autonomous decision making can be realized by following the optimal policy and the best course of action can be derived to achieve the resource allocation and reconfiguration capabilities.

7.3 Further Work and Recommendations

The following sections discuss ideas for further work and recommendations for accomplishing it.

Method Selection

The knowledge utilized to support the method selection is represented by a set of predefined decision rules. With the new problem and method emerging, some of the knowledge may be inconsistent with the reality it described due to the incomplete information it represents. In this case if a decision maker, especially an expert, has high confidence about the desired result of the method selection and the expected result is not consistent with the one that the MIDAS gives, the advisor needs to make changes to the corresponding knowledge to adapt itself to this type of decision problem. Thus, the advisor will produce the desired result next time when the same type of the problem occurs. This implies that if the MIDAS possesses the adaptive capability, the method selection process can be handled more effectively.

In addition, the method selection can be facilitated using a web-based decision support system. The web-based application allows the different users to share their resources such as knowledge and method base. As a result, the knowledge base and method base can be extended and updated more efficiently, and this will increase the capabilities of the MIDAS. The web-based application also allows the ontology based technique to be fulfilled, which will make the system more manageable.

Investigation of the Interactions between the Actions

In this dissertation, the actions are assumed to be executed independently in the autonomous decision making process, that is, performing one action does not trigger the occurrence of other actions. In some cases there are interactions between different actions so that the execution of one action often results in the execution of another. The interactions between actions affect the course of events and as a result will lead to a change in the decision making strategy. Therefore, in order to make proper decisions, a new policy needs to be formed through a Markov decision process with the consideration of the interactions between the actions.

Investigation of the Interdependencies of the Systems

In a complex system operation, the performance of some subsystems may depend on the states of the other subsystems. For example, when chilled water system supplies the cooling fluid to cool the electrical system, it requires certain electrical power to keep the chillers and pumps operating properly. These interdependencies among the systems affect the consequence of the decision made in each system state, thus they need to be taken into account when choosing the best course of action. In order to handle the interdependencies among the systems, one can use the abstraction strategy mentioned in Section 6.2.4 or a more sophisticated technique capable of solving the dynamic decision making problem with interdependent subsystems.

With the consideration of the interactions between the actions and interdependencies among the subsystems, the resource allocation problem can be modeled more accurately, and improved decisions on resource distribution can be made. This may be an important improvement for the system operation, especially for large complex system operation under uncertain conditions.

APPENDIX A

FUNDAMENTALS OF DECISION MAKING

A.1 Structure of Decision Making Process

In the past decades, various methods and techniques were proposed to facilitate the decision making process in engineering design. Each method is designed to solve a class of problems by following a specified decision making process. Simon [1960] pointed out that, in general, the decision making process can be categorized into programmed and non-programmed processes. The programmed process is referred to as a decision making process that can be formulated in an analytical way and completed by a computer system without the active involvement of human decision maker. On the other hand, non-programmed decision making process cannot be simply formulated as a predefined analysis model and needs decision maker to make effective decisions to complete the process. Similarly, Keen and Scott-Morton [1978] stated that there are three types of decisions: structured, semi-structured and unstructured or ill-structured decisions. Table 46 lists the properties of the three types of decisions.

In the case of structured decision making, the process is mathematically formulated as an analysis model, thus the inputs and outputs are specified. This makes it suitable to program using computer system to solve this kind of decision making problems. The semi-structured decision making process is often composed by two types of decision: one is well predefined and the other is vague at the beginning of the process. In the unstructured decision making, the decision making process varies with the particular decision task. Since it can not be modeled mathematically, the unstructured decision making does not have an explicit list of inputs and outputs and cannot be programmed.

Table 46: Properties of Three Decision Making Processes

Structured	Semi-structured	Unstructured
The decision task is explicitly defined and the input data and the results of their process are specific	Some elements of the process are well structured and explicitly predefined	The process of reaching to the decision is always different, depending on the particular decision task
The process that is followed for making the decision is always the same	Some other are rather vague in the beginning of the process	The objective of the decision making process, the input data and the results are not explicitly defined
Programmed	Some elements of the process are programmed, the others are non-programmed	Non-programmed

A.2 Level of Decision Making

It is clear that some decisions are more important than others, whether in their short term or long term impact. The appropriate significance of a decision often determines how much time and resource should be spent to fulfill the decision and how risky a bad decision would be. Based on the effects of the decisions, the decision making can broadly be classified into three levels: strategic, tactical, and operational, as shown in Figure 81.

Strategic decisions are made by the top level management and are the most important decisions that affect the final result of a problem. The strategic decisions concern general direction, overall objectives and long term goals of the decision making problem. For instance, in aircraft design, the decisions about what product should be launched to gain the market share, which concept should be selected for further analysis and what approaches will be utilized to investigate the problem belong to the category of strategic decisions. Since strategic decisions have critical impact on the final solution of the decision making problem and are far into the future the decision making process often involves the risk and uncertainty analysis.

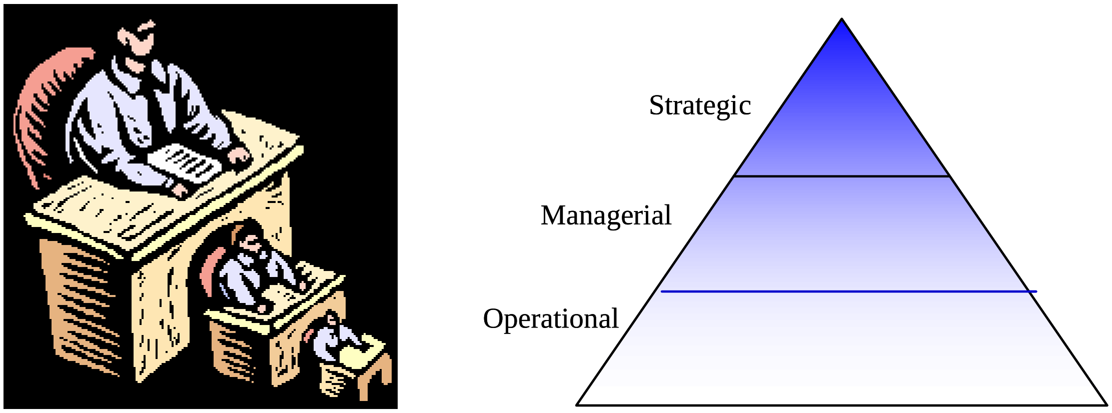


Figure 81: Levels of Decision Making

Tactical decisions are concerned with the best use of resources to achieve mid-term objectives in order to fulfill the overall goal. These decisions are made by middle level decision makers to support the strategic decisions. Tactical decisions often have moderate consequence which is relatively important to the final result. In tactical decision making process, since decisions are often made based on incomplete information uncertainties and risks always need to be taken into account. For example, once the strategic decisions have been made on what product should be selected for launching a tactical decision would be to decide when to launch it.

Operational decisions are usually made at lower level in order to put tactical decisions into effect. They are basic decisions that have immediate impact. Their impacts are usually short term, short range and low cost. The consequences of a bad operational decision will be minimal, although a series of bad operational decisions can cause harm.

For example, if the tactical decision is to use probabilistic design to develop new concept for personal air vehicle, the operational decision would involve how to set up the problem, and what objective function will be used for optimization or concept selection.

Though the consequences of the decisions vary a lot, decision making should be carefully carried out at each level since the achievement of the overall goal requires the combined effects of all the decisions. The higher level decisions provide directions and objectives to the lower level decisions while lower level decisions put the actions into effect to support the higher level decisions. For a new project, such as the design of a large commercial aircraft, the decision making should be performed hierarchically in order to successfully reach the ultimate goal. If nearly all the decision making is taking place at the operational level, then not enough thinking and planning will be done at the strategic level. As a result, the directions and goals are most likely out of control, and DM will have to passively deal with the forces around them, for example, changes may need to be done in the late design phases which will dramatically increase the cost.

APPENDIX B

ADDITIONAL MCDM METHODS IN MIDAS

Most of the methods in MIDAS have been described in Section 2.6.2 and 2.6.3. This section will introduce some additional MCDM methods in MIDAS and explain the JPDM technique in detail.

B.1 Joint Probability Decision Making Technique

Among the decision-making techniques, traditional single criterion approaches fail to account for the entire system. On the other hand, current multi-criteria approaches require deterministic information for the system and environment, while such information is not typically available at the conceptual or preliminary phases. Moreover, the use of new technologies adds more uncertainty to the design process due to readiness or availability issues.

The Joint Probabilistic Decision Making (JPDM) technique provides a formulation that is capable of dealing with the multiple criteria and capturing uncertainties which often exist in today's systems design. This technique utilizes the multivariate probability theory to construct a joint probability distribution (Figure 24) which reflects multi-criteria and probabilistic natures of the decision making problems. Based on the type of the joint probability distribution, JPDM is classified into Joint Probability Model (JPM) and Empirical Distribution Function (EDF) model. JPM is a parametric model whose Joint Probability Density Function (JPDF) can be derived from the marginal PDFs analytically using the relative statistics. EDF is an empirical model whose joint probability distribution can be obtained from a set of data generated by some sampling technique. Figure 82 shows the JPM and EDF models in the JPDM environment for a two dimensional decision making problem.

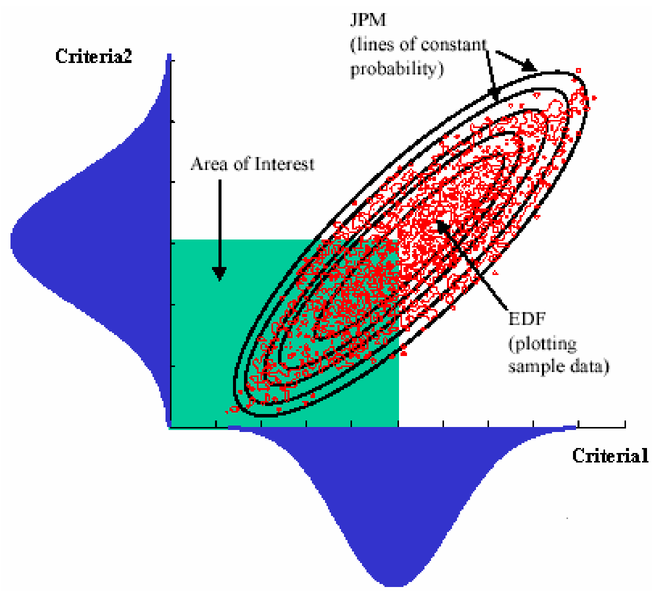


Figure 82: JPDM technique visualization

The joint probability distribution is generated and serves in conjunction with a criterion value range of interest as a universally applicable objective function. The objective function, referred to as Probability of Success (POS), is a meaningful metric that allows the decision maker to make a decision based on the chance of satisfying the criteria concurrently. In the JPDM, the PoS is obtained by integrating the joint probability density function over the area of criterion values that are of interest to the customer for JPM model, or by counting the number of the occurance of the alternative solutions within the area of interest for EDF model.

The JPDM technique can handle two types of problems: optimization and product selection [Bandte, 2000]. For an optimization problem, the POS is optimized to account for all criteria concurrently. In a product selection problem, the JPDM provides a compensatory technique that allows the comparison of alternatives on an equal basis.

Figure 83 depicts the JPDM process for optimization and product selection. First of all, the criteria based on which solutions are evaluated require to be identified. Then control and noise variables need to be determined and used as the inputs of the analysis, and an appropriate distribution should be assigned to each noise variable to represent its uncertainty. The responses of the decision making problem can be calculated in the modeling and simulation environment which often encompasses some sizing and synthesis programs. Then the joint probability distribution of the criteria can be constructed using JPM or EDF models and then the joint POS can be computed over the area of interest. The relative weights of the criteria, representing the decision maker's preference information, are applied when the joint POS is calculated. Then the calculated POS is compared with the required POS which is predefined as one of the requirements. If the calculated POS is less than the required POS, necessary modifications, such as the ranges of the control variables or the relative weights of criteria, need to be made. As a result, the process will be carried out iteratively until the requirements are satisfied.

Though JPDM is a powerful technique to handle the MCDM problem, it has its own limitations. In the EDF model, large amounts of data are required and need to be accurate. However, the data are usually not easy to be obtained in conceptual and preliminary design phases. The JPM model requires a correlation function which often is not available in the early design stages. In addition, the calculation of the POS does not take the absolute location of the JPDF into account, which leads the JPDM to become awkward for concept selection when the calculated POSs of the alternatives are very similar but their JPDF locations are very different. Therefore, for the general case the value of the POS cannot accurately represent the goodness of the alternative because it does not take the deviation of the criteria from their target values into account. These limitations make JPDM incapable of dealing with some of the decision making problems alone.

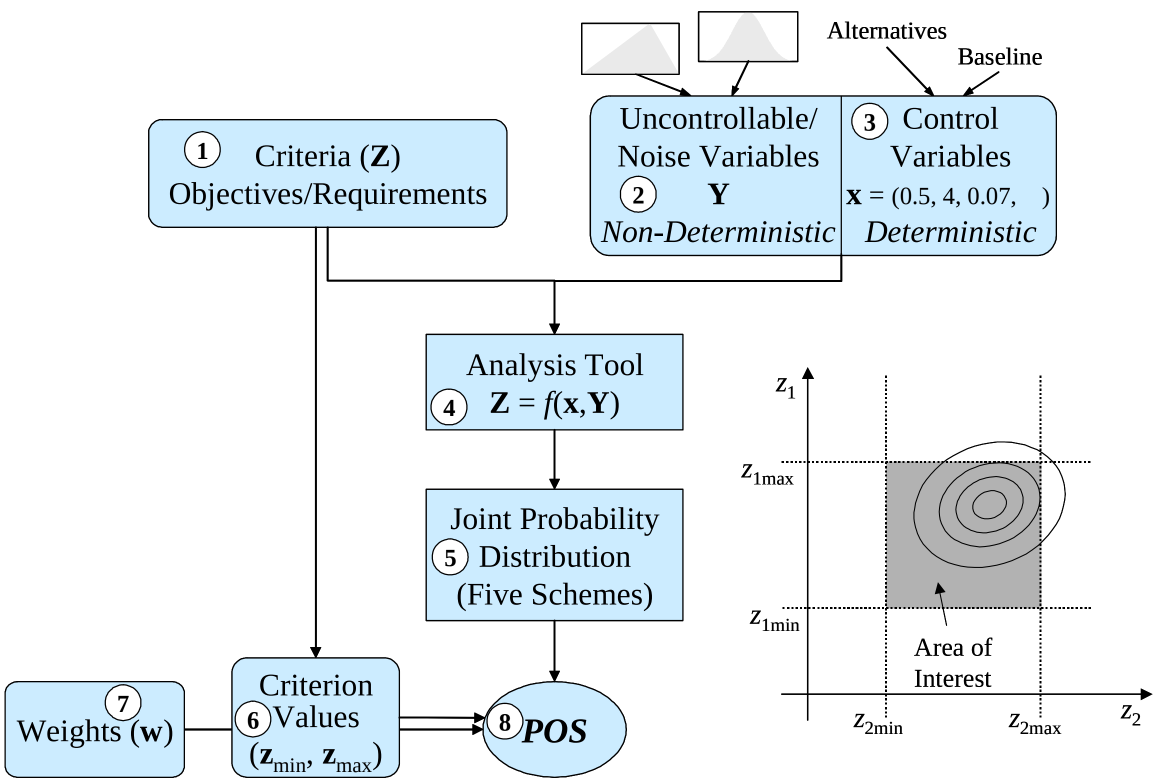


Figure 83: Joint Probability Decision Making Technique

B.2 Multi-Attribute Utility Theory

Multi-Attribute Utility Theory (MAUT) is based on the use of utility functions. A utility function represents a mapping of the decision maker's preference onto a mathematical function so allows the preference information to be expressed numerically [Ang and Tang, 1984]. For a decision making problem with multiple attributes, a utility function is assigned to each attribute to reflect the decision maker's preference information. Usually, a more preferred performance value of the attribute obtains a higher utility value. For example, if cost is identified as an attribute its associated utility function would have higher utility values for lower cost values

The multi-attribute utility function can be obtained from the assessment of the utility function for each single attribute. The most widely used multi-attribute utility function form is the additive one:

$$U(x_1, x_2, \cdots, x_n) = \sum_{i=1}^{n} w_i u_i(x_i) \qquad (61)$$

where $\sum_{i=1}^{n} w_i = 1$, w_i is the relative weight of attribute x_i and $u_i(x_i)$ is the associated utility function.

The multi-attribute utility function given by Equation (61) is based on two assumptions which are verified to be appropriate for many realistic decision making problems [Keeney and Raiffa, 1993]. They are:

- The utility functions of all the attributes are independent each other;
- The relative weight of an attribute can be determined regardless of the relative weights of other attributes.

Besides the above additive model, Edwards [1977] also proposed a simple method to assess weights for each of the attributes to reflect its relative importance to the decision. The method utilizes a 10 point scale to represent the relative importance of the attribute.

10 points are assigned to the lease important attribute and 1 point is assigned to the most important attribute. The final relative weights are computed by normalizing the sum of the points to one. Furthermore, Edwards and Barron [1994] pointed out that it is necessary to consider the amplitude of the utility values when the importance of the attribute is compared. They proposed to use swing weighting method to derive the relative weights for each of the attributes, thus more realistic weights can be obtained.

B3. MaxMin

MaxiMin, originated from economics, is a widely used MADM technique. The MaxiMin method is based upon a decision making strategy that tries to avoid the worst possible performance by maximizing the minimal preferred performance value of each criterion. This method chooses the alternative for which the performance of its weakest criterion is the highest. That is, the MaxiMin identifies the weakest criterion for each alternative first, and then selects the alternative that has the highest value in its weakest criterion. The mathematical description of the MaxiMin method is given by Equation (62).

$$A^{+} = \{A_j \,|\, \max_j(\min_i(x_{ji}))\}, \tag{62}$$

From Equation (62), one can clearly see that the MaxiMin method "is selecting the maximum (across alternatives) of the minimum (across criteria) values" [Bandte, 2000]. The decision is made based on the value of the weakest criterion of an alternative in spite of the values of all other criteria. Thus, this method reduces the multi-criteria decision making problem to a single criterion decision. This simplicity makes the MaxiMin method a widely used method, particularly in game theory.

However, the Maximin method has its own disadvantages. This method only considers the weakest criterion of an alternative so it just utilizes a small amount of the available information during the decision making process. This fact often results in throwing out an alternative which has worse weakest criterion but is much better in all other criteria than the selected alternative. In addition, the use of the MaxiMin method requires that all criteria are comparable in order to measure the criteria on a common scale, which is another limitation of the MaxiMin method [Linkov et al., 2004].

B4. Genetic Algorithm (GA)

The Genetic Algorithm (GA) is a type of evolutionary algorithm used in computing to find approximate solutions to optimization and search problems. The basis of GA is the use of an adaptive heuristic global search algorithm originated from the evolutionary ideas of natural selection and genetic. GA is designed to simulate Darwin's evolutionary process of survival of the fittest in natural system. By mimicking this process, genetic algorithm is able to evolve solutions to realistic problems by performing an intelligent exploitation of a random search within a defined search space. It has been demonstrated that GA is capable of efficiently finding the global optimum for a MCDM problem. The GA has five major steps: initialization, evaluation, selection, crossover and mutation. Figure 84 depicts the steps of the genetic algorithm.

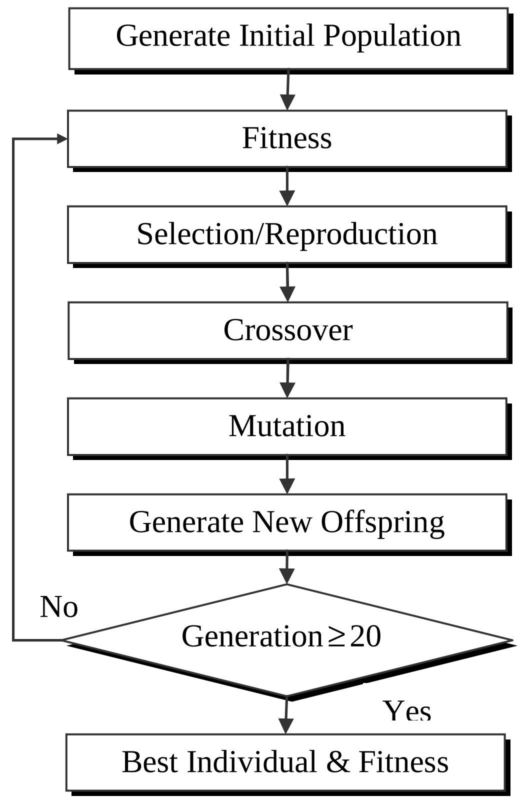


Figure 84: Genetic Algorithm

The GA starts from the creation of an initial population which is chosen randomly from the space defined by the independent variables. The individuals of the population are usually encoded as binary strings of 0s and 1s (called chromosomes). The individuals then are evaluated and a fitness value is assigned to each individual. Once evaluated, the "parents" for each generation are stochastically selected based on their fitness, and this step is known as "selection". After the new population is established, the genetic material of the parents is combined to create children by performing a crossover operation. The crossover is accomplished by randomly selecting a splice point in the binary string and then swapping bits between the parents at the splice. Once the crossover is done, the mutation operation is applied. In the process of mutation, the value of a bit is changed (0 changes to 1 and vise versa) with a specified mutation probability. Thus a new pool is established and their value is evaluated again. If the best individuals and fitness are obtained the algorithm stops, otherwise a new iteration is executed until the desired results are reached

APPENDIX C

INTEGRATED RECONFIGURABLE INTELLIGENT SYSTEMS

During the last decades, incremental improvements in ship design, operation, and capability have been achieved. With the rapidly changing fiscal and threat environment modern ship design is putting more emphasis on reducing operating cost and manning workload, and increasing ship survivability and mission effectiveness. The Office of Naval Research (ONR) proposed an Integrated Engineering Plant (IEP) concept to meet such requirements for next generation naval ship. IEP is a unified system that removes traditional system-level barriers between the various ship plants, such as propulsion, weapon, electrical and cooling systems. Thus, the ship plants can share the resources and information managements systems which leverage the resources and deliver the information to the plants from a system point of view. IEP is a highly decentralized system in which plant components can perform the predefined or self controlled tasks. In addition, the IEP system allows the next generation Navy ships to operate under major disruptions involving cascading failures and provide continuous mobility, power, thermal management and fluid transfer for vital shipboard systems, as a result, reducing manpower requirements and increasing overall ship survivability and mission effectiveness. This revolutionary change in naval architecture and ship engineering requires a total ship systems engineering design approach which is able to formulate and implement the design methods and tools to the ship systems and capable of extensive, autonomous decision making. Figure 85 depicts the IEP concept.

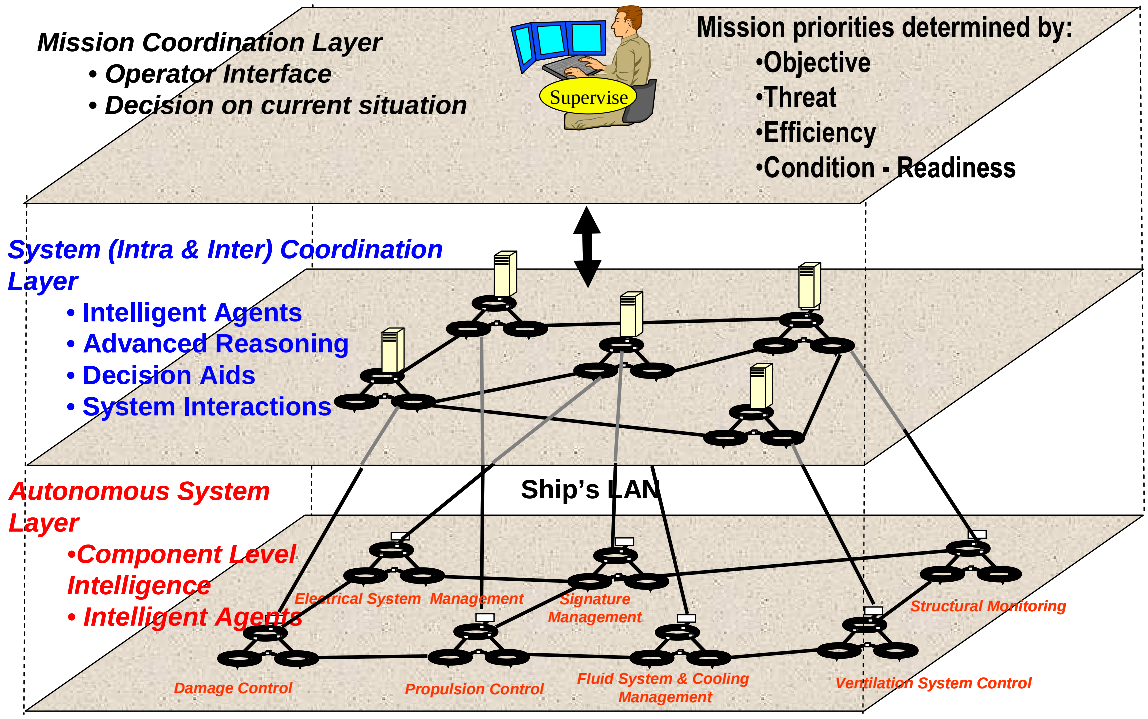


Figure 85: IEP Concept [Walks and Mearman, 2005]

Aerospace Systems Design Laboratory (ASDL) at Georgia Institute of Technology formulated an Integrated Reconfigurable Intelligent System (IRIS) framework as a possible solution to the IEP concept. The IRIS integrates many intelligent systems onboard to collect the information about the environment and ship state, assess the situation and then take a best course of action to reconfigure the ship into the state that most suitable to handle the situation at hand. Therefore, the IRIS designed ship is envisioned to be self-monitoring, self-assessing and self-reacting, as shown in Figure 86.

The "integrated" in IRIS indicates that the design of the system is shaped by the integration of intelligent and reconfigurable subsystems. The IRIS framework utilizes an integrated simulation environment to model the interdependency between the systems. This integration helps to reduce manpower and increase mission effectiveness, survivability, and reliability of the overall system.

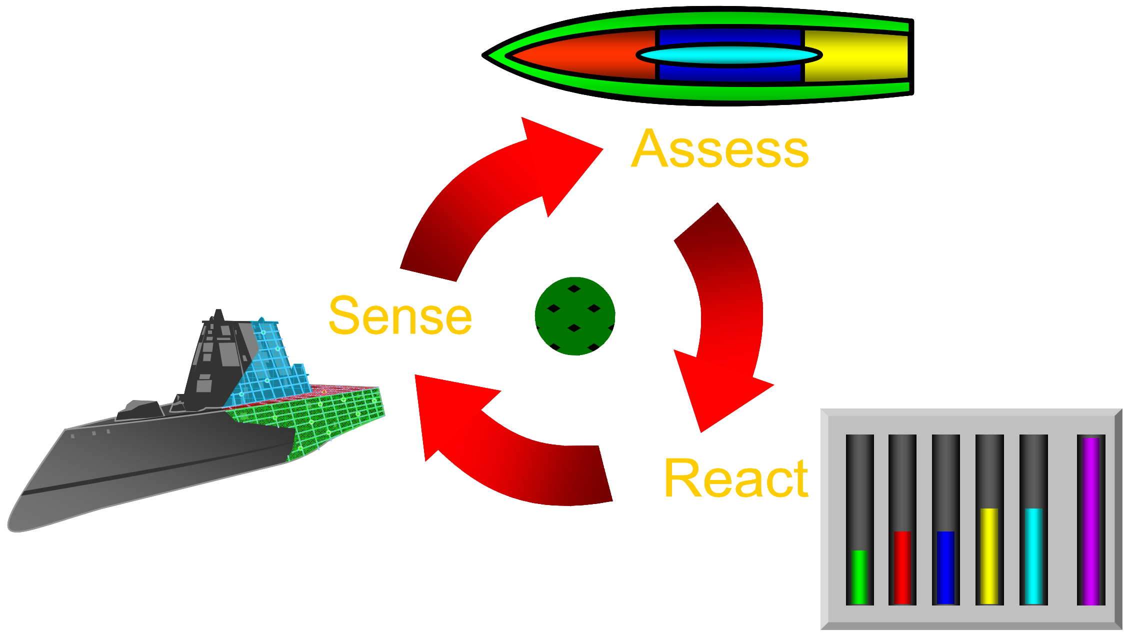


Figure 86: IRIS Framework

The reconfiguration is the overall goal that the IRIS eventually needs to achieve. The IRIS system is capable of fulfilling three type of configuration: design reconfiguration, mission reconfiguration and dynamic reconfiguration. Design reconfiguration allows one to use a modular architecture that utilizes equipment, and allows one to remove and replace particular platforms seamlessly as new technologies become available. Mission reconfiguration can be accomplished by reconfiguring all of the ship subsystems into their optimal states to perform the given mission. Dynamic reconfiguration gives the ship the ability to diagnose, react, and continue to operate under major disruptions.

The intelligence of the IRIS system lies on the fact that the ship uses smart sensors to accurately sense and assess situations. In addition, the system is aware of its status by gathering data from the sensors onboard the ship, and then assesses the obtained information to either make autonomous decision or provide suggestion to the human operator to support their decision making activities.

The IRIS framework integrates various systems including people, products and processes that provide a capability to satisfy a stated need or objective. The IRIS employs a modeling and simulation environment to integrate the design of complex systems consisting of propulsion, weapon, radar, cooling and damage control subsystems and these subsystems work dependently to optimize the overall objective.

APPENDIX D

ADDITIONAL SIMULATION RESULTS

Table 47: Optimal Policy and Maximum Immediate Reward Policy for Scenario 1

Optimal Policy

State i	π_{i1}	π_{i2}	π_{i3}	State i	π_{i1}	π_{i2}	π_{i3}	State i	π_{i1}	π_{i2}	π_{i3}
1	0.87	0.13	0	1	0.52	0.48	0	1	0.51	0.49	0
2	0.80	0.20	0	2	0.73	0.27	0	2	0.62	0.38	0
3	0.99	0.01	0	3	0.45	0.55	0	3	0.72	0.28	0
4	0.93	0.07	0	4	0.63	0.37	0	4	0.64	0.36	0
5	0.83	0.17	0	5	0.92	0.08	0	5	0.72	0.28	0
6	0.46	0.54	0	6	0.63	0.37	0	6	0.92	0.08	0
7	0.37	0	0.63	7	0.42	0	0.58	7	0.40	0	0.60
8	0.69	0	0.31	8	0.33	0	0.67	8	0.41	0	0.59
9	0.78	0	0.22	9	0.23	0	0.77	9	0.36	0	0.64

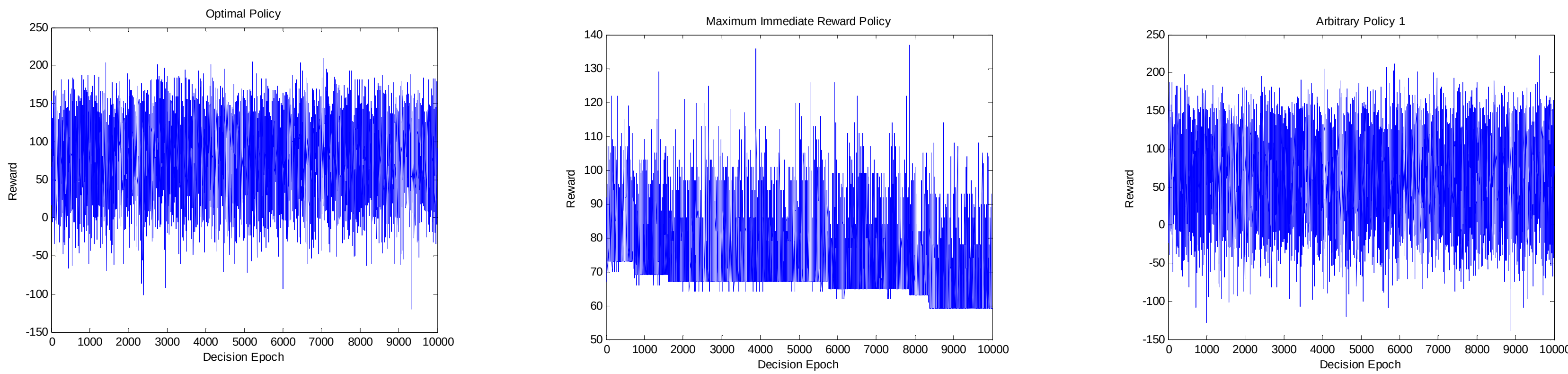
Maximum Immediate Reward Policy

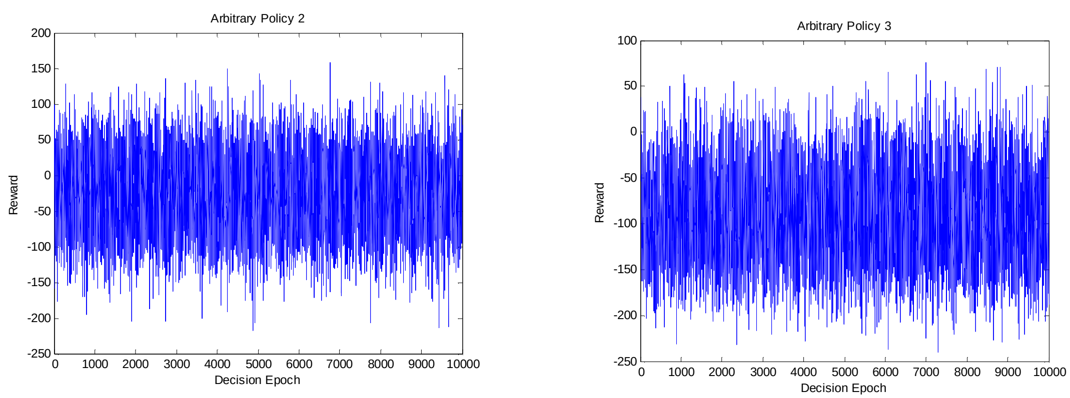
State i	π_{i1}	π_{i2}	π_{i3}	State i	π_{i1}	π_{i2}	π_{i3}	State i	π_{i1}	π_{i2}	π_{i3}
1	1	0	0	1	1	0	0	1	1	0	0
2	1	0	0	2	1	0	0	2	1	0	0
3	1	0	0	3	1	0	0	3	0	1	0
4	1	0	0	4	1	0	0	4	1	0	0
5	1	0	0	5	1	0	0	5	1	0	0
6	1	0	0	6	0	1	0	6	0	1	0
7	0	0	1	7	0	0	1	7	0	0	1
8	0	0	1	8	0	0	1	8	0	0	1
9	0	0	1	9	1	0	0	9	1	0	0

↑ For agents with high mission priority | ↑ For agents with mid mission priority | ↑ For agents with low mission priority

Starting Point

Agent	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
State	1	8	5	1	1	8	5	3	8	8	4	7	9	3	5	4

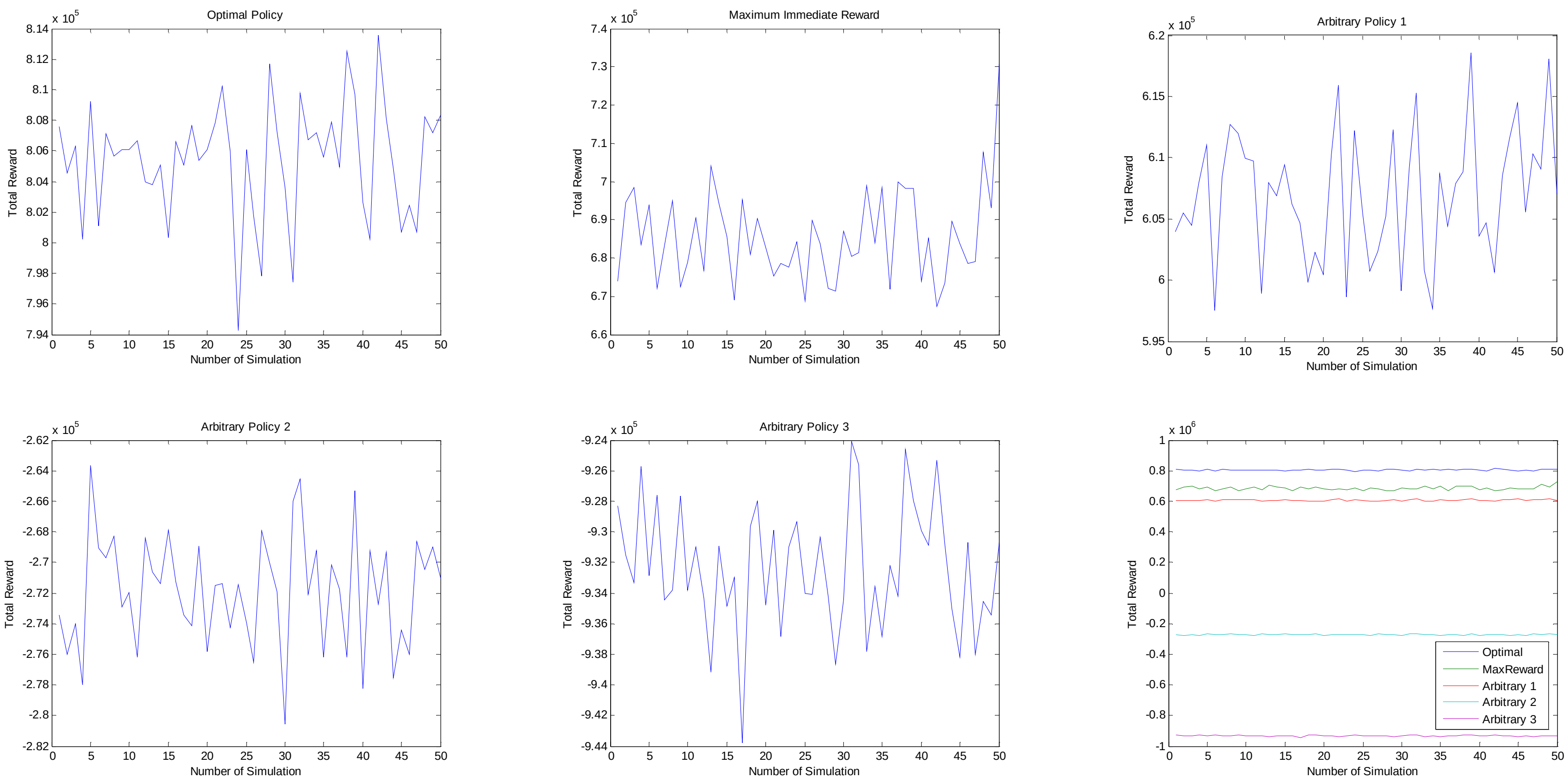




Reward Statistic

Policy	μ	σ
Optimal	80.7618	43.867
Max Immediate Reward	67.3912	10.4301
Arbitrary 1	60.3946	50.4687
Arbitrary 2	-27.344	51.371
Arbitrary 3	-92.8306	46.9509

Figure 87: Reward Trajectories of Five Policies for Scenario 1



Total Reward Statistic

	Optimal	Max Imme Rwd	Arbitrary 1	Arbitrary 2	Arbitrary 3
μ	8.05E+05	6.86E+05	6.07E+05	-2.72E+05	-9.32E+05
σ	3.85E+03	1.21E+04	5.28E+03	3.70E+03	4.12E+03

Figure 88: Total Rewards of Five Policies for Scenario 1

Table 48: Optimal Policy and Maximum Immediate Reward Policy for Scenario 2

Optimal Policy

State i	π_{i1}	π_{i2}	π_{i3}
1	1.00E+00	3.89E-11	0
2	1.00E+00	7.56E-11	0
3	1.00E+00	2.51E-10	0
4	1.00E+00	3.36E-11	0
5	1.00E+00	4.02E-11	0
6	1.00E+00	4.08E-09	0
7	4.60E-02	0	9.54E-01
8	3.49E-01	0	6.51E-01
9	5.80E-01	0	4.20E-01

State i	π_{i1}	π_{i2}	π_{i3}
1	1.00E+00	1.22E-10	0
2	1.00E+00	7.07E-11	0
3	1.00E+00	4.07E-10	0
4	1.00E+00	1.01E-10	0
5	1.00E+00	2.58E-11	0
6	7.99E-01	2.01E-01	0
7	4.96E-09	0	1.00E+00
8	3.62E-09	0	1.00E+00
9	6.71E-10	0	1.00E+00

State i	π_{i1}	π_{i2}	π_{i3}
1	1.00E+00	9.66E-11	0
2	1.00E+00	5.79E-11	0
3	1.00E+00	4.96E-10	0
4	1.00E+00	9.47E-08	0
5	1.00E+00	5.71E-11	0
6	5.73E-01	4.27E-01	0
7	1.06E-09	0	1.00E+00
8	1.33E-09	0	1.00E+00
9	6.83E-10	0	1.00E+00

Maximum Immediate Reward Policy

State i	π_{i1}	π_{i2}	π_{i3}
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0
5	1	0	0
6	0	1	0
7	0	0	1
8	0	0	1
9	1	0	0

State i	π_{i1}	π_{i2}	π_{i3}
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0
5	1	0	0
6	0	1	0
7	0	0	1
8	0	0	1
9	1	0	0

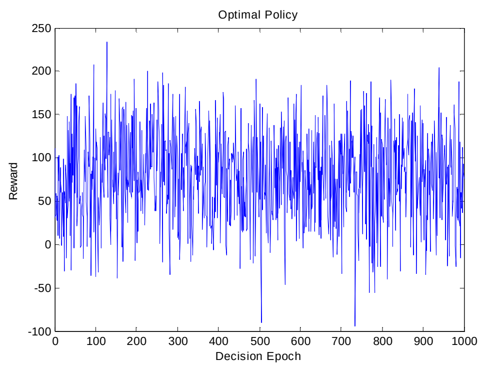
State i	π_{i1}	π_{i2}	π_{i3}
1	1	0	0
2	1	0	0
3	0	1	0
4	1	0	0
5	1	0	0
6	0	1	0
7	0	0	1
8	0	0	1
9	1	0	0

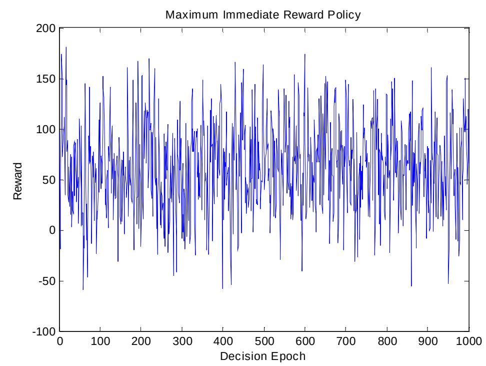
For agents with high mission priority

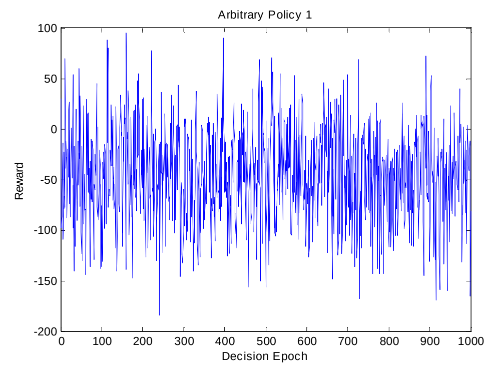
For agents with mid mission priority

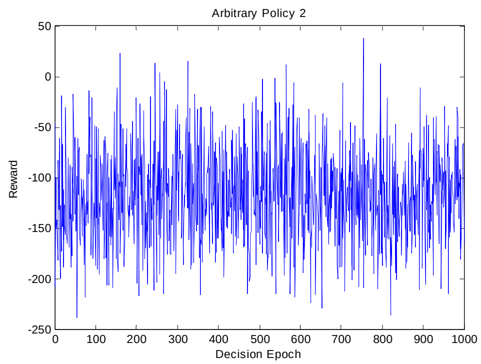
For agents with low mission priority

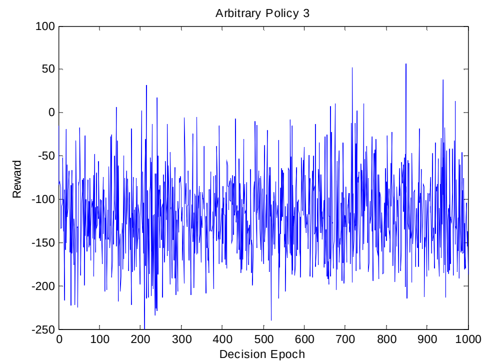
Starting State																
Agent	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
State	8	2	7	5	9	9	6	6	9	9	6	2	8	3	9	5





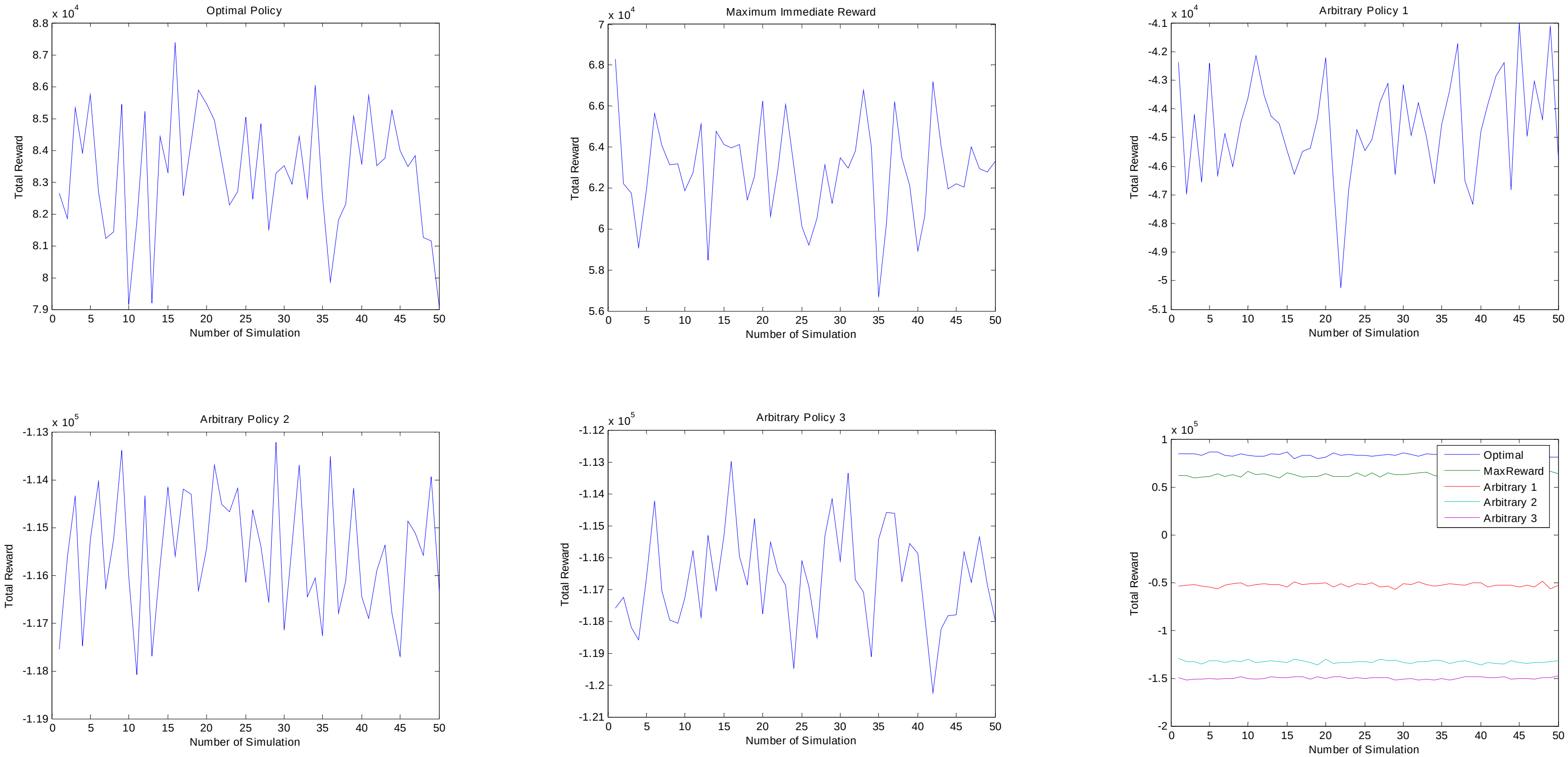






Policy	Reward Statistic μ	σ
Optimal	86.843	48.6576
Max Immediate Reward	64.046	43.1143
Arbitrary 1	-43.525	47.0374
Arbitrary 2	-114.421	47.4634
Arbitrary 3	-118.137	45.9696

Figure 89: Reward Trajectories of Five Policies for Scenario 2



Total Reward Statistic

	Optimal	Max Imme Rwd	Arbitrary 1	Arbitrary 2	Arbitrary 3
μ	8.37E+04	6.29E+04	-4.50E+04	-1.15E+05	-1.16E+05
σ	1.91E+03	2.02E+03	1.85E+03	1.42E+03	1.71E+03

Figure 90: Total Rewards of Five Policies for Scenario 2

REFERENCES

[1] Adelman, L. *Evaluating Decision Support and Expert Systems*. New York, John Wiley, 1992.

[2] Altman, E. *Constrained Markov Decision Processes*, Chapman & Hall/CRC, 1999.

[3] Anderson, D. and McNeill, G. Artificial Neural Networks Technology. from http://www.dacs.dtic.mil/techs/neural/neural4.html, 1992. [Date Accessed: June, 2005].

[4] Andriole, S. J. *Handbook for the Design, Development, Evaluation and Application of Interactive Decision Support Systems*. NJ, Petrocelli, Princeton, 1989.

[5] Ang, A. H. and Tang, W. H. "Probability Concepts in Engineering Planning and Design". *Decision, Risk, and Reliability*. Vol. 2: John Wiley & Sons, Inc. , 1984.

[6] Asimow, M. *Introduction to Design*. NJ, Prentice-Hall, Inc., Englewood Cliffs, 1962.

[7] Atallah, M. J. *(Ed.). Algorithms and Theory of Computation Handbook.* Boca Raton, CRC Press, 1999.

[8] Bacchus, F. "The 2000 Ai Planning System Competition". *Artificial Intelligence Magazine*. Vol. 22(3), 2001.

[9] Baker, A. P. "The Role of Mission Requirements, Vehicle Attributes, Technologies and Uncertainty in Rotorcraft System Design". Ph.D Thesis, Georgia Institute of Technology, 2002

[10] Bandte, O. "A Probabilistic Multi-Criteria Decision Making Technique for Conceptual and Preliminary Aerospace System Design". Ph.D. Thesis, School of Aerospace Engineering, Georgia Institute of Technology, 2000

[11] Barbera, S., Hammond, P. J. and Seidl, C. *Handbook of Utility Theory*, Kluwer Academic Publishers, 1998.

[12] Beck, J. R. and Pauker, S. G. "He Markov Process in Medical Prognosis". *Medical Decision Making*. Vol. 3(4): pp. 419--458, 1983.

[13] Bell Helicopter Textron. from http://www.bellhelicopter.textron.com, 2004. [Date Accessed: September, 2004].

[14] Bellman, R. "Dynamic Programming". *Princeton University Press*: Princeton, N. J, 1957.

[15] Bernoulli, D. "Exposition of a New Theory on the Measurement of Risk, Econometrica". *Econometrica*. 22(1): pp. 23-36, 1954.

[16] Blum, A. L. and Furst, M. L. "Fast Planning through Planning Graph Analysis". *Artificial Intelligence*. Vol. 90(1): pp. 281-300, 1997.

[17] Blumrich, J. F. "Design". *Science*. Vol. 168: pp. 1551-1554, 1970.

[18] Bore, N. K. and Mavris, D. N. "Formulation of a Multi-Mission Sizing Methodology for Competing Configurations". 42nd Aerospace Sciences Meeting and Exhibit, AIAA-2004-0535, American Institute of Aeronautics and Astronautics, 2004.

[19] Boss, R. W. What Is an Expert System? Eric Digest. from http://www.ericdigests.org/pre-9220/expert.htm, 1991. [Date Accessed: August, 2005]

[20] Boutilier, C. "Sequential Optimality and Coordination in Multiagent System". in Proceeding of the 1999 International Joint Conference on Artificial Intelligence, 1999.

[21] Boutilier, C., Brafman, R. and Geib, C. "Prioritized Goal Decomposition of Markov Decision Processes: Towards a Synthesis of Classical and Decision Theoretic Planning". in Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, 1997.

[22] Braham, J. "Inventive Ideas Grow with 'Triz.'". *Machine Design*. Vol. 67: 18, 1995.

[23] Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. *Classification and Regression Trees*. New York, Chapman & Hall, 1984.

[24] Cassandra, A. R. Partially Observable Markov Decision Processes. from http://www.cassandra.org/pomdp/index.shtml, 2003. [Date Accessed: March, 2006].

[25] Chapple, M. Data Mining: An Introduction. as of 2006, from http://databases.about.com/od/datamining/a/datamining.htm, 2006. [Date Accessed: May, 2006].

[26] Charnes, A. and Cooper, W. W. "Goal Programming and Multiple Objective Optimization - Part 1". *European Journal of Operational Research 1*: pp. 39-54, 1977.

[27] Chen, W., Sahai, A., Messac, A. and Sundararaj, G. "Physical Programming for Robust Design". 40th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit, AIAA-99-1206, American Institute of Aeronautics and Astronautics, 1999.

[28] Chi, R. T. and Turban, E. "Distributed Intelligent Executive Information Systems, Decision Support Systems". Vol. 14(2): pp. 117-130, 1995.

[29] Curry, B. and Moutinho, L. "Expert Systems and Marketing Strategy: An Application to Site Location Decision". *Journal of Marketing Channels*. Vol.1(1): pp. 23-37, 1991.

[30] D'Epenoux, F. "A Probabilistic Production and Inventory Problem". *Management Science*. Vol. 10(1): pp.98-108, 1963.

[31] Deb, K. *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, Wiley & Sons, 2001.

[32] Defense Acquisition University Press. *Systems Engineering Fundamentals*. Fort Belvoir, Virginia, 2001.

[33] DeLaurentis, D. A. "A Probabilistic Approach to Aircraft Design Emphasizing Stability and Control Uncertainties". PhD Thesis, Georgia Institute of Technology, 1998

[34] DeLaurentis, D. A., Kang, T., Lim, C., Mavris, D. N. and Schrage, D. P. "System-of–Systems Modeling for Personal Air Vehicles". 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 2004-6205, American Institute of Aeronautics and Astronautics, 2002.

[35] Dieter, G. E. *Engineering Design*. Boston, McGraw-Hill, 2000.

[36] DoD. "Dod Guide to Integrated Product and Process Development, Systems Engineering, Office of the under Secretary of Defense (Acquisition and Technology)". Washington, D.C., 1996.

[37] Dolgov, D. A. and Durfee, E. H. "Optimal Resource Allocation and Policy Formulation in Loosely-Coupled Markov Decision Processes". in Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling, 2004.

[38] Edwards, W. "How to Use Multiattribute Utility Measurement for Social Decision Making". *IEEE Transactions on Systems, Man, and Cybernetics,*. SMC-7: pp. 326-340, 1977.

[39] Edwards, W. and Barron, F. H. "Smarts and Smarter: Improved Simple Methods for Multiattribute Utility Measurements". *Organizational Behavior and Human Decision Processes*. 60: pp. 306-325, 1994.

[40] Engelmore, R. S. and Feigenbaum, E. Knowledge-Based System in Japan. from http://www.wtec.org/loyola/kb/c1_s1.htm, 1993. [Date Accessed: July, 2005].

[41] Evans, G. W. "An Overview of Techniques for Solving Multiobjective Mathematical Problems". *Management Science*. Vol. 30(6): pp 1268 – 1282, 1984.

[42] Fairmount Automation. Automated Damage Control System Development, Chilled Water Reduce-Scale Advanced Demonstrator (Cw-Rsad), from http://www.fairmountautomation.com/Services/AutomatedDamageControl_RSAD.htm, 2006. [Date Accessed: August, 2006].

[43] Feasible Solution. from http://www.nist.gov/dads/HTML/feasiblesltn.html, 2005. [Date Accessed: July, 2005].

[44] Feigenbaum, E., Friedland, P. E., Johnson, B. B., Nii, H. P., Schorr, H., Shrobe, H. and Engelmore, R. "Knowledge-Based Systems in Japan". *Report of the JTEC Panel*, 1993.

[45] Fielding, J. P. *Introduction to Aircraft Design*. New York, Cambridge University Press, 1999.

[46] Fonseca and L, G. Bernoulli and St. Petersburg Paradox., from http://cepa.newschool.edu/het/essays/uncert/bernoulhyp.htm, 2006. [Date Accessed: March, 2006].

[47] Frawley, W., G., P.-S. and Matheus, C. "Knowledge Discovery in Databases: An Overview". *Knowledge Discovery in Databases*. AAAI/MIT Press: pp. 1-27, 1991.

[48] Garcia, E. "Development of a Framework for Assessment of Capability and Throughput with the National Airspace System". Ph.D Thesis, Georgia Institute of Technology, 2002

[49] Garvey, P. R. *Probability Methods for Cost Uncertainty Analysis: A Systems Engineering Perspective*. New York, Marcel Dekker, Inc., 2000.

[50] Geffner, H. "Perspectives on Artificial Intelligence Planning". in Proceedings Eighteenth National Conference on Artificial Intelligence 2002.

[51] Gershon, M. and Duckstein, L. "A Procedure for Selection of a Multiobjective Technique with Application to Water and Mineral Resources". *Applied Mathematics and Computation*. Vol. 14: pp. 245-271, 1984.

[52] Gonzalez, C. "Decision Support for Real-Time, Dynamic Decision-Making Tasks". *Organizational Behavior and Human Decision Processes*. 96: pp. 142-154., 2005.

[53] Groen Brothers Aviation. from http://www.groenbros.com, 2004. [Date Accessed: June, 2004].

[54] Hazelrigg, G. A. *System Engineering: An Approach to Information-Based Design*. New Jersey, Prentice Hall, Inc.Upper Saddle Rive, 1996.

[55] Hazelrigg, G. A. "Validation of Engineering Design Alterative Selection Methods". *Engineering Optimization*. Vol. 35(2): pp. 103-120, 2003.

[56] Hazen, G. B. "Stochastic Trees: A New Technique for Temporal Medical Decision Modeling". *Medical Decision Making*. Vol. 12, 1992.

[57] Hobbs, B. J. "What Can We Learn from Experiments in Multiobjective Decision Analysis?" *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. SMC-16(3): pp 384-394, 1986.

[58] Hollenberg, J. P. "Markov Cycle Trees: A New Representation for Complex Markov Processes". *Medical Decision Making*. Vol. 4: 529, 1984.

[59] Hollingsworth, P. "Requirements Controlled Design: A Method for Discovery of Discontinuous System Boundaries in the Requirements Hyperspace". Ph.D Thesis, Georgia Institute of Technology, 2004

[60] Howard, R. A. "Dynamic Programming and Markov Processes". *MIT Press*: Cambridge, MA, 1960.

[61] Howe, D. *Aircraft Conceptual Design Synthesis*, Professional Engineering Publishing Limited, London, 2000.

[62] Hwang, C.-L. and Masud, A. S. M. *Multiple Objective Decision Making – Mehtods and Applications (Lecture Notes in Economics and Mathematical System; 164)*. New York, Springer Verlag, Berlin, Heidelberg, 1979.

[63] Hwang, C. L. and Yoon, K. *Multiple Attribute Decision Making Methods and Applications: A State of the Art Survey*, New York: Springer-Verlag, 1981.

[64] Inference Engine. from http://www.rt-sys.com/ie.htm, 2005. [Date Accessed: July, 2005].

[65] Intelegen Inc. An Introduction to Neural Networks. from http://brain.web-us.com/brain/neural.html, 2005. [Date Accessed: July, 2005].

[66] Kaelbling, L. P., Littman, M. L. and Moore, A. W. "Reinforcement Learning: A Survey". *Journal of Artificial Intelligence Research*. Vol. 4: pp. 237-285, 1996.

[67] Keen, P. G. W. and Scoot-Morton, M. S. *Decision Support Systems: An Organizational Perspective, Reading,*, MA: Addison-Wesley, 1978.

[68] Keeney, R. L. and Raiffa, H. *Decision with Multiple Objectives: Preferences and Value Tradeoffs*. New York, John Wiley and Sons, Inc., 1976.

[69] Keeney, R. L. and Raiffa, H. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, Cambridge University Press, 1993.

[70] Kirby, M. "A Methodology for Technology Identification, Evaluation, and Selection in Conceptual and Preliminary Aircraft Design". Ph.D Thesis, Georgia Institute of Technology, 2001

[71] Kirby, M. R. and Mavris, D. N. "A Method for Technology Selection Based on Benefit, Available Schedule and Budget Resources". 2000 World Aviation Congress and Exposition, SAEAIAA 2000-01-5563, 2000.

[72] Kleijnen. *Statistical Tools for Simulation Practitioners*. New York, Marcel Dekker, 1987.

[73] Kroo, I. *Aircraft Design: Synthesis and Analysis*, Stanford, CA, Desktop Aeronautics, Inc., 2004.

[74] Leong, T. Y. "Dynamic Decision Modeling in Medicine: A Critique of Existing Formalisms". in Proceedings 17th Annual Symposium on Computer Applications in Medical Care, 1993.

[75] Leong, T. Y. "Multiple Perspective Reasoning". *Artificial Intelligence*. 105(1-2): pp. 209-261, 1998.

[76] Lewe, J.-H. "An Integrated Decision-Making Framework for Transportation Architectures: Application to Aviation Systems Design". Ph.D. Thesis, Georgia Institute of Technology, 2005

[77] Li, Y., Mavris, D. N. and DeLaurentis, D. A. "The Investigation of a Decision-Making Technique Using the Loss Function". 4th AIAA Aviation Technology, Integration and Operations (ATIO) Forum, AIAA-2004-6205, 2004.

[78] Linkov, I., Varghese, A., Jamil, S., Seager, T. P., Kiker, G. and Bridges, T. "Multi-Criteria Decision Analysis: A Framework for Structuring Remedial Decisions at the Contaminated Sites". *Comparative Risk Assessment and Environmental Decision Making*. Springer, New York: pp. 15-54, 2004.

[79] Little, J. D. C. "Models and Managers: The Concept of a Decision Calculus, Management Science". Vol. 16(8): pp. B466-B485, 1970.

[80] Lu, J., Quaddus, M. A., Poh, K. L. and Williams, R. "The Design of a Knowledge-Based Guidance System for an Intelligent Multiple Objective Decision Support System (Imodss)". in Proceedings of the 10th Australasian Conference on Information Systems, 1999.

[81] Luger, G. F. and Stubblefield, W. A. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, London, Addison-Wesley, 1998.

[82] MacCrimmon, K. R. "An Overview of Multiple Objective Decision Making". in J. L. Cochrane and L. M. Zeeny (eds) Multiple Criteria Decision Making, The University of South Carolina Press, pp. 7-31, 1973.

[83] Martin, R. The St. Petersburg Paradox. from http://plato.stanford.edu/entries/paradox-stpetersburg, 2004. [Date Accessed: August, 2005].

[84] Marx, W. J., Mavris, D. N. and Schrage, D. P. "Integrating Design and Manufacturing for the High Speed Civil Transport". 19th ICAS Congress / AIAA Aircraft Systems Conference, 94-10.8.4, International Council of the Aeronautical Sciences, 1994.

[85] Matsatsinis, N. F. and Y., S. *Intelligent Support Systems for Marketing Decisions*, Kluwer Academic Publishers, 2003.

[86] Mavris, D. N. and DeLaurentis, D. "Methodology for Examining the Simultaneous Impact of Requirements, Vehicle Characteristics, and Technologies on Military Aircraft Design". 22nd Congress of the International Council on the Aeronautical Sciences (ICAS), 2000-1.4.4, International Council of the Aeronautical Sciences, 2000a.

[87] Mavris, D. N. and DeLaurentis, D. A. "An Integrated Approach to Military Aircraft Selection and Concept Evaluation". 1st AIAA Aircraft Engineering, Technology, and Operations Congress, 95-3921, American Institute of Aeronautics and Astronautics, 1995.

[88] Mavris, D. N. and DeLaurentis, D. A. "A Probabilistic Approach for Examining Aircraft Concept Feasibility and Viability". *Aircraft Design*. Vol. 3(2): pp. 79-101, 2000b.

[89] Mavris, D. N., DeLaurentis, D. A., Bandte, O. and Hale, M. A. "A Stochastic Approach to Multi-Disciplinary Aircraft Analysis and Design". 36th Aerospace Sciences Meeting and Exhibit AIAA-98-0912, 1998.

[90] McDermott, D. "The 1998 Ai Planning Systems Competition". *Artificial Intelligence Magazine*. Vol. 21(2): pp. 35-56, 2000.

[91] Messac, A. "Physical Programming: Effective Optimization for Computational Design". *American Institute of Aeronautics and Astronautics journal*. Vol. 34(1): pp. 149-158 1996.

[92] Meuleau, N., Hauskrecht, M., Kim, K.-E., Preshkin, L., Kaelbling, L., Dean, T. and Boutilier, C. "Solving Very Large Weakly Coupled Markov Decision

Processes". in Proceedings of the Fifteenth National Conference on Arti cial Intelligence, 1998.

[93] Moore, A. W. Markov Systems, Markov Decision Processes, and Dynamic Programming. from http://www.autonlab.org/tutorials/mdp.html, 2005. [Date Accessed: February, 2006].

[94] NASA LaRC. Revolutionary Aerospace System Concepts Program, Nasa Langley Research Center (Nasa Larc). from http://rasc.larc.nasa.gov/rasc_new/RASC.html, 2002. [Date Accessed: March, 2003].

[95] Neufville, R. D. *Applied Systems Analysis: Engineering Planning and Technology Management*. New York, McGraw-Hill, 1990.

[96] Neural Network. from http://www.pcai.com/web/ai_info/neural_nets.html, 2002. [Date Accessed: April, 2005].

[97] Newell, A. and Simon, H. A. "GPS: A Program That Simulates Human Thought". *In Feigenbaum, E. A., and Feldman, J., eds., Computers and Thought*: New York, 279--293, McGraw-Hill, 1963.

[98] Noble, J. S. and Tanchoco, J. M. A. *Design for Economics*. New York., John Wiley & Sons, Inc., 1993.

[99] Ozernoy, M. V. "Choosing the “Best” Multiple Criteria Decision-Making Method". *Information Systems and Operational Research*. Vol. 30(2): pp. 159-171, 1992.

[100] Ozernoy, V. M. "A Framework for Choosing the Most Appropriate Discrete Alternative Multiple Criteria Decision-Making Method in Decision Support Systems and Expert Systems". *Toward Interactive and Intelligent Decision Support Systems*. Vol. 2 of Lecture notes in economics and mathematical systems: pp. 56-64, 1987.

[101] Parsaye, K. "Acquiring and Verifying Knowledge Automatically". *AI Expert*. Vol. 3(5): pp. 48-63, 1988.

[102] Poh, K. L. "A Knowledge-Based Guidance System for Multi-Attribute Decision Making". *Artificial Intelligence in Engineering*. Vol. 12(33): pp. 315-326, 1998.

[103] Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, John Wiley and Sons, 1994.

[104] Raiffa, H. "Decision Analysis". *Introductory Lectures on choices under uncertainty*: Addison-Wesley, 1968.

[105] Rich, E. *Artificial Intelligence*. Singapore, McGraw-Hill, 1983

[106] Robinson Helicopter Company. from http://www.robinsonheli.com/newscopter.htm, 2003. [Date Accessed: September, 2003].

[107] Roman, F., Rolander, N., Fernandez, G. M., Bras, B., Allen, J., Mistree, F., Chastang, P., Depince, P. and Bennis, F. "Selection without Reflection Is a Risky Business..." 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA-2004-4429, 2004.

[108] Saaty, T. L. *The Analytic Hierarchy Process*. New York, McGraw-Hill, 1988.

[109] Saaty, T. L. *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. New York, McGraw-Hill, 1980.

[110] Sage, A. P. *Decision Support Systems Engineering*. New York, John Wiley & Sons, 1991.

[111] Scheidt, D. H. "Intelligent Agent-Based Control". *Johns Hopkins APL Technical Digest*. Vol. 23(4): pp. 383-395, 2002.

[112] Schoen, H., Rosenstein, H., Stanzione, K. A. and S., W. J. "User’s Manual for Vascomp Ii". NASA Ames Research Center,

[113] Schrage, D. P., Mavris, D. N. and Wasikowski, M. "Gtpdp - a Rotary Wing Aircraft Preliminary Design and Performance Estimation Program Including Optimization and Cost". AHS Forum, Vertical Lift Aircraft Design Conference, 1986.

[114] Sen, P. and Yang, J. B. *Multiple Criteria Decision Support in Engineering Design*, Springer-Verlag, London, 1998.

[115] Simon, H. A. *The New Science of Management Decision , Harper and Row*. New York, 1960.

[116] Singh, S. and D., C. "How to Dynamically Merge Markov Decision Processes, Advanced in Neural Information Processing Systems". *Advances in Neural Information Processing Systems*. Vol. 10, 1998.

[117] Sprague, R. H. and Carlson, E. D. *Building Effective Decision Support Systems*, Prentice Hall, Englewood Cliffs, NJ, 1982.

[118] Stergiou, C. and Siganos, D. Neural Networks. from http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#Contents, 2005. [Date Accessed: July, 2005].

[119] Sullivan, L. P. "Quality Function Deployment". *Quality Progress*. Vol. 19(6): pp. 39-50, 1986.

[120] Sutton, R. S. and Barto, A. G. "Reinforcement Learning: An Introduction". *The MIT Press*: Cambridge, Massachusetts London, England, 1998.

[121] Tate, D. "A Roadmap for Decomposition: Activities, Theories, and Tools for System Design". Ph.D Thesis, Department of Mechanical Engineering, MIT, 1999

[122] Tecle, A. and Duckstein, L. "A Procedure for Selecting Mcdm Techniques for Forest Resources Management". in Multiple Criteria Decision Making Proceeding of the Ninth International Conference: Theory and Applications in Business, Industry, and Government, New York, Springer-Verlag, 1992.

[123] U.S. Army. *Army Programs Decision Risk Analysis (Dra)*, Fort Lee, VA, 1990.

[124] Walks, J. P. and Mearman, J. F. "Integrated Engineering Plant". *2005 Distributed Intelligence for Automated Survivability (DIAS) Program Review*. Northrop Grumman Corporation, 2005.

[125] Webster's New World Dictionary of American English. New York, Prentice Hall, 1996.

[126] Wiki. Markov Decision Process. from http://en.wikipedia.org/wiki/Markov_decision_process, 2005. [Date Accessed: July, 2005].

[127] Xuan, P., Lesser, V. and Zilberstein, S. "Formal Modeling of Communication Decisions in Cooperative Multiagent Systems". in Proceedings of the Second Workshop on Game-Theoretic and Decision-Theoretic Agents, 2000.

[128] Zeleny, M. *Multiple Criteria Decision Making*. New York, McGraw-Hill, 1982.

VITA

Yongchang Li was born in Tongchuan, Shannxi, P.R. China in 1976. In 1993 he attended Beijing University of Aeronautics and Astronautics (BUAA) where he received his B.S. in Aerospace Engineering. He joined Aerospace System Design Lab at Georgia Institute of Technology in 2001 and received his M.S. in Aerospace Engineering in December of 2002.