INTEGER PROGRAMMING APPROACHES FOR SOME NON-CONVEX AND STOCHASTIC OPTIMIZATION PROBLEMS

A Thesis
Presented to
The Academic Faculty

by

James R. Luedtke

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology
December 2007

INTEGER PROGRAMMING APPROACHES FOR SOME NON-CONVEX AND STOCHASTIC OPTIMIZATION PROBLEMS

Approved by:

Professor Shabbir Ahmed, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor George Nemhauser, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Bill Cook
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor R. Gary Parker
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Zonghao Gu
Ilog, Inc.

Date Approved: 25 July 2007

To my wife,

Leigh,

And to our son,

Rowan

ACKNOWLEDGEMENTS

First and foremost, I express the deepest thanks to my advisors Prof. Shabbir Ahmed and Prof. George Nemhauser. I believe they went above and beyond in supporting my research, and in allowing me flexibility to explore ideas that were interesting to me. In addition, they were extremely helpful in my successful job search. I also want to thank a number of other faculty at Georgia Tech. Thanks to Prof. Gary Parker, who served as an academic advisor to me throughout my Georgia Tech career, and who also serves on my dissertation committee. Thanks also to the remaining members of my committee, Prof. Bill Cook and Dr. Zonghao Gu. Thanks to Prof. Chip White for advising me for my first couple of years at Georgia Tech. My time working with Prof. White has left a strong impression on my research perspective. Special thanks to Prof. Alex Shapiro for volunteering to help in my job search and for giving me insightful comments and suggestions on my research. I know that the respect prospective employers have for Prof. Shapiro's opinion was a great help for me. Thanks also to Prof. Anton Kleywegt for giving me a good recommendation and for teaching me the secrets of revenue management. I have had tremendous opportunities to learn from so many excellent teachers at Georgia Tech, and I want to specifically thank Prof. Martin Savelsbergh, Prof. Arkadi Nemirovski, and Prof. Doron Lubinsky (in addition to already mentioned Profs. Ahmed, Cook, Nemhauser, and Shapiro) for offering courses which I found particularly valuable.

Of course, I have also had amazing support from my fellow students during my time at Georgia Tech. From helping me figure out CPLEX, Linux, Latex, C, C++, Python, etc., to letting me ramble on about my ridiculous research ideas, to working with me on class projects, to giving me feedback on presentations, to just being willing to put up with me as an office mate, neighbor or office mate of someone else I was bothering, I thank (in alphabetical order): Deniz Dogan, Faram Engineer, Daniel Espinoza, Yun Fong, Ricardo Fukasawa, Renan Garcia, Marcos Goycoolea, Yongpei Guan, Yetkin Ileri, Brad

Jones, Rajeev Namboothiri, Jerry O'Neal, Amandeep Parmar, Anuj Shah, Martin Smith, Sriram Subramanian, Tolga Tezcan, and Juan Pablo Vielma. And if you are reading this looking for your name, I am obviously an idiot and forgot to put it here, so I also thank you, an especially thank you for forgiving me for forgetting to put your name.

Finally, I extend my utmost gratitude to the Achievement Rewards for College Scientists (ARCS) Atlanta chapter for supporting me for five years with a graduate fellowship, and especially to Betsy and Harald Hansen for funding the fellowship I received.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

In this dissertation we study several non-convex and stochastic optimization problems. The common theme is the use of integer programming techniques including valid inequalities and reformulation to solve these problems.

We first study a strategic planning model in which the activities to be planned, such as production and distribution in a supply network, require technology to be installed before they can be performed. The technology improves over time, so that a decision-maker has incentive to delay starting an activity to take advantage of better technology and lower operational costs. The model captures the fundamental trade-off between delaying the start-time of an activity and the need for some activities to be performed now. Models of this type are used in the oil industry to plan the development of oil fields. This problem is naturally formulated as a mixed-integer program with a bilinear objective. We develop a series of progressively more compact mixed-integer linear formulations, along with classes of valid inequalities that make the formulations strong. We also present a specialized branch-and-cut algorithm to solve an extremely compact concave formulation. In addition, we study a special case in which the activities are constrained to be nondecreasing over time, a constraint that was present in the motivating application. Computational results indicate that these formulations can be used to solve large-scale instances.

We next study methods for optimization with joint probabilistic constraints. A probabilistic constraint arises from a model in which data in the constraints is random, and a decision-maker wishes to obtain a solution which satisfies the random constraints with a given high probability. One difficulty in solving such probabilistic programs is that evaluating feasibility requires multi-dimensional integration, and hence is difficult in general. We propose to address this difficulty by solving approximation problems based on Monte Carlo samples of the random data. We demonstrate that this scheme can be used to yield both feasible solutions and lower bounds. We conduct preliminary computational tests of the

sampling approach which indicate that it can be used to yield good feasible solutions and reasonable bounds on their quality. We next discuss how the non-convex sample approximation problem can be solved when only the right-hand side is random by using mixed-integer programming (MIP). We give a MIP formulation and study the relaxation corresponding to a single row of the probabilistic constraint. We obtain two strengthened formulations. As a byproduct of this analysis, we obtain new results for the previously studied mixing set, subject to an additional knapsack inequality. We present computational results which indicate that by using our strengthened formulations, instances that are considerably larger than have been considered before can be solved to optimality.

Finally, we study stochastic programming problems under stochastic dominance constraints. A stochastic dominance constraint states that a random outcome which depends on the decision variables should stochastically dominate a given reference random variable. Such constraints allow for flexible management of risk in an optimization framework. We present new formulations for both first and second order stochastic dominance which are significantly more compact than existing formulations. In addition, for the non-convex first order constraint, we present a specialized branching scheme and heuristics that can be used with the new formulation. We conduct computational tests which illustrate the benefits of the new formulations.

CHAPTER I

INTRODUCTION

1.1 Background

In this dissertation, we study several optimization problems in which the feasible region or the objective is not convex. A classical example of such a non-convex optimization problem is a mixed-integer programming (MIP) problem given by

$$\begin{aligned} \min \quad & cx + hy \\ \text{s.t.} \quad & Ax + Gy \geq b \\ & x \in \mathbf{R}^n_+, y \in \mathbb{Z}^p_+ \end{aligned} \tag{1}$$

where $\mathbf{R}^n_+$ is the set of nonnegative real n-dimensional vectors, and $\mathbb{Z}^p_+$ is the set of non-negative integral p-dimensional vectors. The vectors x and y represent the decisions to be made. An instance of problem (1) is specified by the data (c, h, A, G, b) where $c \in \mathbf{R}^n$ and $h \in \mathbf{R}^p$ specify the costs of the decisions, A is an m by n matrix, G is an m by p matrix and $b \in \mathbf{R}^m$. The data (A, G, b) specify a set of linear inequalities the decision vectors must satisfy. The set $S = \{x \in \mathbf{R}^n_+, y \in \mathbb{Z}^p_+ : Ax + Gy \geq b\}$ is called the feasible region of (1). The feasible region of (1) is not convex because of the constraint that y take on only integer values.

A successful technique for solving non-convex optimization problems, and MIP problems in particular, is the *branch-and-cut* method, which we will briefly describe. See, for example, [50] for a much more extensive discussion of branch-and-cut and other algorithms for solving MIP problems.

We first discuss the *branch-and-bound* method and then discuss how the bounds used in this method can be improved using valid inequalities and reformulation. The branch-and-cut method is basically a branch-and-bound method in which the bounds are improved using valid inequalities.

1.1.1 Branch-and-Bound

Branch-and-bound is an enumeration scheme in which we search for the optimal solution by progressively dividing the feasible region into smaller subregions. In doing so, a search tree, also known as a branch-and-bound tree, is constructed where each node in the tree corresponds to a particular subregion of the feasible region. The idea is to avoid complete enumeration by calculating a lower bound on the value of the best possible solution in the subregion at each node. If this lower bound exceeds the objective value of a solution we have already found, then we do not need to explore this subregion any further. Algorithm 1 is a generic version of the branch-and-bound method to solve a general problem of the form

$$z^* = \min\{f(x) : x \in S\}.$$

The algorithm maintains an upper bound ub on the optimal value z^* and a set of nodes

Algorithm 1: Generic Branch-and-bound Algorithm

```
1  Set Nodes = {(S, −∞)}, ub = +∞;
2  while Nodes ≠ ∅ do
3  |  Choose (F, lb) ∈ Nodes;
4  |  Let hub = UpperBound(F);
5  |  if hub < ub then
6  |  |  Set ub = hub;
7  |  |  Set Nodes = {(F′, lb′) ∈ Nodes : lb′ < ub};
8  |  end
9  |  Let clb = LowerBound(F);
10 |  if clb < ub then
11 |  |  Choose F1 and F2 such that F = F1 ∪ F2;
12 |  |  Set Nodes = Nodes ∪ {(F1, clb), (F2, clb)};
13 |  end
14 end
```

Nodes where each node is specified by a subset F of the feasible region S and a lower bound lb which satisfies $f(x) \geq \mathsf{lb}$ for all $x \in F$. An iteration of the algorithm begins by selecting a node (F, lb) in Nodes. In line 4 a function $\texttt{UpperBound}(F)$ is called, which attempts to find an upper bound on the minimum cost of any solution $x \in F$. Typically an upper bound is obtained by running a heuristic which attempts to find low cost feasible solution $x \in F$. If the heuristic succeeds, $\texttt{UpperBound}(F)$ returns the objective cost of this solution $f(x)$,

otherwise it returns $+\infty$. If a feasible solution with objective value less than the current upper bound ub is found, the upper bound is updated in line 6, and the nodes in Nodes which have a lower bound as large as the new upper bound are discarded. Next, the function `LowerBound`(F) is called, which returns a lower bound clb which satisfies $\mathsf{clb} \leq f(x)$ for all $x \in F$. The function `LowerBound`(F) may also detect that $F = \emptyset$, in which case it returns $+\infty$. If $\mathsf{clb} \geq \mathsf{ub}$, there can be no solution in F which has an objective better than the current best value we have, and hence we do not explore this subregion further. Otherwise, in lines 11 and 12 the algorithm divides the region F into two subregions and adds these to Nodes. The algorithm terminates when there are no nodes left in Nodes, after which it reports that ub is the optimal value if it is finite, otherwise it reports that $S = \emptyset$.

To apply the generic branch-and-bound algorithm to solve a specific problem, one must specify:

- **Node selection:** how to choose a node in Nodes (line 3),
- **Upper bound:** how to implement the function `UpperBound`(F) (line 4),
- **Lower bound:** how to implement the function `LowerBound`(F) (line 9), and
- **Branching:** how to divide a feasible subregion F (line 11).

As an example, we discuss typical options for each of these tasks for branch-and-bound applied to MIP problems (1). Two common node selection strategies are *depth-first* and *best bound*. In the depth-first strategy, one of the two most recently created nodes is selected. In the best bound strategy, a node which has the least lower bound is selected.

In mixed-integer programming, the lower and upper bound functions can be accomplished together by solving the linear programming (LP) *relaxation*

$$\begin{aligned} \min \quad & cx + hy \\ \text{s.t.} \quad & Ax + Gy \geq b \\ & x \in \mathbf{R}^n_+, y \in \mathbf{R}^p_+ \end{aligned} \tag{2}$$

where the constraint $y \in \mathbb{Z}^p_+$ has been *relaxed* to $y \in \mathbf{R}^p_+$. Because any solution feasible to (1) is also feasible to (2), the optimal solution to (2) yields a lower bound to the optimal

solution of (1). Solving the LP relaxation can also serve as an attempt to find an upper bound, since if the optimal solution (x^*, y^*) happens to have $y^* \in \mathbb{Z}^p_+$, then (x^*, y^*) is a feasible solution to (1). If $y^* \notin \mathbb{Z}^p_+$, then the upper bound function failed and reports $+\infty$ as the upper bound. In mixed-integer programming, the feasible subregion F at a node is typically specified by a set of additional linear inequalities, say $By \geq d$, applied to the integer variables. That is, $F = \{x \in \mathbf{R}^n_+, y \in \mathbb{Z}^p_+ : Ax + Gy \geq b, By \geq d\}$. Thus, the lower bound for a node with feasible subregion F specified in this manner can be calculated by solving the *node* LP relaxation

$$\begin{aligned} \min \quad & cx + hy \\ \text{s.t.} \quad & Ax + Gy \geq b \\ & By \geq d \\ & x \in \mathbf{R}^n_+, y \in \mathbf{R}^p_+. \end{aligned}$$

Branching in mixed-integer programming is typically done by specifying constraints on the integer variables. Most often, the constraints added are just bounds on the integer variables. A desirable property of a branching strategy is that the optimal solution (x^*, y^*) for the relaxation at the current node should not be feasible to the relaxations of either of the subdivisions created from this node. This can be accomplished by branching on integer variables by selecting a variable, say y_j, which has a fractional value $y^*_j = f$ in the optimal solution of the current relaxation. Then, the current feasible subregion F is divided into $F = F_1 \cup F_2$ with $F_1 = \{(x, y) \in F : y_j \leq \lfloor f \rfloor\}$ and $F_2 = \{(x, y) \in F : y_j \geq \lfloor f \rfloor + 1\}$ where $\lfloor f \rfloor = \max\{z \in \mathbb{Z} : z \leq f\}$. Note that the relation $F = F_1 \cup F_2$ is valid because y_j is integer for all $(x, y) \in F \subseteq S$. Also, in this case it happens that $F_1 \cap F_2 = \emptyset$, which is a nice property since it implies solutions in F will not be considered in multiple subregions, although it is not a necessary property.

The computational efficiency of a branch-and-bound algorithm is determined by how many iterations are performed (i.e., how many nodes are explored), and how long each iteration takes. The most time-consuming step in an iteration is typically the calculation of the lower bound. In mixed-integer programming this is usually done by solving the

LP relaxation, and hence the size of the formulation (number of rows and variables) will have an impact on how long each iteration takes. Assuming the lower bound can be calculated "sufficiently fast," the computational success of a branch-and-bound algorithm will be determined by how many nodes must be explored. In particular, it is crucial that the computed lower bounds be "tight," that is, they should be close to the true optimal value so that lower bound comparisons in lines 7 and 10 of the branch-and-bound algorithm will frequently allow subregions to be excluded from the search.

1.1.2 Improving the Lower Bounds

Because of the importance of the quality of the lower bounds used in a branch-and-bound algorithm, we now discuss two methods for improving the lower bounds for the case of mixed-integer programming: valid inequalities and reformulation.

Valid inequalities

An inequality $\alpha y \geq \alpha_0$, or (α, α_0), is called a *valid inequality* for a set S if it is satisfied by all points in S. By definition, if a valid inequality is added to the linear programming relaxation for a feasible region S, the LP will still be a relaxation of S. Since adding constraints to a linear program may increase the optimal objective, the lower bound obtained by the LP can be improved by adding valid inequalities. For example, suppose we have solved the LP relaxation of a pure integer version of (1) (in which the continuous variables x are not present), and obtain a solution y^* which is not integer. Then, we can attempt to find a valid inequality which the solution y^* does not satisfy, and hence by adding this inequality to the linear program we would "cut off" this solution. In this context, a valid inequality is often called a *cutting plane*. We then re-solve the linear program with this inequality added, and since the previous optimal solution is now infeasible, there is a chance the objective value will increase, leading to an improved lower bound.

Given a collection $\mathcal{F}$ of valid inequalities for S, and a relaxation solution y^*, the problem of checking whether there exists a valid inequality $(\alpha, \alpha_0) \in \mathcal{F}$ which cuts off y^* and finding one if one exists is known as the *separation* problem. The separation problem should be efficiently solvable in order to use valid inequalities from a class $\mathcal{F}$ to strengthen the lower

bounds from a linear programming relaxation.

The *convex hull* of a set S, denoted by $\text{conv}(S)$, is defined to be the set of all points which are convex combinations of points in S. That is, $x \in \text{conv}(S)$ if and only if there exists finitely many points $x^1, \ldots, x^k$ each in S and a vector $\lambda \in \mathbf{R}^k_+$ with $\sum_{i=1}^k \lambda_i = 1$ such that $x = \sum_{i=1}^k \lambda_i x^i$. When S is the feasible region of a mixed-integer program, $\text{conv}(S)$ is a polyhedron. If we knew a set of valid inequalities which described this polyhedron, then the mixed-integer program could be solved as a linear program. However, typically exponentially many inequalities are required to describe $\text{conv}(S)$ and so it would be necessary to separate these inequalities only as needed.

Unfortunately, separating over the convex hull of a set S is as hard as optimizing over the set S, and so it is not practical to solve hard optimization problems purely by adding valid inequalities. However, we can hope to find classes of inequalities which we can separate efficiently and which lead to improved lower bounds. A successful technique for generating valid inequalities to strengthen the linear programming relaxation is to generate inequalities based on a relatively simple substructure of the overall problem. A classic pioneering example of this technique is the use of cutting planes obtained from single knapsack rows of an integer program [19]. In this case, cutting planes are generated from constraints of the form

$$\sum_{i=1}^{p} a_i y_i \leq b$$

which appear in the integer programming formulation.

Another example of a relatively simple substructure, which is more closely related to the work in this dissertation, is the union of finitely many sets, known as a *disjunction*. For example, suppose in the case of mixed-integer programming that the feasible region $S = \{x \in \mathbf{R}^n_+, y \in \mathbb{Z}^p_+ : Ax + Gy \geq b\}$ includes a constraint of the form

$$\sum_{i \in I} y_i = 1.$$

Let $P = \{x \in \mathbf{R}^n_+, y \in \mathbf{R}^p_+ : Ax + Gy \geq b\}$ be the feasible region of the linear programming

relaxation of S. Then,

$$S = \bigcup_{i \in I} \{(x, y) \in S : y_i = 1\} \subseteq \bigcup_{i \in I} \{(x, y) \in P : y_i = 1\}.$$

Hence, valid inequalities for the set $D = \cup_{i \in I} \{(x, y) \in P : y_i = 1\}$ are also valid for S. Furthermore, since $|I| \leq p$, optimization over D can be done efficiently by separately optimizing over $|I|$ polyhedra. Thus, it is theoretically possible to separate over the set $\text{conv}(D)$ efficiently. In fact, it is possible to write an explicit linear program which solves the separation problem over $\text{conv}(D)$ [5]. This linear program is $|I|$ times larger than the original linear program, so separation with this approach may not be efficient when $|I|$ is fairly large. However, it may be possible to exploit problem specific structure to separate over the inequalities which define $\text{conv}(D)$ more efficiently, and this is an approach we employ often in this dissertation.

Reformulation

Given an optimization problem

$$\min\{f(x) : x \in S\} \tag{3}$$

the problem

$$\min\{g(w) : w \in W\} \tag{4}$$

is a *reformulation* of (3) if for all $x \in S$ there exists $w \in W$ with $g(w) \leq f(x)$ and conversely, for all $w \in W$ there exists $x \in S$ with $f(x) \leq g(w)$. Thus, if w^* is an optimal solution to (4), then there exists a solution x^* feasible to (3) which satisfies $f(x^*) \leq g(w^*)$. In fact, this solution must be optimal to (3) since for any $x \in S$ there exists $w \in W$ with $f(x) \geq g(w) \geq g(w^*) \geq f(x^*)$. Thus, the reformulation (4) can be used to solve (3).

The motivation for reformulation is that the reformulated problem (4) may be more amenable to solution by branch-and-cut. In particular, a reformulation may lead to a relaxation which yields a better lower bound or can be solved more efficiently. For example, in the second chapter of this dissertation we study a problem which is initially formulated as a mixed-integer program with bilinear objective. If we relax the integrality restrictions

in this formulation, we obtain a relaxation which is a non-convex bilinear program, and hence is not efficiently solvable. However, we are able to reformulate this problem as a mixed-integer program with linear objective, so that when the integrality restrictions are relaxed, a linear programming relaxation is obtained. In many cases, a reformulation which yields better relaxation lower bounds can be obtained by *adding* variables and constraints to the original formulation. Such a reformulation is sometimes called an *extended formulation*. A classical example of a reformulation of this type is the extended formulation for the uncapacitated lot sizing problem [7, 25, 55]. In this case, there is a trade-off between the possibly improved lower bound obtained from the reformulation relaxation and the increased time to obtain the lower bound due to the increased formulation size. In the example of the uncapacitated lot sizing problem with T periods, the extended formulation has $O(T^2)$ variables and constraints as opposed to $O(T)$ variables and constraints in the original formulation. However, the linear relaxation of the extended formulation already gives the convex hull of this formulation, whereas exponentially many valid inequalities are required to define the convex hull of the original formulation.

1.2 Dissertation Overview

In the second chapter of this dissertation we study a problem which has a non-convex objective. This problem arises from a strategic planning model in which the activities to be planned, such as production and distribution in a supply network, require technology to be installed before they can be performed. The technology improves over time, so that a decision-maker has incentive to delay starting an activity to take advantage of better technology and lower operational costs. The model captures the fundamental trade-off between delaying the start-time of an activity and the need for some activities to be performed now. Models of this type are used in the oil industry to plan the development of oil fields. This problem is naturally formulated as a mixed-integer program with a bilinear objective. We develop a series of progressively more compact mixed-integer linear formulations, along with classes of valid inequalities that make the formulations strong. Our approach in studying

these formulations is to consider the relaxation obtained from a single activity in the problem. The inequalities we derive for the different formulations are sufficient to define the convex hull of the single activity relaxation. We also present a specialized branch-and-cut algorithm to solve an extremely compact concave formulation. In addition, we study a special case in which the activities are constrained to be nondecreasing over time, a constraint that was present in the motivating application. We show how the formulations we have developed can be strengthened using this additional constraint, and in particular, we find that for the compact concave formulation a very strong formulation is obtained with the addition of just one valid inequality for each activity. Computational results indicate that these formulations can be used to solve large-scale instances, whereas a straightforward linearization of the mixed-integer bilinear formulation fails to solve even small instances. In addition, we find that due to its compactness, the concave formulation can be used to generate good feasible solutions and reasonable optimality guarantees for problems which are so large that just solving the linear programming relaxation of the mixed-integer programming formulations takes prohibitively long.

In the third chapter of the dissertation we study optimization problems with joint probabilistic constraints. A probabilistic constraint arises from a model in which data in the constraints is random, and a decision-maker wishes to obtain a solution which satisfies the random constraints with a specified (usually high) probability. Except in a few special cases, probabilistic constraints define a non-convex feasible region. In addition, solving problems with probabilistic constraints is difficult because checking feasibility requires evaluating a multi-dimensional integral. We propose to address this latter difficulty by solving approximation problems based on Monte Carlo samples of the random data. We demonstrate that the sample approximation problems we propose can be used to yield both feasible solutions and lower bounds. This is in contrast to recent approximations of probabilistic constraints [17, 51, 52] which yield feasible solutions, but no lower bounds, so that the conservatism of these solutions cannot be evaluated. The disadvantage of the sample approximation we propose is that the feasible region is non-convex, even if the deterministic feasible region is convex and the constraints which are enforced with high probability are linear. We conduct

preliminary computational tests of the sampling approach which indicate that the sample approximation problem can be used to yield good feasible solutions and reasonable bounds on the quality of these solutions.

The sample approximation scheme we have studied can be applied to a very wide range of problems with probabilistic constraints. However, the use of this scheme may be limited due to the difficulty of solving the non-convex approximation problem. We therefore discuss how the sample approximation problem can be solved in an important special case by using mixed-integer programming. In particular, we consider a probabilistically constrained program (PCP) in which only the right-hand side is random. We give a mixed-integer programming formulation for the sample approximation problem of PCP and study the relaxation corresponding to a single row of the probabilistic constraint. We use results on a previously studied mixed-integer set, known as a *mixing set* [3, 32, 48], to obtain two strengthened formulations. The first is strengthened with an exponential class of valid inequalities and the second is an extended formulation which is as strong as the first augmented with all these inequalities. As a byproduct of our analysis, we obtain new valid inequalities for both formulations of the mixing set subject to an additional knapsack inequality. For the extended formulation, the new class of inequalities we introduce characterizes the convex hull of the single row probabilistic constraint when all realizations of the right-hand side are equally likely, in which case the knapsack inequality simplifies to a cardinality constraint. We present computational results which indicate that by using our strengthened formulations, instances of probabilistically constrained linear programs that are considerably larger than have been considered before can be solved to optimality. In particular, this approach allows us to solve instances with a large number of possible realizations, so that the sample approximation problem for this special case can be solved even with a large sample size.

In the fourth and final chapter of this dissertation, we introduce new formulations for stochastic programming under stochastic dominance constraints. A stochastic dominance constraint states that a random outcome which depends on our decisions should stochastically dominate a given reference random variable. Such constraints allow for flexible management of risk in an optimization framework. The first type of stochastic dominance

constraint we consider is second order stochastic dominance, which defines a convex feasible region. For this case, we present a linear programming formulation which has the advantage that it has $O(N)$ rows, as opposed to $O(N^2)$ in an existing formulation [22], where N is the number of possible outcomes of the random inputs. Computational results indicate that, when using a commercial linear programming solver, large instances are solved two orders of magnitude faster using this new formulation than using the existing formulation.

The second type of stochastic dominance constraint we consider is first order stochastic dominance, which defines a non-convex feasible region. We present a mixed-integer programming formulation for first order stochastic dominance which has $O(N)$ rows, as opposed to $O(N^2)$ rows in an existing formulation [53, 54]. In addition, the relaxation of this formulation is a formulation for second order dominance, which has been shown to be a tight relaxation of first order dominance constraints [54]. We also develop a specialized branching strategy and specialized heuristics for use with the new formulation. Computational results indicate that this formulation can be used to find optimal and provably near-optimal solutions for large instances.

CHAPTER II

STRATEGIC PLANNING WITH START-TIME DEPENDENT VARIABLE COSTS

2.1 Introduction

We study a multiple period strategic planning model in which the activities to be planned, such as production at supply points, have technology dependent costs. Specifically, we assume technology must be installed before an activity can be performed, and that once installed it cannot be changed for the remainder of the planning horizon. In general, technology improves over time so that a decision maker has incentive to delay installation to benefit from reduced operating costs. On the other hand, immediate requirements such as customer demands may require some activity to be performed now. In this chapter of the dissertation we present a planning model that captures this basic trade-off.

This research was motivated by a strategic planning model in the upstream oil and gas industry in which a firm is planning the long-term development of oil fields and transportation modes between the fields and downstream processing facilities. This development involves the installation of facilities that cost billions of dollars. A generic version of this application consists of a simple production and distribution planning problem with constraints

$$\sum_{j \in J} v_{ijt} - x_{it} = 0 \qquad \forall i \in I, t \in \mathcal{T} \tag{5}$$

$$\sum_{i \in I} v_{ijt} = D_{jt} \quad \forall j \in J, t \in \mathcal{T} \tag{6}$$

$$0 \leq x_{it} \leq M_i \quad \forall i \in I, t \in \mathcal{T} \tag{7}$$

where $\mathcal{T} = \{1, \ldots, T\}$ is an index set of time periods, T is the planning horizon, I is an index set of production nodes with per-period capacity M_i for all $i \in I$ and J is an index set of demand nodes with demands D_{jt} for each $j \in J$ and $t \in \mathcal{T}$. The variables x_{it} represent the production quantities at production node i in time period t and the variables v_{ijt} represent

the distribution quantities from i to j in period t. In the motivating application, the time periods are years and the time horizon is twenty to thirty years, reflecting the long-term planning of very expensive investments. If the production and distribution costs are linear, the problem of minimizing costs to meet demand decomposes into T simple transportation problems. However, in this application the production and distribution costs are *start-time dependent*. That is, before we can produce at a node (or send flow on an arc), we must first install technology at that node, and the technology installed will determine the per unit cost of production at that node *over the entire planning horizon*. In addition, the period in which we install technology will determine the fixed cost for the installation, if any. Therefore, our planning problem becomes a question of if and when to install technology at each supply node and distribution arc, and given these decisions how much to produce at each node and distribute along each arc to minimize the total cost of meeting demand over the planning horizon.

A natural formulation of the problem introduces binary variables to model the decision of which period technology will be installed (which we refer to as the start period), and leads to a mixed-integer program with bilinear objective. We present a straightforward compact linearization of this formulation, which yields a mixed-integer linear formulation. However, this linearization yields poor lower bounds, and hence is not computationally useful. Another approach, due to Adams and Sherali [1], that has been used in solving general mixed-integer bilinear programs is to create a linearization of the formulation by defining new variables which represent all product terms in the objective. This approach can yield linearizations which yield very good lower bounds. Although the resulting linear program may be intractably large, they develop specialized Lagrangian relaxation methods to generate strong lower bounds without actually solving the full linear program. We follow a similar approach in attempting to generate strong linear formulations for our problem, but we exploit problem specific structure to generate strong formulations that are relatively small. In particular, we focus on the substructure arising from a single activity problem, and develop strong formulations for it.

The remainder of this chapter is organized as follows. We begin by reviewing some

related literature in Section 2.2. We present the problem definition in Section 2.3 and present our study of integer linear formulations for this problem in Section 2.4. This section may be considered a case study in formulation development for a discrete problem initially defined with a nonlinear objective. We begin with an extended formulation in Section 2.4.1, which is theoretically interesting because it is integral in the single activity case. However, we are interested in solving problems with a large number of activities and a long planning horizon, so that this formulation will be too large to use for solving large instances. Thus, in Section 2.4.2 we introduce a compact linearization of the mixed-integer bilinear program and show how it can be strengthened using ideas from the classical lot sizing problem. In Section 2.4.3 we derive an even more compact formulation in which no linearizing variables are introduced. The only variables are the activity levels and binary variables capturing the decision of which period will be the start period. In Section 2.5 we show how a concave minimization formulation of the problem can be solved in a branch-and-cut framework without adding even the binary variables, leading to a formulation with very small linear programming relaxations. We show in Section 2.6 how the formulations we have developed can be strengthened when an additional constraint that the activities be nondecreasing over time is present. We present computational results in Section 2.7 comparing the different formulations and testing the effectiveness of the valid inequalities developed for them. Finally, we make some concluding remarks and discuss directions for future research in Section 2.8.

2.2 Related Literature

The production and distribution planning problem described in the introduction can be interpreted as a dynamic facility location problem, where installing technology corresponds to opening a facility. In the dynamic facility location problem, we must select which facilities to open and when so as to meet demand over the horizon at minimum cost. Typically in these models, (see, e.g.,[66]), there is a fixed cost to open a facility, which depends on the period in which the facility is opened, and once opened, production at a facility incurs a variable cost which depends on the period in which *production occurs*. This is in contrast

to our model which assumes the variable cost depends on the period in which *the facility is opened*. This is indeed an added complication since, in the special case when there are no fixed costs, the standard dynamic facility location problem becomes a simple transportation problem, whereas our model remains NP-hard even with $T = 2$ (see Section 2.3). A notable exception is the model initially presented in [68] in which variable costs depend on the period in which the facility opens as well as on the period in which production occurs. This model is essentially an uncapacitated version of the extended formulation we present in Section 2.4.1. However, the authors then immediately assume the variable costs will not depend on the period in which the facility opens, and obtain a more compact formulation for which they develop a dual ascent method to solve. We are interested in solving problems in which the number of activities to be planned (or potential facilities to open) is large, so that the extended formulation is too large to be of practical use. Thus, our focus is to develop compact formulations which are still strong. We remark here that for the formulation we develop in Section 2.4.2 we will assume the variable costs depend only on the period in which the technology is installed and *not* on the period in which the production occurs, whereas the subsequent formulations can handle joint dependence of the variable costs on the start period and on the period in which activity occurs.

Another related area of work is the capacity expansion literature [46]. In [58], Rajagopalan presents a capacity expansion and replacement model which allows fixed and variable costs to depend on the period in which capacity is installed. The key assumption which enables the development of a strong formulation for this model is that capacity can be installed in small increments relative to demand, so that it is reasonable to assume a continuous amount of capacity can be installed to exactly satisfy demand. Furthermore, the model in [58] allows expansion to occur repeatedly, and allows older technology to be retired and replaced with newer technology. In contrast, our model is more restrictive; capacity can be installed only once, and is a discrete decision of whether to install or not. Then, given that capacity has been installed, there is a secondary continuous decision of how much to use. The model of [58] would be more appropriate if capacity represents, for example, computers or even trucks in a fleet; our model would be more appropriate if

capacity represents, for example, building a power plant or installing a pipeline. Another difference is that the model in [58] considers capacity expansion of only one type of capacity, or in our context, a single activity. Of course, with multiple activities, the formulation of [58] can still be used by combining the single activity formulations for each activity. However, in terms of methodology, [58] introduces a disaggregate, or extended formulation, of the model, which is similar in spirit to the extended formulation we will present in Section 2.4.1. Because we are interested in solving problems with a large number of activities, we seek more compact formulations and use the extended formulation only for theoretical purposes, whereas [58] use the extended formulation for computation. The paper [60] considers a capacity expansion problem in which the capacity expansion decision is assumed to be discrete as in our model, but it only captures the fixed costs of the capacity; the variable costs are assumed to be insignificant. This is in contrast to our model, in which the variable costs are significant. Other works in this area include [39], [44] and [59], all of which allow installation of continuous amounts of capacity. The paper [44] considers a multiple product model and focuses on the trade-off between installing capacity dedicated for each product type and flexible capacity capable of producing all products. The paper [59] is notable in that it includes a model in which technological improvements can be random, although it is not clear that it could be solved for reasonable sized problems.

2.3 Problem Definition and Complexity

We consider a multiple period planning model with time horizon T periods, and let $\mathcal{T} = \{1, \ldots, T\}$. We have a set A of unique activities to plan, with $|A| = m$. For each activity $a \in A$, we let x_{at} be a decision variable representing the activity level in period $t \in \mathcal{T}$. For each activity, the vector $x_a \in \mathbf{R}^T$ of activity levels over the horizon is constrained to be in a set $X_a \subseteq [0, M_a]^T$ where $M_a > 0$ represents an upper bound on the activity which, for simplicity, is common over all periods.

The activities to be planned are linked by additional variables and constraints which represent system demands that motivate activities to be performed. The linking variables are given by u_j for $j \in N$ for some index set N with $|N| = n$. The linking constraints are

specified by the set $C \subseteq \mathbf{R}^{n+Tm}$ so that any feasible solution (u, x) must satisfy $(u, x) \in C$. We assume for convenience that C is a polyhedron, but the approach can also be applied if C is described by a polyhedron subject to additional constraints such as integrality on some of the variables, so that it is natural to optimize over C using a branch-and-cut framework. In the motivating application the activities are production and distribution in a network and C is given by the network flow balance constraints and the requirement that demands must be met.

The primary complication, and the motivation for the present work, is the non-convex objective. Before an activity can be performed, an enabling technology must be installed. We refer to the period in which the technology is installed as the *start period*. Activity levels can only be positive in periods on or after the start period. If $s \in \mathcal{T}$ is the start period, then a fixed charge of $f_{as} \geq 0$ is incurred, and the variable cost of production in period $t \geq s$ is $c_{ast} \geq 0$. If an activity is never performed, then we take the start period, by definition, to be period $T + 1$. For each activity a we introduce a vector $y_a \in \{0, 1\}^T$ of binary decision variables where we have $y_{at} = 1$ if and only if period t is the start period of activity a. Since each activity can only be started once, each binary vector must satisfy

$$\sum_{t \in \mathcal{T}} y_{at} \leq 1. \tag{8}$$

Define $Y = \Big\{ y \in \{0, 1\}^T : (8) \Big\}$ and $Y^R = \big\{ y \in \mathbf{R}^T_+ : (8) \big\}$.

We state our strategic planning problem as the mixed-integer bilinear program

$$\begin{aligned} \text{(SP)} \qquad \min \quad & \sum_{a \in A} \sum_{s=1}^{T} \Big(f_{as} + \sum_{t=s}^{T} c_{ast} x_{at} \Big) y_{as} \quad + \sum_{j \in N} d_j u_j \\ \text{s.t.} \quad & x_{at} - \sum_{s=1}^{t} M_a y_{as} \leq 0 \qquad\qquad \forall a \in A, t \in \mathcal{T} \\ & (u, x) \in C, \ x_a \in X_a, \ y_a \in Y \quad \forall a \in A. \end{aligned} \tag{9}$$

The first term in the objective is bilinear, and captures the fixed and variable costs over the horizon for each activity. For an activity a, if period s is the start period, then $y_{as} = 1$ and the objective records the cost $f_{as} + \sum_{t=s}^{T} c_{ast} x_{at}$, correctly capturing the fixed and variable costs over the entire horizon for this activity. In addition to the costs of the activities, the

objective also includes a linear cost term $\sum_{j \in N} d_j u_j$ for the linking variables, where $d \in \mathbf{R}^n$. The constraints (9) ensure that for each activity there is no positive activity before its start period.

Our approach to developing formulations for this problem is to study the substructure corresponding to a single fixed activity a. Therefore, for the development of our theoretical results, we will suppress the index a. Thus, the substructure of interest which we study is simply

$$\text{(MIBL)} \qquad \min \quad \sum_{s=1}^{T} \Big(f_s + \sum_{t=s}^{T} c_{st} x_t \Big) y_s \tag{10}$$

$$\text{s.t.} \quad x_t - \sum_{s=1}^{t} M y_s \leq 0 \qquad \forall t \in \mathcal{T} \tag{11}$$

$$x \in X, \; y \in Y.$$

Of course, the optimization problem MIBL itself is trivial since it is optimal to perform no activity at a cost of zero. However, by developing strong formulations for MIBL, and for each activity including this in the formulation for SP, we achieve a strong formulation for SP, in which the additional constraints require activity to be performed.

Complexity

We now show that with start-time dependent variable costs, the motivating application, presented in the introduction, is NP-hard. In the presence of fixed costs, the NP-hard uncapacitated facility location problem is a special case of the motivating application with $T = 1$, showing already that this problem is NP-hard. More interesting is that with start-time dependent variable costs, the problem is NP-hard even when there are *no fixed costs* and $T = 2$. As stated in the introduction, the application allows both the production and distribution costs to be start-time dependent. For the complexity proof, it is sufficient to allow only the production costs to be start-time dependent, the production quantities to be uncapacitated, and the variable distribution costs to be time-independent. In addition, the general problem allows variable costs to depend on both the start-time and the period in which the activity is performed. We assume here that the variable costs depend *only* on

the start-time, that is, we take $c_{ist} = \bar{c}_{is}$ for all $t = s, \ldots, T$. We therefore state this special case of the motivating application as

$$\begin{aligned}
\text{(DUFL-S)} \ \min \ & \sum_{i\in I}\sum_{s\in\mathcal{T}}\Big(\bar{c}_{is}\sum_{t=s}^{T} x_{it} + f_{is}\Big)y_{is} + \sum_{i\in I}\sum_{j\in J}\sum_{t\in\mathcal{T}} g_{ij}v_{ijt} \\
\text{s.t.} \ & x_{it} - M\sum_{s=1}^{t} y_{is} \le 0 \qquad \forall i \in I, t \in \mathcal{T} \\
& (5)-(7), \ v \ge 0, \ y_i \in Y \qquad \forall i \in I
\end{aligned}$$

where $M = \max_{t\in\mathcal{T}}\left\{\sum_{j\in J} D_{jt}\right\}$ and all data are nonnegative and rational.

Theorem 1. *DUFL-S is NP-hard even when all fixed costs are zero and $T = 2$.*

Proof. We reduce the uncapacitated facility location problem

$$(UFL) \quad \min_{S\subseteq\tilde{I}}\left\{\sum_{i\in S}\tilde{h}_i + \sum_{j\in\tilde{J}}\min_{i\in S}\tilde{g}_{ij}\right\}$$

to DUFL-S, where $\tilde{I}$ is an index set of possible facility locations, $\tilde{J}$ is an index set of demand sites, $\tilde{h}_i$ is the cost to open facility i and $\tilde{g}_{ij}$ is the cost to serve customer j from facility i.

Given an instance of UFL we construct an instance of DUFL-S which answers the question: is there a feasible solution to UFL with cost at most K? We let $I = \tilde{I}$, $J = \tilde{J}\cup\tilde{I}$ so that our set of supply nodes are the same, but in DUFL-S, our demand nodes include both the supply and demand nodes from UFL. We set the data in DUFL-S as

$$\begin{aligned}
& g_{ij} = \tilde{g}_{ij} \ \forall i \in \tilde{I}, j \in \tilde{J}, && g_{ii} = 0 \ \forall i \in \tilde{I} \\
& g_{ii'} = 2 \quad \forall i, i' \in \tilde{I} \text{ s.t. } i \ne i' && \\
& D_{j1} = 1, D_{j2} = 0 \ \forall j \in \tilde{J}, && D_{i1} = 0, D_{i2} = \tilde{h}_i \ \forall i \in \tilde{I} \\
& \bar{c}_{i1} = 1, \bar{c}_{i2} = 0 \ \forall i \in \tilde{I}, && f_{i1} = 0, f_{i2} = 0 \ \forall i \in \tilde{I}.
\end{aligned}$$

With this data, the objective in DUFL-S becomes

$$\sum_{i\in\tilde{I}} y_{i1}\sum_{t=1}^{2} x_{it} + \sum_{t=1}^{2}\sum_{i\in\tilde{I}}\sum_{j\in\tilde{J}}\tilde{g}_{ij}v_{ijt} + 2\sum_{t=1}^{2}\sum_{i\in\tilde{I}}\sum_{j\in\tilde{I}\setminus\{i\}} v_{ijt}.$$

We claim there exists a solution to UFL with cost at most K if and only if there exists a solution to DUFL-S with cost at most $K + |\tilde{J}|$. Thus, let $S \subseteq \tilde{I}$ be such that

$$\sum_{i\in S}\tilde{h}_i + \sum_{j\in\tilde{J}}\min_{i\in S}\tilde{g}_{ij} \le K.$$

We construct a solution to DUFL-S by setting $y_{i1} = 1$ if and only if $i \in S$ then setting x_{i1} and v_{ij1} to meet first period demand from the nodes in S at minimum cost. This leads to a first period cost in DUFL-S of $|\tilde{J}| + \sum_{j\in\tilde{J}} \min_{i\in S} \tilde{g}_{ij}$. In the second period, we meet the demand at each demand node $i \in \tilde{I}$ by producing at *supply* node $i \in \tilde{I}$ and shipping to demand node i. Because $y_{i1} = 1$ only if $i \in S$, the total second period cost is then $\sum_{i\in S} \tilde{h}_i$. Therefore, the total cost of this solution does not exceed

$$|\tilde{J}| + \sum_{j\in\tilde{J}} \min_{i\in S} \tilde{g}_{ij} + \sum_{i\in S} \tilde{h}_i \leq K + |\tilde{J}|.$$

Now, suppose a solution (x, y, v) to DUFL-S has cost at most $K+|\tilde{J}|$. Let $S = \{i : y_{i1} = 1\}$. Then,

$$\sum_{i\in I} y_{i1} \sum_{t=1}^{2} x_{it} = |\tilde{J}| + \sum_{i\in S} \tilde{h}_i$$

and

$$\sum_{t=1}^{2} \sum_{i\in\tilde{I}} \sum_{j\in\tilde{J}} \tilde{g}_{ij} v_{ijt} + 2 \sum_{t=1}^{2} \sum_{i\in\tilde{I}} \sum_{j\in\tilde{I}\setminus i} v_{ijt} \geq \sum_{i\in\tilde{I}} \sum_{j\in\tilde{J}} \tilde{g}_{ij} v_{ij1} \geq \sum_{j\in\tilde{J}} \min_{i\in S} \tilde{g}_{ij}$$

and hence $\sum_{i\in S} \tilde{h}_i + \sum_{j\in\tilde{J}} \min_{i\in S} \tilde{g}_{ij} \leq K$ completing the proof. □

2.4 Mixed-Integer Linear Formulations

In this section we present strong mixed-integer linear formulations for MIBL. In doing so, we pay close attention to the size of the formulations developed. In particular, we are interested in solving (at least approximately) problems with a large number of activities and a long planning horizon, so that formulations which involve a large number of auxiliary variables may be undesirable.

In this section, we assume $X = [0, M]^T$. For more general $X \subseteq [0, M]^T$, the formulations developed in this section remain valid, as long as we also include the condition that $x \in X$.

2.4.1 Extended Formulation

We begin with an extended formulation, which introduces $O(T^2)$ auxiliary variables, directly contradicting our stated goal of keeping the formulations small. However, this formulation has the nice property that it is integral for the single activity problem, and this property is useful for proving tightness of the more compact formulations we present in the sequel.

We introduce auxiliary variables w_{st} for $1 \leq s \leq t \leq T$, to represent the amount of the activity that is charged the variable costs associated with a start-period s, but is used in period $t \geq s$. This is analogous to the extended formulation for the lot sizing problem [6, 25, 55, 50], in which we can interpret w_{st} as the amount that is produced in period s to meet demand in period t. With these additional variables, MIBL is reformulated as

$$\begin{aligned}
\min \quad & \sum_{s=1}^{T}\sum_{t=s}^{T} c_{st}w_{st} + \sum_{s=1}^{T} f_s y_s & & \\
\text{s.t.} \quad & \sum_{s=1}^{t} w_{st} - x_t \geq 0 & \forall t \in \mathcal{T} & \quad (12)\\
& w_{st} - My_s \leq 0 & \forall 1 \leq s \leq t \leq T & \quad (13)\\
& w \geq 0,\ x \geq 0,\ y \in Y. &&
\end{aligned}$$

We refer to this formulation as EF.

We now study the tightness of this formulation. Let

$$F^E = \left\{(x, y, w) \in [0, M]^T \times Y \times \mathbf{R}_+^{T(T+1)/2} : (12) - (13)\right\}$$

represent the feasible region of the extended formulation, and let P^E represent the polytope obtained by dropping the integrality restriction on the binary variables in F^E. For a set F, let $\text{conv}(F)$ represent the convex hull of F. Analogous to the similar result for the extended formulation of the standard lot sizing problem [6, 50], we have

Theorem 2. $P^E = \text{conv}(F^E)$.

Proof. We prove the equivalent result that the polytope P^E has y integer in all extreme points. Let P_1^E be the polytope given by P^E with $M = 1$. We will show that P_1^E is an integral polytope. The result then follows since $(Mx, y, Mw) \in P^E$ if and only if $(x, y, w) \in P_1^E$, so that if y is integral in all extreme points of P_1^E then it is integral in all extreme points of P^E. We claim that the system defining P_1^E is *totally dual integral* (see, e.g., [50] p. 537). To prove this, consider the linear programming relaxation of EF having $M = 1$ and arbitrary objective given by

$$\max \quad \sum_{t=1}^{T} \tilde{c}_t x_t + \sum_{s=1}^{T}\sum_{t=s}^{T} \tilde{d}_{st} w_{st} + \sum_{t=1}^{T} \tilde{f}_t y_t$$

where the coefficients $(\tilde{c}, \tilde{f}, \tilde{d}) \in \mathbb{Z}^{2T+T(T+1)/2}$ are integer. The dual of this linear program is

$$\begin{aligned} \min \quad \pi_0 \quad \text{s.t.} \quad & \pi_0 - \sum_{t=s}^{T} \sigma_{st} \geq \tilde{f}_s && \forall s \in \mathcal{T} \\ & \sigma_{st} - \gamma_t \geq \tilde{d}_{st} && \forall 1 \leq s \leq t \leq T \\ & \gamma_t \geq \tilde{c}_t && \forall t \in \mathcal{T} \\ & \sigma \geq 0, \gamma \geq 0, \pi_0 \ \geq 0. \end{aligned}$$

It is simple to see that the optimal dual solution is

$$\begin{aligned} \pi_0 &= \max_{s \in \mathcal{T}} \{\tilde{f}_s + \sum_{t=s}^{T} \sigma_{st}\} \\ \sigma_{st} &= (\tilde{d}_{st} + \gamma_t)^+ && \forall 1 \leq s \leq t \leq T \\ \gamma_t &= (\tilde{c}_t)^+ && \forall t \in \mathcal{T} \end{aligned}$$

where we use the notation $(\cdot)^+ = \max\{\cdot, 0\}$. Thus, the dual solution is integral and so the system defining the primal is totally dual integral, and hence P_1^E is an integral polytope. □

We remark that if M is integral, the arguments in the above proof can be used to establish that the extreme points of P^E have x and w integer as well as y.

2.4.2 Linearizing the Bilinear Formulation

In this section, we assume the variable costs depend only on the start-time, that is we assume $c_{st} = \bar{c}_s$ for all $t = s, \ldots, T$. This is an interesting special case because it is already NP-hard, even without fixed costs. In addition, this assumption was satisfied in the application under consideration. Under this assumption, we can rewrite the objective of the MIBL formulation as

$$\sum_{s=1}^{T} \Big(f_s + \bar{c}_s \sum_{t=s}^{T} x_t \Big) y_s \tag{14}$$

A natural way to deal with the bilinear objective term appearing in (14) in a compact way is to introduce linearization variables z_s to capture the bilinear terms $y_s \sum_{t=s}^{T} x_t$ for each $s \in \mathcal{T}$. That is, z_s represents the amount of activity that is *charged* at the variable cost of

period s, $\bar{c}_s$. Then, we obtain the mixed-integer linear formulation

$$\begin{aligned} \min \quad & \sum_{s\in\mathcal{T}} f_s y_s + \sum_{s\in\mathcal{T}} \bar{c}_s z_s \\ \text{s.t.} \quad & z_s - \sum_{t=s}^{T} x_t + (1-y_s)(T-s+1)M \geq 0 \quad \forall s \in \mathcal{T} \\ & z \geq 0,\ x \geq 0,\ y \in Y, \text{ and (11)} \end{aligned} \tag{15}$$

which we refer to as the *weak linearization* (WL) formulation. If period s is the start period, then $y_s = 1$ and constraint (15) ensures that $z_s \geq \sum_{t=s}^{T} x_t$ so the variable cost $\bar{c}_s$ is charged on the activity over the entire horizon. On the other hand, if the activity does not begin in period s, $y_s = 0$ so that (15) is not binding, and because we are minimizing we will have $z_s = 0$, so that none of the activity is charged at the variable cost of period s, as desired. Although this yields a correct formulation, we will see in the computational results, Section 2.7, that the bounds from the linear programming relaxation of this formulation are extremely weak.

2.4.2.1 Strengthening Using Ideas from Lot Sizing

The main problem with the WL formulation is the presence of the weak constraints (15). Indeed, when the binary variables are fractional, it is possible to have positive activity levels, and yet have $z_s = 0$ for all s, so that we pay no variable costs for the activity we perform. Fortunately, it turns out that constraints (15) can be eliminated by using an idea from lot sizing. Note that we can interpret z_s as an *economic* amount that is paid for in period s which can be used by the *physical* activities x_t in any period $t \geq s$. With this interpretation, our formulation appears similar to the lot sizing problem, in which we have to produce to meet demands over time [7, 50]. Using this analogy, we can add the constraints,

$$\sum_{s=1}^{t} z_s \geq \sum_{s=1}^{t} x_s \quad \forall t \in \mathcal{T} \tag{16}$$

which state that the cumulative amount we pay for up to each time period t must be at least as much as the physical activity levels up to period t. Formally, we can argue validity

of (16) by observing that

$$\sum_{s=1}^{t} z_s = \sum_{s=1}^{t} y_s \sum_{j=s}^{T} x_j \geq \sum_{s=1}^{t} y_s \sum_{j=s}^{t} x_j = \sum_{j=1}^{t} x_j \sum_{s=1}^{j} y_s \geq \sum_{s=1}^{t} x_s$$

where the last inequality follows because $\sum_s y_s \leq 1$. The difference between our problem and the standard lot sizing problem is that all activity must be charged the variable cost corresponding to the start period, or equivalently, our economic production variables, z_s, can only be positive in the start period. Thus, we obtain the lot sizing inspired formulation, LS, given by

$$\begin{aligned} \min \quad & \sum_{s\in\mathcal{T}} f_s y_s + \sum_{s\in\mathcal{T}} \bar{c}_s z_s \\ \text{s.t.} \quad & z_s - (T-s+1) M y_s \leq 0 \quad \forall s \in \mathcal{T} \\ & x \geq 0,\ z \geq 0,\ y \in Y,\ (11) \text{ and } (16) \end{aligned} \tag{17}$$

where (17) guarantees that z_s can only be positive in the start period. We would obtain a more compact valid formulation by eliminating the constraints (11) and replacing them with bounds $x_t \leq M$. However, we include (11) since their presence tightens the formulation and allows the commercial MIP solver we use to generate cuts based on these variable upper bounds, and computational results indicate that this benefit far outweighs the increased formulation size.

2.4.2.2 The Convex Hull

In this section we characterize the convex hull of feasible solutions to the LS formulation. Thus, we study the set

$$F^{LS} = \left\{ (x, y, z) \in [0, M]^T \times Y \times \mathbf{R}_+^T : (16) - (17) \right\}.$$

Theorem 3. $\text{conv}(F^{LS})$ *is given by the set of* $(x, y, z) \in \mathbf{R}_+^{3T}$ *which satisfy* (8), (17) *and*

$$\sum_{t\in S} x_t \leq \sum_{t\in L} z_t + M \sum_{t\in S} \sum_{s\in\{1,\ldots,t\}\setminus L} y_s \qquad \forall S, L \subseteq \mathcal{T}. \tag{18}$$

Proof. Note that (16) and (11) are special cases of (18). Let P^{LS} be the set of (x, y, z) which satisfy the inequalities stated in the theorem. We prove $\text{conv}(F^{LS}) \subseteq P^{LS}$ by showing (18)

are valid for F^{LS}. Let $(x, y, z) \in F^{LS}$ and let $i \in \mathcal{T}$ be such that $y_i = 1$. If $y_t = 0$ for all t, set $i = T + 1$. If $i \in L$, then $\sum_{t \in L} z_t = z_i \geq \sum_{t \in \mathcal{T}} x_t \geq \sum_{t \in S} x_t$ and (18) holds. If $i \notin L$, then for $t \in S$ we have $x_t = 0 \leq M \sum_{s \in \{1,\ldots,t\} \setminus L} y_s$ if $t < i$ and $x_t \leq M = M \sum_{s \in \{1,\ldots,t\} \setminus L} y_s$ if $t \geq i$. Hence,

$$\sum_{t \in S} x_t \leq M \sum_{t \in S} \sum_{s \in \{1,\ldots,t\} \setminus L} y_s$$

and (18) holds.

Now, suppose $(x, y, z) \in P^{LS}$. We show that there exists w such that

$$\sum_{t=s}^{T} w_{st} \leq z_s \quad \forall s \in \mathcal{T} \tag{19}$$

and $(x, y, w) \in \text{conv}(F^E)$, as defined in Section 2.4.1. By Theorem 2, $P^E = \text{conv}(F^E)$ and so the latter condition can be verified by checking that (x, y, w) satisfy the inequalities defining the polyhedron P^E. Consider a network G with node set given by $V = \{u, v, r_s \ \forall s \in \mathcal{T}, m_t \ \forall t \in \mathcal{T}\}$. The arcs in this network consist of arcs from u to r_s with capacity z_s for $s \in \mathcal{T}$, arcs from r_s to m_t with capacity My_s for all $1 \leq s \leq t \leq T$, and arcs from m_t to v with capacity x_t for all $t \in \mathcal{T}$. An example of this graph for $T = 3$ is given in Figure 1, where the label on each arc represents its capacity. Since $(x, y, z) \in P^{LS}$ we have $x \in \mathbf{R}^T_+$ and $y \in Y^R$. Thus, if we associate w_{st} with the flow from r_s to m_t in this network, it is easy to check that if this network has a $u - v$ flow of $\sum_{t \in T} x_t$, then the desired w exists. It follows from the max-flow min-cut theorem that if the capacity of every $u - v$ cut in G is at least $\sum_{t \in \mathcal{T}} x_t$, then there exists a $u - v$ flow of this value. Let $U \subset V$ with $u \in U$ and $v \notin U$ and consider the cut defined by U and $V \setminus U$. Let $S = \{t \in \mathcal{T} : m_t \notin U\}$ and $L = \{s \in \mathcal{T} : r_s \notin U\}$. The capacity of this cut is

$$\sum_{s \in L} z_s + \sum_{t \in S} \sum_{s \in \{1,\ldots,t\} \setminus L} My_s - \sum_{t \in S} x_t + \sum_{t \in \mathcal{T}} x_t.$$

Since (x, y, z) satisfies (18) for all $S, L \subseteq \mathcal{T}$, it follows that the capacity of this cut is at least $\sum_{t \in \mathcal{T}} x_t$. Thus, there exists a $u - v$ flow of $\sum_{t \in \mathcal{T}} x_t$.

We complete the proof by demonstrating that the existence of w such that $(x, y, w) \in \text{conv}(F^E)$ and (19) is satisfied implies $(x, y, z) \in \text{conv}(F^{LS})$. So suppose such a w exits. We first observe that there exists a w' such that $(x, y, w') \in \text{conv}(F^E)$ and $\sum_{t=s}^{T} w'_{st} =$

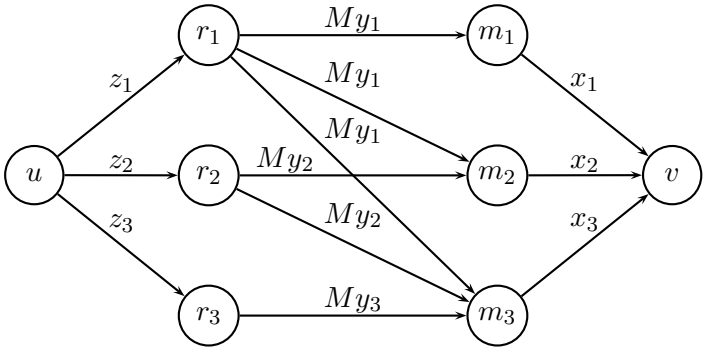

Figure 1: Example of network G with $T = 3$.

z_s $\forall s \in \mathcal{T}$. This follows because $z_s \leq (T-s+1)My_s$ so that where necessary w_{st} can be increased to obtain equality in (19) without violating the inequalities (13), i.e. $w_{st} \leq My_s$. Next, let $(x^i, y^i, w^i), i \in I$ be a set of points of F^E and $\lambda \in \mathbf{R}_+^{|I|}$ such that $\sum_{i\in I} \lambda_i = 1$ and $(x, y, w') = \sum_{i\in I} \lambda_i (x^i, y^i, w^i)$. For each $i \in I$, define z^i by $z^i_s = \sum_{t=s}^T w^i_{st}$ $\forall s \in \mathcal{T}$. Then it is easy to check that $(x, y, z) = \sum_{i\in I} \lambda_i (x^i, y^i, z^i)$ and that $(x^i, y^i, z^i) \in F^{LS}$ for $i \in I$, thus establishing that $(x, y, z) \in \text{conv}(F^{LS})$. □

The proof demonstrates how separation of (18) can be accomplished by finding the minimum cut in a network with $O(T)$ nodes. This immediately implies that separation can be accomplished with $O(T^3)$ complexity by finding the maximum flow in this network. Because of the special structure of the network, it is possible to find a minimum cut with complexity $O(T^2)$ using dynamic programming. For $t \in \mathcal{T}$ define G_t to be the subgraph of G induced by the nodes $\{u, v, r_t, \ldots, r_T, m_t, \ldots, m_T\}$ and for $k = 0, 1, \ldots, T-t+1$, define $v_t(k)$ to be the minimum capacity of a $u - v$ cut, (U, V), on G_t with the restriction that $|V \cap \{m_t, \ldots, m_T\}| = k$. Then, the capacity of the minimum $u - v$ cut in G is given by $\min\{v_1(k) : k = 0, \ldots, T\}$. The values $v_t(k)$ can be computed as follows.

Theorem 4. *$v_T(0) = x_T$, $v_T(1) = z_T$, and for $t = T-1, \ldots, 1$*

$$v_t(0) = \sum_{s=t}^{T} x_s \tag{20}$$

$$v_t(k) = \min\{v_{t+1}(k) + x_t, v_{t+1}(k-1)\} + \min\{z_t, Mky_t\} \qquad k = 1, \ldots, T-t \tag{21}$$

$$v_t(T-t+1) = v_{t+1}(T-t) + z_t. \tag{22}$$

Proof. Fist first prove $v_T(0) = x_T$ and $v_T(1) = z_T$. Note that $|V \cap \{m_T\}| = 0$ implies $m_T \in U$. If also $r_T \in U$, the value of the cut is x_T, whereas if $r_T \in V$, the value of the cut is $z_T + x_T \geq x_T$, hence $v_T(0) = x_T$. The restriction $|V \cap \{m_T\}| = 1$ implies $m_T \notin U$. If $r_T \in U$, the value of the cut is My_T, whereas if $r_T \in V$, the value of the cut is $z_T \leq My_T$ and hence $v_T(1) = z_T$.

We next prove (21). Let $k \in \{1, \ldots, T-t\}$ and (U, V) be any $u - v$ cut in G_t with $|V \cap \{m_t, \ldots, m_T\}| = k$ and let $\text{cap}(U, V)$ represent the capacity of this cut. Consider the following four cases:

Case 1: $r_t \in U$, $m_t \in V$. Then, $|V \cap \{m_{t+1}, \ldots, m_T\}| = k - 1$. Then, $\text{cap}(U, V)$ is given by the capacity of $(U \setminus \{r_t\}, V \setminus \{m_t\})$ in the graph G_{t+1}, which is at least $v_{t+1}(k-1)$, plus Mky_t, representing the capacity of the arcs from r_t to the k arcs in $V \cap \{m_t, \ldots, m_T\}$. Hence, in this case

$$\text{cap}(U, V) \geq v_{t+1}(k-1) + Mky_t. \tag{23}$$

Case 2: $r_t \in U$, $m_t \notin V$. Then, $|V \cap \{m_{t+1}, \ldots, m_T\}| = k$ and hence $\text{cap}(U, V)$ is given by the capacity of $(U \setminus \{r_t\}, V)$ in the graph G_{t+1}, which is at least $v_{t+1}(k)$, plus Mky_t plus x_t, representing the capacity of the arc from $m_t \notin V$ to v. Thus, in this case

$$\text{cap}(U, V) \geq v_{t+1}(k) + Mky_t + x_t. \tag{24}$$

Case 3: $r_t \notin U$, $m_t \in V$. This case is similar to case 1, except the cut does not include the arcs from r_t to arcs in V, and instead includes the arc from u to r_t, yielding

$$\text{cap}(U, V) \geq v_{t+1}(k-1) + z_t. \tag{25}$$

Case 4: $r_t \notin U$, $m_t \notin V$. Similar reasoning as cases 2 and 3 yields

$$\text{cap}(U, V) \geq v_{t+1}(k) + x_t + z_t. \tag{26}$$

Combining (23) - (26), we obtain

$$\begin{aligned}\text{cap}(U, V) &\geq \min\{v_{t+1}(k) + x_t + \min\{z_t, Mky_t\}, v_{t+1}(k-1) + \min\{z_t, Mky_t\}\} \\ &= \min\{v_{t+1}(k) + x_t, v_{t+1}(k-1)\} + \min\{z_t, Mky_t\}.\end{aligned}$$

Since this is true for any $u-v$ cut (U,V) which satisfy $|V\cap\{m_t,\ldots,m_T\}|=k$, this implies

$$v_t(k)\geq \min\{v_{t+1}(k)+x_t, v_{t+1}(k-1)\}+\min\{z_t, Mky_t\}.$$

We now show that by using an optimal cut from G_{t+1}, we can always construct a cut which has capacity equal to this lower bound. Indeed, consider the following two cases.

Case 1: $v_{t+1}(k)+x_t > v_{t+1}(k-1)$. Let (U',V') be a $u-v$ cut in G_{t+1} which has $|V'\cap\{m_{t+1},\ldots,m_T\}|=k-1$ and $\text{cap}(U',V')=v_{t+1}(k-1)$. Then, construct a cut (U,V) by letting $V=V'\cup\{m_t\}$ and $U=U'\cup\{r_t\}$ if $z_t\leq Mky_t$ and letting $V=V'\cup\{m_t,r_t\}$ and $U=U'$ if $z_t>Mky_t$. By construction $\text{cap}(U,V)=v_{t+1}(k-1)+\min\{z_t,Mky_t\}$.

Case 2: $v_{t+1}(k)+x_t \leq v_{t+1}(k-1)$. Let (U',V') be a $u-v$ cut in G_{t+1} which has $|V'\cap\{m_{t+1},\ldots,m_T\}|=k$ and $\text{cap}(U',V')=v_{t+1}(k)$. Then, construct a cut (U,V) by letting $V=V'$ and $U=U'\cup\{m_t,r_t\}$ if $z_t\leq Mky_t$ and letting $V=V'\cup\{r_t\}$ and $U=U'\cup\{m_t\}$ if $z_t>Mky_t$. By construction $\text{cap}(U,V)=v_{t+1}(k)+x_t+\min\{z_t,Mky_t\}$.

In either case, we obtain the lower bound and hence we have proved (21).

Now consider (20). If (U,V) is any $u-v$ cut in G_t which satisfies $|V\cap\{m_t,\ldots,m_T\}|=0$ then $m_s\in U$ for all $s=t,\ldots,T$. Then, it is clear that it is optimal to also have $r_s\in U$ for all $s=t,\ldots,T$, yielding the optimal capacity of $v_t(0)=\sum_{s=t}^{T}x_s$.

Finally, consider (22). Let (U,V) be any $u-v$ cut in G_t which satisfies $|V\cap\{m_t,\ldots,m_T\}|=T-t+1$. Then $m_s\in V$ for all $s=t,\ldots,T$. If $r_t\in U$, then $\text{cap}(U,V)\geq v_{t+1}(T-t)+M(T-t+1)y_t$, whereas if $r_t\notin U$, then $\text{cap}(U,V)\geq v_{t+1}(T-t)+z_t$. Since this is true for any such (U,V) and $z_t\leq(T-t+1)My_t$, this implies $v_t(T-t+1)\geq v_{t+1}(k-1)+z_t$. A (U,V) cut which attains this lower bound can be obtained by letting (U',V') be a cut in G_{t+1} which has $m_s\in V$ for all $s=t+1,\ldots,T$ and has $\text{cap}(U',V')=v_{t+1}(T-t)$ and then setting $U=U'$ and $V=V'\cup\{m_t\}$. □

The recursion given in Theorem 4 can be used to calculate $v_1(k)$ for all $k=0,1,\ldots,T$ in $O(T^2)$ time. In addition, as is standard in dynamic programming, the implementation of the dynamic program can be accomplished in such a way as to yield a (U,V) cut which obtains the minimum.

2.4.2.3 Start-time and Production-period Dependent Variable Costs

The LS formulation does not allow dependence of the variable costs on the period in which the activity occurs (the *activity period* in the sequel). This restriction may limit the applicability of this formulation. For example, in long-term strategic planning models it is common to incorporate discounting of costs to the present value. It is not clear that it is possible incorporate discounting of costs to the current period into this formulation.

The LS formulation can be used in one special case which allows dependence of variable costs on both the start period and the activity period. Suppose the variable cost for production in period t given that start period was $s \leq t$ is given by $c_{st} = \bar{c}_s + \tilde{c}_t$, where $\bar{c} \in \mathbf{R}^T_+$ represents the component of variable cost that is start-time dependent, and $\tilde{c} \in \mathbf{R}^T_+$ represents the component of variable cost that activity-time dependent. In this additive costs case, the LS formulation can be used as-is by using $\bar{c}$ as the start-time dependent variable costs, and adding the term $\sum_{t=1}^{T} \tilde{c}_t x_t$ to the objective. Even if the variable costs do not satisfy this additive assumption, appropriate choice of $\bar{c}$ and $\tilde{c}$ may yield a reasonable approximation.

For general costs c_{st} for $1 \leq s \leq t \leq T$, we can obtain a linear objective by adding the variables w_{st} as in the extended formulation. Although this introduces $O(T^2)$ variables, we could create a formulation which is somewhat of a hybrid between the extended formulation EF and the formulation LS, which has only $O(T)$ constraints. To do so, we replace the $O(T^2)$ constraints (13) of the extended formulation with the aggregated constraints

$$\sum_{t=s}^{T} w_{st} \leq (T - s + 1) M y_s \qquad \forall s \in \mathcal{T}$$

which are analogous to the constraints (17) used in the LS formulation. The additional rows of the extended formulation could then be added as cuts. However, in the next section we will present a linear formulation which allows dependence of variable costs on both the start time and the activity time, and does not require a quadratic number of variables. Because this formulation uses in $O(T)$ constraints *and* variables, we expect this to be a better formulation.

2.4.3 Formulation Based on Bilinear Model

We now present a linear formulation that does not introduce the auxiliary variables $z_t,\ t \in \mathcal{T}$. In addition to using fewer variables, this formulation is more flexible because it allows the variable costs to depend both on the start-time and on the period in which the activities are performed. To obtain a linear objective in this case, we introduce an upper bound variable, μ, on the bilinear term in the objective, and move the bilinear term into the constraints. That is, we simply reformulate MIBL as

$$\begin{aligned} \min \quad & \mu + \sum_{s=1}^{T} f_s y_s \\ \text{s.t.} \quad & \mu - \sum_{s=1}^{T} y_s \sum_{t=s}^{T} c_{st} x_t \geq 0 \\ & x \geq 0,\ y \in Y, \text{ and (11)}. \end{aligned} \tag{27}$$

Now, note that for any fixed feasible binary vector y the constraints reduce to a set of linear constraints. Furthermore, there are only $T+1$ feasible binary vectors, so that the feasible region is the union of exactly $T+1$ polyhedra. Therefore, optimizing a general linear function over this feasible region is easy, and consequently separating over the convex hull of this feasible region is also theoretically easy. In fact, disjunctive programming theory can be used to write an explicit, polynomial sized linear program to separate over this convex hull, see [5]. However, we prefer to have an explicit characterization of inequalities defining the convex hull, and an efficient combinatorial algorithm for separation over these inequalities. We therefore study the convex hull of the set

$$F^{BL} = \left\{(\mu, x, y) \in \mathbf{R} \times [0, M]^T \times Y : (11) \text{ and } (27)\right\}.$$

Theorem 5. $\text{conv}(F^{BL})$ *is given by the set of* $(\mu, x, y) \in \mathbf{R}_+^{T+1} \times Y^R$ *which satisfy* (11) *and*

$$\mu \geq \sum_{t=1}^{T} c_{i_t t} x_t - \sum_{s=1}^{T} \sum_{t=s}^{T} M(c_{i_t t} - c_{st})^+ y_s \tag{28}$$

for all $i_t \in \{1, \ldots, t\}\ \forall t \in \mathcal{T}$.

Proof. Let P^{BL} be the set of $(\mu, x, y) \in \mathbf{R}_+^{T+1} \times Y^R$ which satisfy (11) and (28). Let $(\mu, x, y) \in F^{BL}$ and let $i_t \in \{1, \ldots, t\}\ \forall t \in \mathcal{T}$. If $x = y = \mathbf{0}$, the inequality is trivially valid.

Otherwise, let k be the period such that $y_k = 1$ and $y_t = 0$ for all $t \neq k$. Then $x_t = 0$ for $t = 1, \ldots, k-1$ and

$$\begin{aligned}
\mu &\geq \sum_{t=k}^{T} c_{kt} x_t && (29)\\
&\geq \sum_{t=k}^{T} c_{kt} x_t - \sum_{t=k}^{T} (c_{i_t t} - c_{kt})^+ (M - x_t) && (30)\\
&\geq \sum_{t=k}^{T} c_{kt} x_t + \sum_{t=k}^{T} (c_{i_t t} - c_{kt}) x_t - \sum_{t=k}^{T} M (c_{i_t t} - c_{kt})^+ && (31)\\
&= \sum_{t=1}^{T} c_{i_t t} x_t - \sum_{s=1}^{T} \sum_{t=s}^{T} M (c_{i_t t} - c_{st})^+ y_s
\end{aligned}$$

where (29) follows from (27), (30) follows since $x_t \leq M$ for all $t \in \mathcal{T}$, and (31) follows since $x \geq 0$. This proves that (28) is valid for F^{BL} and hence $\text{conv}(F^{BL}) \subseteq P^{BL}$.

We next prove that $P^{BL} \subseteq \text{conv}(F^{BL})$. Let $(\mu, x, y) \in P^{BL}$ and consider the linear program

$$\begin{aligned}
LP^* = \max \quad & \sum_{t=1}^{T} x_t \gamma_t - \sum_{t=1}^{T} \sum_{s=1}^{t} M y_s \pi_{st} && \text{(BSLP)}\\
\text{s.t.} \quad & \gamma_t - \pi_{st} \leq c_{st} \ \ \forall 1 \leq s \leq t \leq T\\
& \gamma \geq 0, \ \pi \geq 0
\end{aligned}$$

Note that this problem decomposes by t, and that because $y_s \geq 0$ for all s, there exists an optimal solution with $\pi_{st} = (\gamma_t - c_{st})^+$ for all $1 \leq s \leq t \leq T$ and so $LP^* = \sum_{t \in \mathcal{T}} LP_t^*$ where

$$LP_t^* = \max \left\{ x_t \gamma_t - \sum_{s=1}^{t} M y_s (\gamma_t - c_{st})^+ : \gamma_t \geq 0 \right\} \quad \forall t \in \mathcal{T}. \tag{32}$$

Next note that we can assume $\gamma_t \geq \underline{c}_t := \min \{c_{st} : s = 1, \ldots, t\}$ in (32) since if $\gamma_t < \underline{c}_t$, then the second term in the objective disappears, and since $x_t \geq 0$, we will not make the solution worse by increasing γ_t. Additionally, we can assume $\gamma_t \leq \overline{c}_t := \max \{c_{st} : s = 1, \ldots, t\}$. Indeed, if $\gamma_t > \overline{c}_t$, the objective becomes

$$x_t \gamma_t - \sum_{s=1}^{t} M y_s (\gamma_t - c_{st}) = \gamma_t \Big(x_t - M \sum_{s=1}^{t} y_s \Big) + M \sum_{s=1}^{t} c_{st}$$

and because $x_t \leq M \sum_{s=1}^{t} y_s$ we will not make the solution worse by decreasing γ_t. Finally, we claim that we can restrict γ_t to be in the set $\{c_{st} : s = 1, \ldots, t\}$ for all t. Indeed,

between distinct values of c_{st}, the objective is linear, and hence, depending on the sign of the coefficient on γ_t, a solution which is no worse can be obtained by either increasing or decreasing γ_t to the next higher or lower c_{st} value. Thus,

$$\begin{aligned} LP^* &= \sum_{t=1}^{T} \max\left\{ c_{i_t t} x_t - \sum_{s=1}^{t} M y_s (c_{i_t t} - c_{st})^+ : i_t \in \{1,\dots,t\} \right\} \\ &= \max\left\{ \sum_{t=1}^{T} c_{i_t t} x_t - \sum_{s=1}^{T}\sum_{t=s}^{T} M (c_{i_t t} - c_{st})^+ y_s : i_t \in \{1,\dots,t\} \ \ \forall t \in \mathcal{T} \right\} \le \mu \end{aligned}$$

by (28) since $(\mu, x, y) \in P^{BL}$. Now, taking the dual of BSLP, we have

$$\begin{aligned} LP^* = \min \ & \sum_{s=1}^{T}\sum_{t=s}^{T} c_{st} w_{st} \\ \text{s.t. } & 0 \le w_{st} \le M y_s \quad \forall 1 \le s \le t \le T \\ & \sum_{s=1}^{t} w_{st} \ge x_s \qquad \forall s \in \mathcal{T}. \end{aligned}$$

Hence, because we also have $\sum_{t\in\mathcal{T}} y_t \le 1$, there exists $w \in \mathbf{R}_+^{T(T+1)/2}$ such that $(x, w, y) \in P^E$, the polyhedral relaxation of the extended formulation (see Section 2.4.1), and

$$\mu \ge \sum_{s=1}^{T}\sum_{t=s}^{T} c_{st} w_{st}. \tag{33}$$

Now, let $(x^i, y^i, w^i), i \in I$ be extreme points of P^E and $\lambda \in \mathbf{R}_+^{|I|}$ be such that $\sum_{i\in I} \lambda_i = 1$ and $(x, y, w) = \sum_{i\in I} \lambda_i (x^i, y^i, w^i)$. Now, for each $i \in I$, define $\mu^i = \sum_{s=1}^{T}\sum_{t=s}^{T} c_{st} w^i_{st}$. We claim that $(\mu^i, y^i, x^i) \in F^{BL}$ for $i \in I$. Indeed, by Theorem 2, the vectors y^i are integral, and hence $y^i \in Y$ for each i. Also $x^i \ge 0$ and for $t \in \mathcal{T}$ we have $x^i_t \le \sum_{s=1}^{t} w^i_{st} \le M \sum_{s=1}^{t} y^i_s$ so that (11) is satisfied. Finally, if $y^i = \mathbf{0}$, then (27) is trivially satisfied, otherwise let k be the period in which $y^i_k = 1$. Then we have,

$$\mu^i = \sum_{t=k}^{T} c_{kt} w^i_{kt} \ge \sum_{t=k}^{T} c_{kt} x^i_t = \sum_{s=1}^{T} y_s \sum_{t=s}^{T} c_{st} x^i_t$$

so that (27) is again satisfied. Next, we have from (33) that

$$\mu \ge \sum_{s=1}^{T}\sum_{t=s}^{T} c_{st} w_{st} = \sum_{i\in I} \lambda_i \sum_{s=1}^{T}\sum_{t=s}^{T} c_{st} w^i_{st} = \sum_{i\in I} \lambda_i \mu^i.$$

Thus, we conclude that there exists $\alpha \ge 0$ such that $(\mu, x, y) = \sum_{i\in I} \lambda_i (\mu^i, x^i, y^i) + \alpha(1, \mathbf{0}, \mathbf{0})$ with $(\mu^i, x^i, y^i) \in F^{BL}$ for $i \in I$. Observing that $(1, \mathbf{0}, \mathbf{0})$ is a feasible direction for F^{BL}, this establishes that $(\mu, x, y) \in \text{conv}(F^{BL})$ as desired. □

Theorem 5 yields an exponential class of inequalities (28) since for each t, one can choose an index $i_t \in \{1, \ldots, t\}$, leading to $T!$ possible inequalities. We therefore discuss separation of (28). Given a point (μ^*, x^*, y^*), testing whether there is an inequality of the form (28) which this point violates amounts to testing whether $RHS^* > \mu^*$ where

$$RHS^* = \sum_{t=1}^{T} \max_{i_t \in \{1,\ldots,t\}} \left\{ c_{i_t t} x_t^* - \sum_{s=1}^{t} M(c_{i_t t} - c_{st})^+ y_s^* \right\}.$$

Then, define $v(i,t) = \sum_{s=1}^{t} (c_{it} - c_{st})^+ y_s^*$ for $1 \leq i \leq t \leq T$ and observe that these quantities can be calculated in $O(T^3)$ time. Rewriting RHS^* as

$$RHS^* = \sum_{t=1}^{T} \max_{i=1,\ldots,t} \{c_{it} x_t^* - v(i,t)\}$$

we see that we can subsequently calculate RHS^* in $O(T^2)$ time, leading to separation in $O(T^3)$.

Despite the characterization of $\text{conv}(F^{BL})$ given in Theorem 5, it is still not obvious how to obtain a *compact* valid mixed-integer linear formulation when the nonlinear constraints (27) are dropped. A simple option is to use

$$\mu \geq \sum_{t=s}^{T} c_{st} x_t - (1 - y_s) M \sum_{t=s}^{T} c_{st} \quad \forall s \in \mathcal{T} \tag{34}$$

and then add inequalities (28) as needed to strengthen the formulation. This yields a valid formulation since if $y_s = 1$, then the right-hand side of (34) yields the correct cost lower bound for μ, whereas if $y_s = 0$ the right-hand side of (34) will not be positive, and hence will not constrain μ. However, we can avoid adding (34), which are likely to be weak, by observing that a small subset of the inequalities (28) are sufficient to guarantee a valid mixed-integer linear formulation.

Theorem 6. *F^{BL} is given by the set of $(\mu, x, y) \in \mathbf{R} \times [0, M]^T \times Y$ which satisfy* (11) *and*

$$\mu \geq \sum_{t=1}^{k-1} c_{tt} x_t + \sum_{t=k}^{T} c_{kt} x_t - \sum_{t=1}^{k-1} \sum_{s=1}^{t} M(c_{tt} - c_{st})^+ y_s - \sum_{t=k}^{T} \sum_{s=1}^{t} M(c_{kt} - c_{st})^+ y_s \quad \forall k \in \mathcal{T}. \tag{35}$$

Proof. Let

$$G = \left\{ (\mu, x, y) \in \mathbf{R} \times [0, M]^t \times Y : (11) \text{ and } (35) \right\}$$

so that our aim is to prove $G = F^{BL}$. First note that the k^{th} inequality in (35) corresponds to the inequality (28) in which we take $i_t = t$ for $t = 1, \ldots, k-1$, and $i_t = k$ for $t = k, \ldots, T$. Thus, we already know (35) are valid for F^{BL}, and hence $F^{BL} \subseteq G$.

Next, let $(\mu, x, y) \in G$. To show $(\mu, x, y) \in F^{BL}$ we must show (27) is satisfied. If $x = y = \mathbf{0}$, then (35) yields $\mu \geq 0$ for any k, and hence (27) holds. Otherwise, let k be the period such that $y_k = 1$ and $y_t = 0$ for all $t \neq k$. Then, the k^{th} inequality in (35) yields

$$\mu \geq \sum_{t=k}^{T} c_{kt} x_t - \sum_{t=k}^{T} M(c_{kt} - c_{kt})^+ = \sum_{s=1}^{T} y_s \sum_{t=s}^{T} c_{st} x_t$$

so that (27) again holds. □

As a consequence of this theorem, we obtain a new valid mixed-integer linear formulation for this problem, which we refer to as the LBL formulation, and a class of valid inequalities, given in Theorem 5 which can be added to make the formulation as tight as is possible for a single activity.

2.5 Concave Formulation

In this section we demonstrate how problem SP can be formulated as a very compact concave minimization problem, and present a specialized branch-and-cut algorithm to solve this formulation. This formulation is based on rewriting the single activity problem MIBL as

$$\min \{h(x) : x \in X\} \tag{36}$$

where $h : [0, M]^T \to \mathbf{R}$ is given by

$$\begin{aligned} h(x) = \min \quad & \sum_{s=1}^{T} \Big(f_s + \sum_{t=s}^{T} c_{st} x_t \Big) y_s \\ \text{s.t.} \quad & y \in Y, \quad \sum_{s=1}^{t} M y_s \geq x_t \quad \forall t \in \mathcal{T}. \end{aligned}$$

For any vector $x \in \mathbf{R}_+^T$, $x \neq \mathbf{0}$, we let $t_{\min}(x) = \min \{t \in \mathcal{T} : x_t > 0\}$. We also define $t_{\min}(\mathbf{0}) = T + 1$. Then, we can write $h(x)$ as

$$h(x) = \min \left\{ f_s + \sum_{t=s}^{T} c_{st} x_t : s = 1, \ldots, t_{\min}(x) \right\} \tag{37}$$

where we define $f_{T+1} = 0$. That is, for each $x \in [0, M]^T$, $h(x)$ is the minimum cost to perform the activity, subject to the feasibility requirement that the start period, s, of the activity must occur on or before the first period in which the activity level is positive.

We next observe that the function $h(x)$ is concave, although not separable concave. We therefore have a concave minimization formulation of problem SP with simple linear constraints. General purpose methods have been developed to solve such types of NP-hard problems, see e.g. [38]. However, our problem has special structure which we exploit to enable us to solve large scale instances.

Theorem 7. *h is concave over $[0, M]^T$.*

Proof. Let $x^k,\ k \in K$ be a set of points in $[0, M]^T$ and $\lambda \in \mathbf{R}_+^{|K|}$ be such that $\sum_{k\in K} \lambda_k = 1$. Also, let $x = \sum_{k\in K} \lambda_k x^k$. Then,

$$\begin{aligned} h(x) &= \min\left\{ f_s + \sum_{t=p}^{T} c_{st} x_t : s = 1, \ldots, t_{\min}(x) \right\} \\ &= \min\left\{ \sum_{k\in K} \lambda_k \Big(f_s + \sum_{t=s}^{T} c_{st} x_t^k \Big) : s = 1, \ldots, t_{\min}(x) \right\} \\ &\geq \sum_{k\in K} \lambda_k \min\left\{ f_s + \sum_{t=s}^{T} c_{st} x_t^k : s = 1, \ldots, t_{\min}(x^k) \right\} \qquad (38) \\ &= \sum_{k\in K} \lambda_k h(x^k) \end{aligned}$$

where (38) follows since $t_{\min}(x) \leq t_{\min}(x^k)$ for all k such that $\lambda_k > 0$. □

We propose a specialized branch-and-cut method to solve this formulation. First, we reformulate (36) by introducing an objective upper bound variable μ to obtain the formulation

$$\min\left\{ \mu : \mu \geq h(x), x \in [0, M]^T \right\}. \qquad \text{(CM)}$$

In our method, we relax the nonlinear inequality $\mu \geq h(x)$ and subsequently enforce it by branching. In addition, we generate valid inequalities to approximate the non-convex set

$$E = \left\{ (\mu, x) \in \mathbf{R} \times [0, M]^T : \mu \geq h(x) \right\} \qquad (39)$$

in order to obtain tight lower bounds at nodes in our branch and bound tree.

In describing this method, we continue to focus on the single activity problem CM, but it should be understood that in the context of the overall problem SP, the branching may have to be done on each of the activities in the formulation, and consequently, at each node in the tree a choice must be made as to which activity to branch on. Furthermore, we present node relaxations for a single activity, but it should be understood that when we refer to solving a node relaxation, we are solving the relaxation of the overall problem, using the relaxations from all the activities together.

2.5.1 Branching and Lower Bounds

In the mixed-integer linear formulations which had binary variables y, branching on these variables meant we were branching on the decision of which period would be the start period. In this approach, we still branch on this decision, implicit in the definition of h, but we do so without introducing the binary variables. Specifically, we branch on the implicit variable, s, representing the start period of the activity. At every node n in the branch and bound tree, we will have that $s \in \{l(n), \ldots, u(n)\}$ where $1 \leq l(n) \leq u(n) \leq T+1$. Recall that if $T+1$ is the start period, this means the activity never starts. At the root node, node 0, we set $l(0) = 1$ and $u(0) = T+1$.

Lower Bounds

We are interested in deriving lower bounds on the cost function $h(x)$ subject to the restriction that the start period s satisfies $l \leq s \leq u$. Therefore, define the cost function given this restriction by

$$h(x; l, u) = \min \left\{ f_s + \sum_{t=s}^{T} c_{st} x_t : l \leq s \leq \min(u, t_{\min}(x)) \right\}$$

for x such that $x_t = 0$ for $t = 1, \ldots, l-1$. Also, define

$$\underline{f}(l, u) = \min \{ f_s : l \leq s \leq u \}$$
$$\underline{c}_t(l, u) = \min \{ c_{st} : l \leq s \leq \min(t, u) \} \text{ for } t = l, \ldots, T.$$

Then, we have the lower bound given by

Theorem 8.

$$h(x;l,u) \geq \underline{f}(l,u) + \sum_{t=l}^{T} \underline{c}_t(l,u)x_t. \tag{40}$$

Proof. Let x be such that $x_t = 0$ for $t = 0, \ldots, l-1$. For convenience let $m = \min(u, t_{\min}(x))$. Then,

$$\begin{aligned} h(x;l,u) &= \min\left\{ f_s + \sum_{t=s}^{T} c_{st}x_t : l \leq s \leq m \right\} \\ &\geq \underline{f}(l,m) + \min\left\{ \sum_{t=s}^{T} c_{st}x_t : l \leq s \leq m \right\} \\ &\geq \underline{f}(l,u) + \sum_{t=l}^{u} \min\left\{c_{st} : l \leq s \leq t\right\} x_t + \sum_{t=u+1}^{T} \underline{c}_t(l,u)x_t \\ &= \underline{f}(l,u) + \sum_{t=l}^{T} \underline{c}_t(l,u)x_t \end{aligned}$$

where the last inequality follows since for $t \leq u$, $x_t > 0$ implies $t_{\min}(x) \leq t \leq u$. □

At the root node we have no restrictions on the start time so that $l = 1$ and $u = T + 1$ and we obtain the lower bound $h(x) \geq \sum_{t=1}^{T} \underline{c}(1,t)x_t$, where the fixed cost term vanished because, by definition, $f_{T+1} = 0$.

At a node in the branch-and-bound tree, if we have $l \leq s \leq u$, we obtain a lower bound by replacing $\mu \geq h(x)$ with

$$\mu \geq \underline{f}(l,u) + \sum_{t=l}^{T} \underline{c}_t(l,u)x_t. \tag{41}$$

An important property of this lower bound is that it is exact when $l = u$. That is,

$$h(x;l,l) = f_l + \sum_{t=l}^{T} c_{lt}x_t = \underline{f}(l,l) + \sum_{t=l}^{T} \underline{c}_t(l,l)x_t$$

so that if (41) is enforced when $l = u$, we necessarily have $\mu \geq h(x)$.

Branching

At any node n in the tree, we first solve the relaxation for that node obtained by including (41) in the linear program and obtain a solution x. If the optimal objective of the relaxation exceeds the cost of the best incumbent solution, we can fathom this node without branching further. Otherwise, we check whether the inequality $\mu \geq h(x)$ is violated (for

any activity). If not, we have a new incumbent solution, and we need not explore this node any further. If so, then because we have enforced (41), we must have $l(n) < u(n)$, so we can select $k \in \{l(n), \ldots, u(n) - 1\}$. We then create two nodes, d_1 and d_2 by enforcing $s \leq k$ in d_1 and enforcing $s > k$ in d_2. This is achieved by setting $l(d_1) = l(n)$ and $u(d_1) = k$, and $l(d_2) = k + 1$ and $u(d_2) = u(n)$. Thus, the lower bound (41) will be updated in the two nodes, and in addition, we will enforce that $x_t = 0$ for $t = 1, \ldots, k$ in node d_2, reflecting the restriction in that node that the activity cannot start until period $k + 1$.

This branching scheme ensures that each path in the resulting branch-and-bound tree will finitely terminate with a leaf node in which $l(n) = u(n)$ (if not earlier) and will therefore not have to be explored further. Thus the branching scheme is finite for each single activity, and hence will be finite for finitely many activities.

2.5.2 Improving the Lower Bounds

In the LS and LBL formulations we were able to explicitly characterize the convex hull of the set of feasible solutions. We have not been able to do that for the feasible set E of this formulation, given by (39). Fortunately, we can still separate all inequalities valid for E by solving a linear program. The proof follows a direct derivation of this linear program, but we note that it could also be obtained by disjunctive programming theory [5].

Theorem 9. *Let $(\mu, x) \in \mathbf{R} \times [0, M]^T$. Then, $(\mu, x) \in \text{conv}(E)$ if and only if $\mu \geq v^*$ where*

$$\begin{aligned} v^* = \max\ & \beta + \sum_{t=1}^{T} \alpha_t x_t \\ \text{s.t. } & M \sum_{t=s}^{T} \sigma_{st} + \beta \leq f_s \quad \forall s \in \mathcal{T} \\ & \alpha_t - \sigma_{st} \leq c_{st} \qquad \forall 1 \leq s \leq t \leq T \\ & \beta \leq 0, \quad \sigma_{st} \geq 0 \qquad \forall 1 \leq s \leq t \leq T. \end{aligned} \tag{42}$$

Moreover, if $\mu < v^*$, the optimal solution yields an inequality of the form

$$\mu \geq \beta + \sum_{t=1}^{T} \alpha_t x_t, \tag{43}$$

which cuts off (μ, x) and is valid for $\text{conv}(E)$.

Proof. Consider a generic valid inequality for E given by $\gamma\mu \geq \beta + \sum_{t=1}^{T} \alpha_t x_t$. We are interested only in nontrivial inequalities, i.e., those which are not implied by the bounds on x. Note that any such inequality will have a nonzero coefficient on μ. Furthermore, because the direction $(1, \mathbf{0})$ is feasible for E, the coefficient on μ must be positive. Therefore, by scaling we can assume that any nontrivial inequality for E is of the form (43).

Now, observe that $E = \bigcup_{s=1}^{T+1} E_s$ where

$$E_s = \left\{ (\mu, x) \in \mathbf{R} \times [0, M]^T : \mu \geq f_s + \sum_{t=s}^{T} c_{st} x_t,\ x_t = 0, t = 1, \ldots, s-1 \right\}.$$

Note that for each $s \in \mathcal{T}$ the extreme points of E_s are given by $x_t = M\ \forall t \in S$ and $x_t = 0$ otherwise, and $\mu = f_s + M \sum_{t \in S} c_{st}$ for all $S \subseteq \{s, \ldots, T\}$. Also note that $E_{T+1} = \{(0, \mathbf{0})\}$.

Since any extreme point of $\text{conv}(E)$ must be an extreme point of E_s for some $s \in \{1, \ldots, T+1\}$, we conclude that an inequality (43) is valid for E if and only if $\beta \leq 0$ and

$$M \max_{S \subseteq \{s, \ldots, T\}} \sum_{t \in S} (\alpha_t - c_{st}) + \beta \leq f_s \quad \forall s = 1, \ldots, T. \tag{44}$$

The s^{th} condition in (44) is equivalent to $f_s \geq \beta + M\theta_s$ where

$$\begin{aligned} \theta_s = \max \ & \sum_{t=s}^{T} \omega_{st}(\alpha_t - c_{st}) && = \quad \min \ \sum_{t=s}^{T} \sigma_{st} \\ \text{s.t. } & 0 \leq \omega_{st} \leq 1 \ \ \forall t = s, \ldots, T && \text{s.t. } \sigma_{st} \geq \alpha_t - c_{st} \ \ \forall t = s, \ldots, T \\ & && \sigma_{st} \geq 0 \qquad \forall t = s, \ldots, T \end{aligned}$$

by integrality of the first linear program and linear programming duality. It follows that inequality (43) is valid for E if and only if there exists σ such that (β, α, σ) is feasible to (42).

It only remains to prove that v^* exists and is finite. This follows since the linear program (42) is feasible since $\mathbf{0}$ is a feasible solution and bounded since

$$\beta + \sum_{t=1}^{T} \alpha_t x_t \leq f_1 - M \sum_{t=1}^{T} \sigma_{1t} + \sum_{t=1}^{T} x_t (c_{1t} + \sigma_{1t}) \leq f_1 + \sum_{t=1}^{T} c_{1t} x_t.$$

□

2.5.3 Feasible Solutions

Any solution to a node relaxation in the branch-and-bound tree will yield a solution which satisfies the physical constraints of the concave formulation. Thus, all we need to do to obtain a feasible solution at any node is to calculate the true cost of each activity for the levels given by the relaxation solution.

2.6 Special Case: Nondecreasing Activities

We now turn the special case in which we restrict the flows to be nondecreasing over time, that is we add the constraint $x \in X^N = \left\{x \in \mathbf{R}_+^T : x_1 \leq x_2 \leq \cdots \leq x_T \leq M\right\}$. This constraint was present in the application that motivated this work.

As long as we explicitly include the condition $x \in X^N$, all the formulations we have developed remain valid. However, by using the fact that $x \in X^N$, we can obtain stronger formulations, which we shall do in this section.

The mixed-integer bilinear formulation for the problem with this nondecreasing constraint then becomes MIBL-N:

$$\begin{aligned} \min \ & \sum_{s=1}^{T}\Big(f_s + \sum_{t=s}^{T} c_{st}x_t\Big)y_s \\ \text{s.t. } & x_t \leq \sum_{s=1}^{t} My_s \qquad \forall t \in \mathcal{T} \\ & x \in X^N, y \in Y. \end{aligned} \tag{45}$$

2.6.1 Extended Formulation

With the addition of the nondecreasing constraint, the extended formulation, EF, presented in Section 2.4.1 is no longer integral. Therefore, we present the following formulation, EF-N,

which is valid when the nondecreasing constraints are present, and is integral.

$$\begin{aligned}
\min\ & \sum_{s=1}^{T}\sum_{t=s}^{T}\Big(\sum_{i=t}^{T} c_{si}\Big)w_{st} + \sum_{s=1}^{T} f_s y_s \\
\text{s.t.}\ & \sum_{s=1}^{t} w_{st} \geq x_t - x_{t-1} \qquad \forall t \in \mathcal{T} \qquad (46)\\
& \sum_{t=s}^{T} w_{st} \leq M y_s \qquad \forall s \in \mathcal{T} \qquad (47)\\
& x \in X^N, y \in Y, w \geq 0.
\end{aligned}$$

In the above, we define $x_0 := 0$. Note that this formulation has $O(T^2)$ variables as did formulation EF, but unlike formulation EF, it has only $O(T)$ constraints. Let us define

$$F^{EN} = \Big\{(x, y, w) \in X^N \times Y \times \mathbf{R}_+^{T(T+1)/2} : (46) - (47)\Big\}.$$

Theorem 10. *EF-N is a valid formulation for MIBL-N.*

Proof. Let (x, y) be feasible to MIBL-N, with objective cost $\mu = \sum_{s=1}^{T}\Big(f_s + \sum_{t=s}^{T} c_{st}x_t\Big)y_s$. We show there exists $w \in \mathbf{R}_+^{T(T+1)/2}$ such that $(x, y, w) \in F^{EN}$ and

$$\mu = \sum_{s=1}^{T}\sum_{t=s}^{T}\sum_{i=t}^{T} c_{si}w_{st} + \sum_{s=1}^{T} f_s y_s. \qquad (48)$$

First, if $y = \mathbf{0}$, then also $x = \mathbf{0}$ and we can take $w = \mathbf{0}$ and (48) trivially holds. So assume $y \neq \mathbf{0}$ and let k be the period such that $y_k = 1$ and $y_t = 0$ for $t \neq k$. Now let $w \in \mathbf{R}_+^{T(T+1)/2}$ be given by $w_{st} = 0$ for all $t = s, \ldots, T, s \neq k$ and $w_{kt} = x_t - x_{t-1}$ for $t = k, \ldots, T$. Then (46) holds for all t and (47) holds for all $s \neq k$. For $s = k$, we have $\sum_{t=k}^{T} w_{kt} = x_T \leq M = My_k$ so that again (47) holds. Finally, we have

$$\begin{aligned}
\mu = \sum_{s=1}^{T}\Big(f_s + \sum_{t=s}^{T} c_{st}x_t\Big)y_s &= \sum_{s=1}^{T} f_s y_s + \sum_{i=k}^{T} c_{ki}x_i \\
&= \sum_{s=1}^{T} f_s y_s + \sum_{i=k}^{T} c_{ki}\sum_{t=k}^{i}(x_t - x_{t-1}) \\
&= \sum_{s=1}^{T} f_s y_s + \sum_{t=k}^{T}\sum_{i=t}^{T} c_{ki}w_{kt} \\
&= \sum_{s=1}^{T} f_s y_s + \sum_{s=1}^{T}\sum_{t=s}^{T}\Big(\sum_{i=t}^{T} c_{si}\Big)w_{st}
\end{aligned}$$

so that (48) holds.

Next, let $(x, y, w) \in F^{EN}$ and let μ be the cost of this solution in EF-N, given in (48). We show (x, y) is feasible for MIBL-N and $\mu \geq \sum_{s=1}^{T}\Big(f_s + \sum_{t=s}^{T} c_{st}x_t\Big)y_s$. If $y = \mathbf{0}$, then also $x = \mathbf{0}$ and $w = \mathbf{0}$ and the claim is trivial. So suppose $y \neq \mathbf{0}$ and let k be such that $y_k = 1$ and $y_t = 0$ for all $t \neq k$. Then, $w_{st} = 0$ for all $s \neq k, t = s, \ldots, T$ and $\sum_{t=k}^{T} w_{kt} \leq M$. Hence, by (46) we have $w_{kt} = \sum_{s=1}^{t} w_{st} \geq x_t - x_{t-1}$ for $t \geq k$. Thus, by (48) we have

$$\begin{aligned}
\mu &= f_k + \sum_{t=k}^{T}\sum_{i=t}^{T} c_{ki}w_{kt} \\
&\geq f_k + \sum_{t=k}^{T}\sum_{i=t}^{T} c_{ki}(x_t - x_{t-1}) \\
&= f_k + \sum_{t=k}^{T} c_{kt}x_t = \sum_{s=1}^{T}\Big(f_s + \sum_{t=s}^{T} c_{st}x_t\Big)y_s.
\end{aligned}$$

Finally, for $j \in \mathcal{T}$ such that $j < k$ we have

$$x_j = \sum_{t=1}^{j}(x_t - x_{t-1}) \leq \sum_{t=1}^{j}\sum_{s=1}^{t} w_{st} = 0 = My_j$$

and for $j \geq k$ we have

$$x_j \leq \sum_{t=1}^{j}\sum_{s=1}^{t} w_{st} = \sum_{t=k}^{j} w_{kt} \leq My_k = M\sum_{t=1}^{j} y_t$$

by (47) so that (45) holds, completing the proof. □

Let P^{EN} be the polytope defined by relaxing the integrality restrictions in F^{EN}. We have the analog to Theorem 2.

Theorem 11. $P^{EN} = \text{conv}(F^{EN})$.

Proof. As in the proof of Theorem 2 we prove the polytope P^{EN} is integral when $M = 1$ which establishes the theorem since it then implies y is integral in all extreme points of P^{EN} even when $M \neq 1$. So consider the case $M = 1$ and apply the transformation $\delta_t = x_t - x_{t-1}$

for $t = 1, \ldots, T$. Then the constraints in the relaxation P^{EN} become

$$\delta_t - \sum_{s=1}^{t} w_{st} \leq 0 \quad \forall t \in \mathcal{T}$$

$$\sum_{t=s}^{T} w_{st} - y_s \leq 0 \quad \forall s \in \mathcal{T}$$

$$\sum_{t \in \mathcal{T}} y_t \leq 1$$

$$\sum_{t \in \mathcal{T}} \delta_t \leq 1$$

in addition to the nonnegativity on y, w and δ. This constraint matrix is totally unimodular and the right-hand side is integral and hence the corresponding polytope is integral. It follows that the polytope P^{EN} is also integral. □

2.6.2 A Linearized Formulation

In this section, we study how the lot sizing inspired formulation, LS, given in Section 2.4.2 can be strengthened using the nondecreasing restriction on the activities. As in Section 2.4.2, this formulation will work only when the variable cost depends only on the start-time, that is $c_{st} = \bar{c}_s$ for all $1 \leq s \leq t \leq T$. Let

$$F^{LSN} = \left\{(x, y, z) \in X^N \times Y \times \mathbf{R}_+^T : (16) - (17)\right\}$$

be the feasible region of this formulation. The only difference between F^{LSN} and F^{LS} defined in Section 2.4.2 is that in F^{LSN} we restrict $x \in X^N$ rather than just $x \in [0, M]^T$.

We start by showing how the lot sizing inequalities

$$\sum_{s=1}^{t} z_s \geq \sum_{s=1}^{t} x_s \qquad \forall t \in \mathcal{T} \tag{16}$$

which were used to obtain the improved formulation LS can be tightened.

Theorem 12. *The inequalities*

$$\sum_{s=1}^{t} z_s \geq \sum_{s=1}^{t} x_s + (T - t)x_t \qquad \forall t \in \mathcal{T} \tag{49}$$

are valid for F^{LSN}.

Proof. Let $(x, y, z) \in F^{LSN}$ and let k be the period in which $y_k > 0$ (if $y_t = 0\ \forall t \in \mathcal{T}$ the inequality holds trivially). Because $z_t = 0$ for all $t \neq k$, we must have $z_k \geq \sum_{s=k}^{T} x_s$ and $x_t = 0$ for $t = 1, \ldots, k-1$. Thus, if $t \geq k$, we have

$$\sum_{s=1}^{t} z_s = z_k \geq \sum_{s=k}^{T} x_s \geq \sum_{s=k}^{t} x_s + (T-t)x_t = \sum_{s=1}^{t} x_s + (T-t)x_t$$

and if $t < k$, then

$$\sum_{s=1}^{t} z_s = 0 = \sum_{s=1}^{t} x_s + (T-t)x_t.$$

Thus, for all t, the inequality (49) is valid. □

We refer to the formulation obtained from LS by adding the constraints $x \in X^N$, and replacing the constraints (16) with the constraints (49) as LS-N. We now present a class of inequalities valid for F^{LSN} which are similar to the class given in (18).

Theorem 13. *Let $L, S \subseteq \mathcal{T}$ with $S = \{s_1, \ldots, s_l\}$ and $s_1 < s_2 < \cdots < s_l$. Then, the inequality*

$$\sum_{i=1}^{l} (s_{i+1} - s_i) x_{s_i} \leq \sum_{t \in L} z_t + M \sum_{i=1}^{l} \sum_{t \in \{1,\ldots,s_i\} \setminus L} (s_{i+1} - s_i) y_t \tag{50}$$

is valid for F^{LSN} where $s_{l+1} := T + 1$.

Proof. Let k be such that $y_k = 1$ and $y_t = 0$ for all $t \neq k$. Then $x_t = z_t = 0$ for all $t < k$, and $z_k \geq \sum_{t=k}^{T} x_t$. Suppose first that $k \in L$. Then,

$$\sum_{i=1}^{l} (s_{i+1} - s_i) x_{s_i} = \sum_{i=1}^{l} \sum_{t=s_i}^{s_{i+1}-1} x_{s_i} \leq \sum_{i=1}^{l} \sum_{t=s_i}^{s_{i+1}-1} x_t = \sum_{t=1}^{T} x_t \leq z_k = \sum_{t \in L} z_t$$

and hence (50) holds.

Now suppose $k \notin L$. Let $j = \min\{i = 1, \ldots, l+1 : s_i \geq k\}$. Then,

$$\sum_{i=1}^{l} (s_{i+1} - s_i) x_{s_i} = \sum_{i=j}^{l} (s_{i+1} - s_i) x_{s_i} \leq M(s_{l+1} - s_j) = M(T - s_j + 1)$$

and

$$\sum_{i=1}^{l} \sum_{t \in \{1,\ldots,s_i\} \setminus L} (s_{i+1} - s_i) y_t = M \sum_{i=j}^{l} (s_{i+1} - s_i) = M(T - s_j + 1)$$

and hence (50) holds, completing the proof. □

We conjecture that inequalities (50) are sufficient to define the convex hull of F^{LSN}, but have not yet been able to prove this. This is an area for further work.

We now discuss separation of inequalities (50). We begin by writing (50) in a slightly different form. For $S \subseteq \mathcal{T}$ and $t \in \mathcal{T}$, define

$$g_t(S) = \min\{k \in S \cup \{T+1\} : k \geq t\}.$$

Then,

$$\begin{aligned}\sum_{i=1}^{l} \sum_{t \in \{1,\ldots,s_i\} \setminus L} (s_{i+1} - s_i) y_t &= \sum_{t \in \mathcal{T} \setminus L} \sum_{i : s_i \geq t} (s_{i+1} - s_i) y_t \\ &= \sum_{t \in \mathcal{T} \setminus L} (T - g_t(S) + 1) y_t.\end{aligned}$$

Given $(x, y, z) \in X^N \times Y^R \times \mathbf{R}_+^T$ satisfying (16) - (17) the separation problem to determine if there is an inequality (50) which is violated is

$$\min_{\substack{S=\{s_1,\ldots,s_l\} \subseteq \mathcal{T} \\ L \subseteq \mathcal{T}}} \Big\{ \sum_{t \in L} z_t + M \sum_{t \in \mathcal{T} \setminus L} (T - g_t(S) + 1) y_t - \sum_{i=1}^{l} (s_{i+1} - s_i) x_{s_i} \Big\}.$$

However, note that for a given set S, the optimal set L is simply determined by letting $t \in L$ if $z_t \leq M(T - g_t(S) + 1) y_t$ for each $t \in \mathcal{T}$. Thus, the separation problem reduces to

$$\min_{S=\{s_1,\ldots,s_l\} \subseteq \mathcal{T}} \Big\{ \sum_{t=1}^{T} \min\{z_t, M(T - g_t(S) + 1) y_t\} - \sum_{i=1}^{l} (s_{i+1} - s_i) x_{s_i} \Big\}.$$

Now construct an acyclic network with node set $\{0, 1, \ldots, T, T+1\}$ and arcs from s to t for all $0 \leq s < t \leq T+1$, with cost

$$d_{st} = \sum_{j=s+1}^{t} \min\{z_j, M(T - t + 1) y_j\} - (t - s) x_s$$

where we define $z_{T+1} = x_0 = 0$. Each path from 0 to $T+1$ in this network, say $(0, s_1, \ldots, s_l, T+1)$, corresponds to a set $S = \{s_1, \ldots, s_l\} \subseteq \mathcal{T}$ and vice versa. The cost of the path $(0, s_1, \ldots, s_l, T+1)$ is

$$\begin{aligned}\sum_{i=0}^{l} d_{s_i, s_{i+1}} &= \sum_{i=0}^{l} \sum_{j=s_i+1}^{s_{i+1}} \min\{z_j, M(T - s_{i+1} + 1) y_j\} - \sum_{i=1}^{l} (s_{i+1} - s_i) x_{s_i} \\ &= \sum_{t=1}^{T} \min\{z_t, M(T - g_t(S) + 1) y_t\} - \sum_{i=1}^{l} (s_{i+1} - s_i) x_{s_i} \qquad (51)\end{aligned}$$

where (51) follows since $g_t(S) = s_{i+1}$ for j such that $s_i < j \leq s_{i+1}$. It follows that separation of inequalities (50) can be accomplished by finding a shortest path from 0 to $T+1$ in this network. Calculating the cost coefficients can be done in $O(T^2)$ time and finding the shortest path can then be accomplished in $O(T^2)$ by finding the shortest path from s to $T+1$ in the order $s = T, \ldots, 1, 0$. Thus, separation of inequalities (50) can be accomplished with complexity $O(T^2)$.

2.6.3 Formulation Based on Bilinear Model

We now study how the LBL formulation introduced in Section 2.4.3 can be strengthened using the nondecreasing restriction $x \in X^N$. Let

$$F^{BLN} = \left\{(\mu, x, y) \in \mathbf{R} \times X^N \times Y : (45) \text{ and } (27)\right\}.$$

F^{BLN} represents the feasible region of MIBL-N, with the bilinear term in the objective moved into the constraints.

We next show that if $(\mu, x, y) \in \mathbf{R} \times X^N \times Y^R$ satisfy (45) then separation of (μ, x, y) from $\text{conv}(F^{BLN})$ can be accomplished by solving

$$\begin{aligned} V(x,y) = \max_{\delta,\alpha} \quad & \sum_{t=1}^{T}(x_t - x_{t-1})\delta_t - \sum_{s=1}^{T} M y_s \alpha_s & & \\ \text{s.t.} \quad & \delta_t - \alpha_s \leq \sum_{i=t}^{T} c_{si} & \forall 1 \leq s \leq t \leq T & \qquad \text{(BLLP-P)} \\ & \delta \geq 0,\ \alpha \geq 0, & & \end{aligned}$$

or equivalently, its dual

$$\begin{aligned} \min_{w} \quad & \sum_{s=1}^{T}\sum_{t=s}^{T}\sum_{i=t}^{T} c_{si} w_{st} & & \\ \text{s.t.} \quad & \sum_{s=1}^{t} w_{st} \geq x_t - x_{t-1} & \forall t \in \mathcal{T} & \qquad \text{(BLLP-D)} \\ & \sum_{t=s}^{T} w_{st} \leq M y_s & \forall s \in \mathcal{T} & \\ & w \geq 0, & & \end{aligned}$$

which is recognized to be a minimum cost network flow problem on a bipartite network. We first show that feasible solutions to BLLP-P yield valid inequalities for F^{BLN}.

Lemma 14. *Suppose (δ, α) is feasible to BLLP-P. Then*

$$\mu \geq \sum_{t=1}^{T} \delta_t (x_t - x_{t-1}) - \sum_{s=1}^{T} M \alpha_s y_s \tag{52}$$

is a valid inequality for F^{BLN}.

Proof. Let $(\mu, x, y) \in F^{BLN}$. By linear programming duality, if (δ, α) is feasible to BLLP-P then

$$\sum_{t=1}^{T} (x_t - x_{t-1}) \delta_t - \sum_{s=1}^{T} M y_s \alpha_s \leq \sum_{s=1}^{T} \sum_{t=s}^{T} \sum_{i=t}^{T} c_{si} w_{st} \tag{53}$$

for all w feasible to BLLP-D. Now, by Theorem 10, since (x, y) is feasible to MIBL-N there exists w such that (x, y, w) is feasible to EF-N, and

$$\sum_{s=1}^{T} \sum_{t=s}^{T} \sum_{i=t}^{T} c_{si} w_{st} = \sum_{s=1}^{T} y_s \sum_{t=s}^{T} c_{st} x_t. \tag{54}$$

But, then w is feasible to BLLP-D and so (53) holds. Combining (53) with (54) yields

$$\sum_{t=1}^{T} (x_t - x_{t-1}) \delta_t - \sum_{s=1}^{T} M y_s \alpha_s \leq \sum_{s=1}^{T} y_s \sum_{t=s}^{T} c_{st} x_t \leq \mu$$

where the last inequality follows from (27), and hence (52) holds. □

Further exploiting the relationship between BLLP-D and the EF-N, we obtain

Theorem 15. *Let $(\mu, x, y) \in \mathbf{R} \times X^N \times Y^R$ satisfy (45). Then, $(\mu, x, y) \in \text{conv}(F^{BLN})$ if and only if $V(x, y) \leq \mu$.*

Proof. First note that BLLP-P is feasible since $(\mathbf{0}, \mathbf{0})$ is a feasible solution. We claim that also BLLP-D is feasible. First, by similar arguments to the proof of Theorem 11, we have $\{(x, y) \in X^N \times Y^R : (45)\} = \text{conv}\big(\{(x, y) \in X^N \times Y : (45)\}\big)$. Hence, there exist (x^i, y^i) with y^i integer for $i \in I$ and $\lambda \in \mathbf{R}_+^{|I|}$ such that $\sum_{i \in I} \lambda_i = 1$ and $\sum_{i \in I} \lambda_i (x^i, y^i) = (x, y)$. Because EF-N is a valid formulation for MIBL-N, there exists w^i for $i \in I$ such that $(x^i, y^i, w^i) \in F^{EN}$. Thus, $w = \sum_{i \in I} \lambda_i w^i$ is feasible for BLLP-D. Therefore, both BLLP-P and BLLP-D have optimal solutions, so $V(x, y)$ is well defined and finite.

Now, the "only if" part follows from Lemma 14. Next suppose $V(x,y) \leq \mu$. Then, there exists w such that $(x,y,w) \in P^{EN}$, and

$$\sum_{s=1}^{T}\sum_{t=s}^{T}\sum_{i=t}^{T} c_{si} w_{st} \leq \mu. \tag{55}$$

By Theorem 11, there exists $(x^i, y^i, w^i) \in F^{EN}$ for $i \in I$ and $\lambda \in \mathbf{R}_+^{|I|}$ such that $\sum_{i\in I} \lambda_i = 1$ and $\sum_{i\in I} \lambda_i (x^i, y^i, w^i) = (x, y, w)$. Now, for $i \in I$ let $\mu_i = \sum_{s=1}^{T} y_s^i \sum_{t=s}^{T} c_{st} x_t^i$. By Theorem 10,

$$\sum_{s=1}^{T}\sum_{t=s}^{T}\sum_{i=t}^{T} c_{si} w_{st}^i \geq \mu_i$$

for each $i \in I$ and hence

$$\begin{aligned}\sum_{i\in I} \lambda_i \mu_i &\leq \sum_{i\in I} \lambda_i \sum_{s=1}^{T}\sum_{t=s}^{T}\sum_{i=t}^{T} c_{si} w_{st}^i \\ &= \sum_{s=1}^{T}\sum_{t=s}^{T}\sum_{i=t}^{T} c_{si} w_{st} \leq \mu\end{aligned}$$

by (55). Thus, there exists $\beta \geq 0$ such that $(\mu, x, y) = \sum_{i\in I} \lambda_i (\mu^i, x^i, y^i) + \beta(1, \mathbf{0}, \mathbf{0})$ and so $(\mu, x, y) \in \mathrm{conv}(F^{BLN})$ since $(1, \mathbf{0}, \mathbf{0})$ is a feasible direction for F^{BLN}. □

Thus, if $(\mu, x, y) \in \mathbf{R} \times X^N \times Y^R$ satisfies (45) then separation is indeed accomplished by solving BLLP-P. If $V(x,y) > \mu$ and (δ, α) is the optimal solution to BLLP-P, then (52) defines a valid inequality which is violated by (μ, x, y). Note that BLLP-D has fewer rows than BLLP-P, so that it may be computationally advantageous to solve BLLP-D rather than BLLP-P.

As in the case without the nondecreasing restriction, it is sufficient to take a subset of size T of the valid inequalities defined in Lemma 14 to yield a valid formulation. For $k \in \mathcal{T}$ we define $\hat{\alpha}^k$ by

$$\hat{\alpha}_s^k = \begin{cases} \max\left\{\left(\sum_{i=t}^{T} c_{ki} - \sum_{i=t}^{T} c_{si}\right)^+ : k \leq t \leq T\right\} & s = 1, \ldots, k-1 \\ 0 & s = k \\ \max\left\{\left(\sum_{i=t}^{T} c_{ki} - \sum_{i=t}^{T} c_{si}\right)^+ : s \leq t \leq T\right\} & s = k+1, \ldots, T \end{cases}$$

and define $\hat{\delta}^k$ by

$$\hat{\delta}_t^k = \begin{cases} \min\Big\{\hat{\alpha}_s^k + \sum_{i=t}^T c_{si} : 1 \le s \le t\Big\} & t = 1, \ldots, k-1 \\ \sum_{i=t}^T c_{ki} & t = k, \ldots, T. \end{cases}$$

We have an analog to Theorem 6.

Theorem 16. *F^{BLN} is given by the set of $(\mu, x, y) \in \mathbf{R} \times X^N \times Y$ which satisfy (45) and*

$$\mu \ge \sum_{t=1}^T \hat{\delta}_t^k (x_t - x_{t-1}) - \sum_{s=1}^T M \hat{\alpha}_s^k y_s \qquad \forall k \in \mathcal{T}. \tag{56}$$

Proof. Let

$$GN = \big\{(\mu, x, y) \in \mathbf{R} \times X^N \times Y : (45) \text{ and } (56)\big\}$$

so that we must prove $GN = F^{BLN}$. We first show each inequality (56) is valid for F^{BLN} and hence $F^{BLN} \subseteq GN$. Consider inequality (56) for a $k \in \mathcal{T}$. This inequality corresponds to an inequality of the form (52), with $\delta_t = \hat{\delta}_t^k$ for $t \in \mathcal{T}$ and $\alpha_s = \hat{\alpha}_s^k$ for $s \in \mathcal{T}$. Thus, we verify that $(\hat{\delta}^k, \hat{\alpha}^k)$ is feasible to BLLP-P and hence yields a valid inequality by Lemma 14. It is immediate that $\hat{\delta}^k \ge 0$ and $\hat{\alpha}^k \ge 0$. Now consider a fixed s and t such that $1 \le s \le t \le T$. If $s > k$ then also $t > k$ and hence

$$\hat{\delta}_t^k - \hat{\alpha}_s^k \le \sum_{i=t}^T c_{ki} - \Big(\sum_{i=t}^T c_{ki} - \sum_{i=t}^T c_{si}\Big) = \sum_{i=t}^T c_{si}.$$

If $s = k$ then $t \ge k$ and hence

$$\hat{\delta}_t^k - \hat{\alpha}_s^k = \sum_{i=t}^T c_{si}.$$

If $s < k$ and $t \ge k$ then

$$\hat{\delta}_t^k - \hat{\alpha}_s^k \le \sum_{i=t}^T c_{ki} - \Big(\sum_{i=t}^T c_{ki} - \sum_{i=t}^T c_{si}\Big) = \sum_{i=t}^T c_{si}.$$

Finally, if $s < k$ and $t < k$ then

$$\hat{\delta}_t^k - \hat{\alpha}_s^k \le \hat{\alpha}_s^k + \sum_{i=t}^T c_{si} - \hat{\alpha}_s^k = \sum_{i=t}^T c_{si}$$

and so $(\hat{\delta}^k, \hat{\alpha}^k)$ is feasible to BLLP-P.

Next let $(\mu, x, y) \in GN$. We show (27) is satisfied, and hence $(\mu, x, y) \in F^{BLN}$. Let $k \in \mathcal{T}$ be the period such that $y_k = 1$ (if $y = x = \mathbf{0}$ (27) is trivially satisfied). Because $x_t = 0$ for $t = 1, \ldots, k-1$ and $y_s = 0$ for all $s \neq k$ inequality (56) for this k yields

$$\begin{aligned}
\mu &\geq \sum_{t=k}^{T} (x_t - x_{t-1}) \sum_{i=t}^{T} c_{ki} \\
&= \sum_{t=k}^{T} (x_t - x_{t-1}) \sum_{i=t}^{T} c_{ki} \\
&= \sum_{i=k}^{T} c_{ki} \sum_{t=k}^{i} (x_t - x_{t-1}) = \sum_{i=k}^{T} c_{ki} x_i = \sum_{s=1}^{T} y_s \sum_{t=s}^{T} c_{st} x_t
\end{aligned}$$

hence (27) is satisfied. □

The coefficients $(\hat{\delta}^k, \hat{\alpha}^k)$ take on a simpler form in the case of "improving technology," that is, if $c_{si} \geq c_{ti}$ whenever $s \leq t \leq i$. In this case, for each k, $\hat{\alpha}^k_s = 0$ for $s = 1, \ldots, k$ and

$$\hat{\alpha}^k_s = \sum_{i=s}^{T} c_{ki} - \sum_{i=s}^{T} c_{si}$$

and $\hat{\delta}^k_t = \sum_{i=t}^{T} c_{ti}$ for $t = 1, \ldots, k-1$ and $\hat{\delta}^k_t = \sum_{i=t}^{T} c_{ki}$ for $t = k, \ldots, T$.

Theorem 16 yields a new mixed-integer linear formulation which is valid when the activities are restricted to be nondecreasing. We refer to this formulation as LBLN. This formulation can be strengthened by adding violated inequalities obtained from solving BLLP-P.

As an alternative or supplement to adding inequalities obtained from solving BLLP-P we can add inequalities which are valid for the concave formulation CM. In the next section we will see that with the addition of the constraints $x \in X^N$, the convex hull of the CM formulation is given by a single valid inequality, (58). Although this inequality does not involve the binary variables y, this inequality is still valid for F^{BLN}. Therefore, by adding this inequality to the LBLN formulation, we will obtain a linear relaxation that yields a bound as strong as possible from a single activity. The disadvantage of this relative to using inequalities obtained from solving BLLP-P which involve the binary variable y is that (58) will not be strengthened by branching on the y variables. This disadvantage can be overcome at the expense of a more cumbersome implementation by adding the branching constraints (60) of the CM formulation as cuts which are locally valid at each node in the

branch-and-bound tree. An advantage of using the LBLN formulation strengthened with (58) over just using the more compact CM formulation is that by using the binary variables we are not required to implement a specialized branching mechanism. In addition, the presence of the binary variables may allow for generation of general purpose cutting planes which consider more of the problem than a single activity.

2.6.4 Concave Formulation

The branch-and-cut method of Section 2.5 can still be used to solve formulation CM when we add the constraints $x \in X^N$. However, by using the nondecreasing restriction, we can significantly improve the lower bounds. Thus, we seek valid inequalities for the set

$$E^N = \big\{(\mu, x) \in \mathbf{R} \times X^N : \mu \geq h(x)\big\}$$

or equivalently, lower bounds on $h(x)$ which are valid for $x \in X^N$.

Note that the extreme points of X^N are given by the points $M^t \in \mathbf{R}^T$ for $t = 1, \ldots, T, T+1$ with $M^t_s = 0$ for $s = 1, \ldots, t-1$ and $M^t_s = M$ for $s = t, \ldots, T$. We then have the following lower bound on h.

Theorem 17. *If $x \in X^N$, then*

$$h(x) \geq \sum_{t=1}^{T} \big(h(M^t) - h(M^{t+1})\big) x_t / M. \tag{57}$$

Proof. Let us evaluate the right-hand side of (57) for $x = M^k$ for some $k \in \{1, \ldots, T, T+1\}$.

$$\sum_{t=1}^{T} \big(h(M^t) - h(M^{t+1})\big) x_t / M = \sum_{t=k}^{T} \big(h(M^t) - h(M^{t+1})\big) = h(M^k).$$

Thus, the lower bound is valid for all extreme points of X^N. The result then follows from concavity of h, Theorem 7. □

Thus, we can strengthen the CM formulation by adding the inequality

$$\mu \geq \sum_{t=1}^{T} \big(h(M^t) - h(M^{t+1})\big) x_t / M \tag{58}$$

Combined with the inequalities that define the feasible region, this single inequality is sufficient to define the convex hull of E^N.

Theorem 18.

$$\operatorname{conv}(E^N) = \left\{(\mu, x) \in \mathbf{R} \times X^N : (58)\right\}.$$

Proof. Let

$$lb(x) = \sum_{t=1}^{T} \big(h(M^t) - h(M^{t+1})\big) x_t / M$$

and let $H = \left\{(\mu, x) \in \mathbf{R} \times X^N : \mu \geq lb(x)\right\}$ so that we must prove $\operatorname{conv}(E^N) = H$. We immediately have $\operatorname{conv}(E^N) \subseteq H$ by validity of the lower bound (57).

The converse is also immediate since the extreme points of H are given by $(lb(M^t), M^t)$ for $t = 1, \ldots, T, T{+}1$, and we have seen in the proof of Theorem 17 that $lb(M^t) = h(M^t)$ so that these points also satisfy $lb(M^t) \geq h(M^t)$ and hence are feasible to E^N. This completes the proof since the ray $(1, \mathbf{0})$ is the only feasible direction for both H and E^N. □

In the context of the branch-and-bound algorithm described in Section 2.5, we seek lower bounds on $h(x; l, u)$; the cost function subject to the restriction that the start-time s be in the set $\{l, \ldots, u\}$ for $1 \leq l \leq u \leq T + 1$. In particular, we seek lower bounds that are valid and tight under the additional restriction that $x \in X^N$. Recall the definition

$$\underline{f}(l, u) = \min\left\{f_s : s = l, \ldots, u\right\}.$$

We then have the following lower bound.

Theorem 19. *Let $1 \leq l \leq u \leq T{+}1$ and $x \in X^N$ be such that $x_t = 0$ for all $t = 1, \ldots, l-1$. Then,*

$$h(x; l, u) \geq \underline{f}(l, u) + \sum_{t=l}^{T} \big(h(M^t; l, u) - h(M^{t+1}; l, u)\big) x_t / M. \tag{59}$$

Furthermore, the above inequality is satisfied at equality if either $l = u$, or $x = M^k$ for some $k \geq l$.

Proof. We first prove the inequality is satisfied at equality for $x = M^k$ for any $k \geq l$. Fix some $k \geq l$. The right-hand side of the inequality becomes

$$\underline{f}(l, u) + \sum_{t=k}^{T} \big(h(M^t; l, u) - h(M^{t+1}; l, u)\big) = \underline{f}(l, u) + h(M^k; l, u) - \underline{f}(l, u) = h(M^k; l, u)$$

where the first equality follows since $h(M^{T+1}; l, u) = \underline{f}(l, u)$. Now, note that under the restriction on the start time $s \geq l$ the extreme points in the x space are given by M^k for

$k = l, \ldots, T$. The validity of the lower bound then follows from concavity of $h(\cdot; l, u)$, which can be established in a manner identical to the proof that h is concave. Finally, if $l = u$, then the right-hand side of (59) becomes

$$\begin{aligned}
&\underline{f}(l,l) + \sum_{t=l}^{T} \big(h(M^t; l, l) - h(M^{t+1}; l, l)\big) x_t / M \\
&\qquad = \underline{f}(l,l) + \sum_{t=l}^{T} \big[f_l + M \sum_{k=t}^{T} c_{lk} - \big(f_l + M \sum_{k=t+1}^{T} c_{lk}\big)\big] x_t / M \\
&\qquad = f_l + \sum_{t=l}^{T} c_{lt} x_t = h(x; l, l).
\end{aligned}$$

□

Hence, at a node in which we have restricted $s \in \{l, \ldots, u\}$ we add the inequality

$$\mu \geq \underline{f}(l,u) + \sum_{t=l}^{T} \big(h(M^t; l, u) - h(M^{t+1}; l, u)\big) x_t / M \tag{60}$$

which is valid under this restriction. By Theorem 19 we know that if $l = u$, this inequality will force the cost to be correct. Also, using arguments similar to those in the proof of Theorem 18 we can show that this inequality defines the convex hull of the single activity relaxation under this restriction.

2.7 Computational Results

We performed computational tests to compare the different formulations and to investigate the effect of using the valid inequalities. We tested six different formulations: WL, LS, LS.C, LBL, LBL.C and CM. WL refers to the Weak Linearization formulation of Section 2.4.2. LS refers to the lot sizing inspired formulation of Section 2.4.2.1 and LBL refers to the formulation presented in Section 2.4.3. LS.C and LBL.C refer to the LS and LBL formulations, using the valid inequalities (18) and (28) respectively. CM refers to the concave minimization formulation of Section 2.5, solved with the specialized branch-and-cut algorithm. We did not test the extended formulation which is not practical for large instances.

Table 1 summarizes the number of variables and rows, not including cuts, in each of the formulations. For the extended formulation EF, we list only the approximate size, to

emphasize that it is quadratic in the number of periods. The number of rows in the LS formulations includes the inequalities (11) which are not necessary for the LS formulation, but yield significantly better computational results. The constraint matrices in all formulations

Table 1: Sizes of the different formulations.

Formulation	# Variables	# Rows
EF	$O(T^2\|A\|)$	$O(T^2\|A\|)$
LS	$3T\|A\|$	$(3T+1)\|A\|$
LBL	$(2T+1)\|A\|$	$(2T+1)\|A\|$
CM	$(T+1)\|A\|$	$T\|A\|$

are quite sparse, having $O(T^2|A|)$ nonzeros, even in the extended formulation.

In Section 2.7.1 we describe the instances used for the tests. In Section 2.7.2 we make some comments about the implementation of our methods. In Section 2.7.3 we present the results on instances which do not have the nondecreasing constraint, and in Section 2.7.4 we present results on the same set of instances, but with the nondecreasing constraint present.

2.7.1 Test Instances

We conducted our tests on instances of the production and distribution planning problem presented in the introduction. For this problem, the activities consist of production and distribution in a transportation network with $|I|$ supply points, $|J|$ demand points and arcs between all supply and demand points. Therefore, $|A| = |I|(|J|+1)$. We randomly generated instances which have characteristics similar to data in the application that motivated this work. The variable costs for these instances depends only on the start-time, and not on the period in which the activity occurs, so that the formulations of Section 2.4.2 apply. In all cases, the variable costs decrease at a constant rate as the start period is delayed. Fixed costs for installing technology were not considered in the motivating application, but because fixed costs may be present in other applications we generated instances with and without fixed costs. For instances with fixed costs, the fixed cost does not depend on the start period.

2.7.2 Implementation Comments

We used CPLEX 9.0 as our mixed-integer programming solver, and implemented the addition of valid inequalities using CPLEX cut callback routines. For the implementation of the specialized branch-and-cut algorithm to solve the CM formulation we used CPLEX to solve the linear programming relaxations, select nodes to explore and manage the branch-and-cut tree. We implemented our custom branching strategy using the branch callback routine provided by CPLEX.

We let CPLEX generate the cuts it generates by default, and in particular, because of the network structure of our test instances, CPLEX was able to generate many flow cover inequalities for the LS and LBL formulations.

We investigated different strategies for using the valid inequalities we have developed, including generating locally and globally valid inequalities at nodes throughout the search tree, and at varying frequencies. However, we found that the simple strategy of generating globally valid inequalities at the root node was most effective, and therefore this is the strategy we used.

For the production and distribution planning application, we are deciding if and when to install technology at supply points as well as on distribution arcs in the network. Because the decisions of when to start a supply node affect the decisions of when to start distribution on arcs from that node, it makes sense to put priority on these decisions, and we have done so in our implementations. For the LS and LBL formulations, this is done by giving the corresponding binary variables appropriate priority levels in CPLEX. When solving the CM formulation the activity selection criterion we use favors selection of a supply activity over a distribution activity. The criterion is simply to branch on the activity which has the maximum disparity between the actual cost of the activity over the horizon $h_a(x)$ and the lower bound on the cost given by the current value of μ_a. This tends to favor selection of supply activities because the supply activity levels are generally larger than the distribution activity levels (since one supply output can be split on many distribution arcs).

2.7.3 Results

We first conducted tests on a set of small instances to compare the solution times and number of nodes required to solve instances to optimality. For these instances we used a time limit of one hour. We used a set of 30 instances, half with and half without fixed costs. Table 2 lists each of the different sizes of instances in this test set and the geometric average time, taken over five instances of each size, to solve these instances to optimality. For the WL formulation, none of the instances could be solved to optimality within the one hour time limit, so we report the average remaining optimality gaps and the average gap between the best solution found by WL and the optimal solution (UB Gap). The average remaining optimality gaps with the WL formulation were huge, 57% on average. More significantly from a practical standpoint, the best solution found within the hour time limit was 6.1% more costly than the optimal solution on average. In contrast, Tables 2 and 3 indicate that these instances could be solved in minutes using the LS, LS.C, LBL and LBL.C formulations. An asterisk next to an item in these tables indicates that the reported average is a lower bound on the true average since some of the instances included were not solved to optimality in an hour (this only occurs for CM). Table 3 shows summary statistics for 30 instances, including the geometric averages of the number of nodes and time required to solve the instances. It also shows the average and maximum ratio of the solution time for each formulation to the best solution time over all the formulations, as well as the number of instances in which each formulation had the best time.

Table 2: Results for small instances.

		WL Gaps		Average Time (s)				
FC?	$(\|I\|, \|J\|, T)$	Opt	UB	LS	LS.C	LBL	LBL.C	CM
No	(10,5,10)	78.9%	1.4%	30.1	28.3	18.4	16.2	*54.0
	(15,5,10)	76.8%	1.5%	46.4	45.1	41.0	28.6	*497.9
	(10,10,10)	55.6%	0.9%	70.5	50.9	52.8	35.0	*506.3
Yes	(10,5,10)	46.0%	10.3%	16.6	20.8	6.5	6.6	*227.9
	(15,5,10)	49.6%	11.3%	162.0	196.5	81.1	89.9	*1577.3
	(10,10,10)	38.8%	11.2%	47.4	50.8	24.6	23.7	*373.3

* Some times were capped at one hour: average is a lower bound.

Tables 2 and 3 indicate that for the LS and LBL formulations on small instances, using

Table 3: Summary results for 30 small instances.

	LS	LS.C	LBL	LBL.C	CM
Ave time (s)	48.2	48.8	28.3	24.7	*349.7
Ave ratio to best time	2.27	2.36	1.35	1.09	*47.4
Max ratio to best time	4.91	5.16	5.23	1.55	*367.2
# Best time	0	2	14	14	0
Ave nodes	626	407	1042	407	*38236

the new cuts reduces the number of nodes on average over all instances, slightly reduces the average solution time for instances without fixed costs, but does not help the solution times for instances with fixed costs. These tables also indicate that LBL.C tends to solve these sized instances to optimality significantly faster than the alternatives, whereas solving CM takes significantly longer than the alternatives.

We also tested the formulations, excluding the WL formulation, on a set of 90 large instances, 45 without and 45 with fixed costs. We used a time limit of two hours for these instances, and none of the formulations were able to provably solve any of these instances to optimality within this time limit.

Table 4 lists the instance sizes and the average optimality gap obtained within the time limit using the different formulations. Each entry in this table is an average over five instances of the size given in the row. For the LS formulation, the root LP relaxation was not solved within the time limit for one of the largest instances with fixed costs, and so the average gap reported is over the other four instances. From this table we observe that the LBL.C formulation yields the smallest optimality gap after two hours in almost all cases. In addition, using the valid inequalities in the LBL formulation usually reduces the optimality gap obtained after two hours. In contrast to this, using the valid inequalities in the LS formulation does not always reduce the optimality gap. The explanation for this is that in the larger LS formulation, the additional time spent solving the linear programming relaxations when using the valid inequalities outweighs the improved lower bounds obtained by using them. An important observation to make from this table is that although these formulations could not solve these large instances to optimality within two hours, the optimality gaps were usually reasonable, with most being not much larger than one percent. The instances

Table 4: Average optimality gaps for large instances.

		Average Optimality Gap				
FC?	$(\|I\|, \|J\|, T)$	LS	LS.C	LBL	LBL.C	CM
No	$(50, 20, 10)$	0.39%	0.36%	0.47%	0.32%	0.67%
	$(75, 20, 10)$	0.25%	0.21%	0.29%	0.19%	0.50%
	$(100, 20, 10)$	0.26%	0.19%	0.33%	0.18%	0.54%
	$(50, 20, 15)$	0.58%	0.44%	0.70%	0.44%	0.77%
	$(75, 20, 15)$	0.55%	0.73%	0.75%	0.40%	0.77%
	$(100, 20, 15)$	0.86%	0.71%	0.71%	0.44%	0.85%
	$(50, 20, 20)$	2.42%	2.38%	1.88%	1.25%	1.28%
	$(75, 20, 20)$	1.58%	1.54%	1.50%	0.86%	1.18%
	$(100, 20, 20)$	1.72%	1.87%	1.92%	1.26%	1.36%
Yes	$(50, 20, 10)$	0.62%	0.48%	0.60%	0.56%	0.91%
	$(75, 20, 10)$	0.51%	0.44%	0.51%	0.52%	0.82%
	$(100, 20, 10)$	0.44%	0.59%	0.52%	0.47%	0.81%
	$(50, 20, 15)$	0.94%	2.27%	1.01%	1.01%	1.36%
	$(75, 20, 15)$	1.65%	1.88%	0.89%	0.83%	1.17%
	$(100, 20, 15)$	2.12%	2.05%	0.79%	0.74%	1.39%
	$(50, 20, 20)$	5.32%	5.26%	1.86%	1.85%	2.96%
	$(75, 20, 20)$	5.58%	5.58%	2.31%	2.60%	4.17%
	$(100, 20, 20)$	*3.39%	*3.42%	3.34%	3.31%	3.98%

* Average is for four of the five instances at that size.

with fixed costs and $T = 20$ periods are an exception, with average optimality gaps in the 3-4% range in the best cases. It is also evident from Table 4 that the instances with fixed costs are more difficult to solve than those without fixed costs.

Table 5: Results for 45 large instances without fixed costs.

	LS	LS.C	LBL	LBL.C	CM
Ave % above UB^*	0.34%	0.39%	0.25%	0.11%	0.03%
Max % above UB^*	2.48%	2.33%	2.00%	1.22%	0.51%
# Best UB	2	5	1	16	22
Ave LB gap to UB^*	0.62%	0.55%	0.71%	0.49%	0.85%
Max LB gap to UB^*	1.63%	1.80%	1.58%	1.39%	1.66%
# Best LB	0	4	0	43	0
Root LP time	554.1	558.4	195.0	193.7	13.2

Tables 5 and 6 present summary statistics for the instances without and with fixed costs, respectively. In these tables we report for each of the formulations the average and maximum percent by which the cost of the best solution found by the formulation exceeds UB^*, the value of the best solution found over the six formulations. We also indicate the number of

instances for which each formulation found the best solution. These results indicate that for large instances without fixed costs the concave formulation CM is most effective for finding good feasible solutions, whereas for the instances with fixed costs the LBL formulations (with and without cuts) are most effective. These results may be explained by looking at the average times to solve the root LP using formulation CM. For the instances without fixed costs, the root LP of CM solves much more quickly than the alternatives, allowing it to search many more nodes for improved feasible solutions within the two hour time limit. On the other hand, for the instances with fixed costs, the root LP time is faster for formulation CM, but not so much faster as to outweigh the benefits of having binary variables present in the LBL formulation, which enables CPLEX to use its heuristics to search for good solutions. The results on the root LP solve times suggest that CM may be the only viable formulation to approximately solve even larger instances than the ones we have presented. For example, for the largest instances with fixed costs, with $|I| = 100, |J| = 20, T = 20$, the geometric average time to solve the LBL relaxation was 1823 seconds, whereas the average for CM was 259 seconds.

Table 6: Results for 45 large instances with fixed costs.

	LS	LS.C	LBL	LBL.C	CM
Ave % above UB^*	*1.21%	*1.43%	0.11%	0.15%	0.61%
Max % above UB^*	*5.46%	*5.83%	2.04%	3.05%	4.71%
# Best UB	8	7	17	16	1
Ave LB gap to UB^*	*1.10%	*1.06%	1.21%	1.18%	1.36%
Max LB gap to UB^*	*4.01%	*4.01%	3.92%	3.89%	4.04%
# Best LB	1	36	0	8	1
Root LP time	*1374.9	*1379.3	245.2	245.1	77.0

* Based on 44 of the 45 instances.

Tables 5 and 6 also show for each formulation the average and maximum percentage gap between the lower bound provided by that formulation and UB^*, defined as the difference between UB^* and the lower bound, divided by UB^*. These results explain why, despite having good performance for finding good feasible solutions, CM yields worse average optimality gaps as shown in Table 4. The lower bounds obtained using CM are significantly worse than those obtained using the other formulations. The primary reason for this is that

this formulation does not have any binary variables, and so CPLEX cannot generate any general purpose cutting planes, such as the flow cover inequalities.

2.7.4 Nondecreasing Special Case

We also conducted experiments using the formulations we have developed to solve instances in which the activities (flows and production quantities in the network) are restricted to be nondecreasing. The instances for these tests are just the instances used for the tests described in Section 2.7.3, with the addition of the nondecreasing constraints. Note that the formulations in this case have $O(T|A|)$ more rows than specified in Table 1.

As in Section 2.7.3, we first conducted tests on a set of small instances. We initially tested the formulations developed in Sections 2.4 and 2.5 which do not use the nondecreasing restriction to strengthen the formulations. Table 7 gives the results. The results are qualitatively similar to those obtained for the small instances without the nondecreasing restriction, although the computation times with the new formulations are longer. In particular, most of the instances could be solved to optimality within the one hour time limit using formulations LS and LBL, whereas the weak linearization formulation left huge optimality gaps.

Table 7: Results for small nondecreasing instances.

		WL Gaps		Average Time (s)				
FC?	$(\|I\|,\|J\|,T)$	Opt	UB	LS	LS.C	LBL	LBL.C	CM
No	(10,5,10)	77.23%	1.51%	54.3	66.9	54.8	40.5	*116.8
	(15,5,10)	75.29%	2.48%	*400.4	*344.5	*276.5	*159.8	*620.9
	(10,10,10)	56.06%	1.16%	*449.1	*350.3	*329.7	*310.2	*648.2
Yes	(10,5,10)	39.34%	8.03%	22.3	25.6	15.1	18.4	121.4
	(15,5,10)	47.35%	11.76%	*209.0	*236.4	211.0	210.7	*474.8
	(10,10,10)	35.80%	9.66%	45.2	43.9	47.8	48.3	*199.0

* Some times were capped at one hour: average is a lower bound.

We next tested the formulations developed in Section 2.6 which are strengthened by using the nondecreasing restriction. These results are given in Tables 8 and 9. Here, LSN refers to the lot sizing formulation, with the stronger inequalities (49) used in place of (16). LSN.C refers to formulation LSN augmented with the inequalities of Theorem 13 added at

the root node. LBLN refers to the formulation derived in Theorem 16. Two strengthened versions of LBLN were tested: LBLN.S adds the inequality (58) which was developed for the concave formulation and LBLN.C adds the inequalities of Lemma 14 as violated at the root node. Finally, CMN refers to the concave minimization formulation, in which (58) is used in the initial relaxation, and (60) is used when r(anching). It is clear that strengthening

Table 8: Results for small nondecreasing instances using nondecreasing restriction.

		Average Time (s)					
FC?	$(\|I\|, \|J\|, T)$	LSN	LSN.C	LBLN	LBLN.S	LBLN.C	CMN
No	(10,5,10)	8.3	7.6	3.8	4.2	10.1	2.2
	(15,5,10)	11.4	11.7	5.0	4.9	12.8	3.3
	(10,10,10)	19.5	17.3	9.3	8.9	20.9	10.0
Yes	(10,5,10)	23.3	20.7	14.9	11.5	23.8	11.2
	(15,5,10)	135.0	139.6	115.8	93.0	143.4	51.7
	(10,10,10)	30.2	28.3	34.6	14.7	38.2	23.6

the formulations using the nondecreasing restriction yields significantly faster computation times. In addition, we see that in contrast to the case without the nondecreasing condition, the concave formulation CMN often yields the smallest computation times despite requiring more nodes to solve the problems. When using CMN, solving each node relaxation is much faster due to the compactness of the formulation and the fact that only a single inequality (58) is required to obtain a reasonable lower bound. Moreover, although more nodes are required to solve CMN than the integer linear formulations, the number is not significantly more, as was the case when the nondecreasing restriction was not present. This may be explained by the fact that we add (60) at each node, which yields the convex hull of the single activity formulation subject to the additional conditions imposed by branching.

Table 9: Summary results for 30 small nondecreasing instances.

	LSN	LSN.C	LBLN	LBLN.S	LBLN.C	CMN
Ave time (s)	23.6	22.4	9.9	10.5	17.1	10.0
Ave ratio to best	3.9	3.8	1.7	1.8	4.1	2.4
Max ratio to best	10.4	9.8	4.0	4.8	18.0	21.8
# Best time	0	0	4	6	3	17
Ave nodes	210.2	161.0	267.3	266.8	239.2	1803.0

We next tested the formulations which were strengthened using the nondecreasing restriction on 90 large instances, 45 with and 45 without fixed costs. The average optimality obtained gaps after the two hour time limit are reported in Table 10. From these tables we observe that for the relatively smaller instances, those with $T \leq 15$, all the formulations did quite well, usually yielding optimality gaps of less than one percent. We also observe that for both the LSN and the LBLN formulations, adding cuts to strengthen the formulation did not significantly affect the results. It appears that the strengthened formulations obtained using the nondecreasing condition yield good formulations without the need to add additional cuts.

Table 10: Average optimality gaps for large nondecreasing instances.

		Average Optimality Gap					
FC?	$(\|I\|, \|J\|, T)$	LSN	LSN.C	LBLN	LBLN.S	LBLN.C	CMN
No	$(50, 20, 10)$	0.24%	0.22%	0.21%	0.20%	0.20%	0.36%
	$(75, 20, 10)$	0.20%	0.20%	0.17%	0.18%	0.17%	0.31%
	$(100, 20, 10)$	0.26%	0.27%	0.22%	0.23%	0.22%	0.31%
	$(50, 20, 15)$	0.42%	0.41%	0.41%	0.40%	0.40%	0.54%
	$(75, 20, 15)$	0.56%	0.54%	0.63%	0.58%	0.52%	0.53%
	$(100, 20, 15)$	0.65%	0.80%	0.49%	0.50%	0.42%	0.40%
	$(50, 20, 20)$	1.75%	2.00%	1.14%	1.22%	1.36%	0.86%
	$(75, 20, 20)$	2.36%	3.04%	0.93%	1.32%	1.26%	0.67%
	$(100, 20, 20)$	*2.94%	*2.83%	1.26%	1.26%	1.64%	0.50%
Yes	$(50, 20, 10)$	0.51%	0.54%	0.44%	0.43%	0.43%	0.70%
	$(75, 20, 10)$	0.66%	0.70%	0.57%	0.55%	0.53%	0.70%
	$(100, 20, 10)$	0.69%	0.69%	0.58%	0.59%	0.62%	0.67%
	$(50, 20, 15)$	1.17%	1.25%	1.24%	1.17%	1.21%	1.22%
	$(75, 20, 15)$	4.55%	4.58%	1.05%	1.05%	1.04%	1.00%
	$(100, 20, 15)$	6.75%	6.75%	0.94%	0.94%	0.93%	0.89%
	$(50, 20, 20)$	12.44%	12.77%	9.94%	10.53%	9.19%	1.96%
	$(75, 20, 20)$	**	**	14.34%	14.18%	13.15%	1.63%
	$(100, 20, 20)$	**	**	10.73%	10.65%	9.55%	1.24%

* Average is for four of the five instances at that size.

** The root linear program was not solved for any of the instances at that size.

In Tables 11 - 13 we present summary statistics for these 90 instances. Table 11 presents the results for the 45 instances without fixed costs. From this table we see that the CMN formulation is clearly the best at generating good feasible solutions. In the worst case, the solution it generated was 0.03% worse than the best solution generated. However, the LBLN

formulation was also not bad, generating solutions which were on average only 0.22% more costly than the best generated. We also observe that the LBLN formulations yield the best lower bounds, but that the lower bounds from the CMN formulation are not significantly worse. The summary results for the smaller instances with fixed costs, those with $T = 10$ or $T = 15$ are presented in Table 12. These results are qualitatively similar to the results for the instances without fixed costs. One difference is that advantage possessed by the CMN formulation in time to solve the root LP is greater for the instances with fixed costs. Finally, in Table 13 we present the summary results for the 15 instances with fixed costs and $T = 20$. We separated these results from those with $T \leq 15$ because for these instances the LSN and LBLN formulations spent the majority of the two hour time limit just solving the root LP relaxation, and hence these formulations did not have time to find good solutions. In fact, the larger LSN formulation was only able to solve the root relaxation in the two hour limit for the five of the 15 instances with $|I| = 50$. In contrast, the CMN formulation solves the root LP relaxation much more quickly and was therefore still able to find good solutions. As a result, the CMN formulation yielded the best solution in all cases, and the best solution found by the other formulations was significantly worse.

Table 11: Summary results for 45 large nondecreasing instances without fixed costs.

	LSN	LSN.C	LBLN	LBLN.S	LBLN.C	CMN
Ave % above UB^*	*0.61%	*0.73%	0.22%	0.27%	0.30%	0.00%
Max % above UB^*	*3.41%	*3.67%	1.41%	1.95%	2.04%	0.03%
# Best UB	1	4	0	0	5	36
Ave LB gap to UB^*	*0.40%	*0.40%	0.39%	0.39%	0.39%	0.50%
Max LB gap to UB^*	*1.09%	*1.09%	1.08%	1.11%	1.10%	1.24%
# Best LB	1	2	16	14	12	0
Root LP time	*821.6	*804.6	304.4	321.4	305.4	109.6

* Based on 44 of the 45 instances.

2.7.5 Comparison of General and Nondecreasing Results

We now compare the computational results obtained with and without the nondecreasing restriction. Table 14 presents the average optimality gaps obtained after the two hour time limit (as in Tables 4 and 10) using the LBL formulation with and without cuts and

Table 12: Summary results for 30 large nondecreasing instances with fixed costs and $T \leq 15$.

	LSN	LSN.C	LBLN	LBLN.S	LBLN.C	CMN
Ave % above UB^*	1.79%	1.83%	0.14%	0.13%	0.13%	0.00%
Max % above UB^*	10.90%	10.90%	0.62%	0.41%	0.38%	0.02%
# Best UB	0	0	2	1	1	27
Ave LB gap to UB^*	0.71%	0.71%	0.66%	0.66%	0.66%	0.86%
Max LB gap to UB^*	1.16%	1.14%	1.04%	1.05%	1.05%	1.35%
# Best LB	0	1	12	11	6	0
Root LP time	1163.2	1131.0	378.3	372.3	379.8	82.3

Table 13: Summary results for 15 large nondecreasing instances with fixed costs and $T = 20$.

	LSN	LSN.C	LBLN	LBLN.S	LBLN.C	CMN
Ave % above UB^*	*12.5%	*12.7%	11.8%	11.9%	10.6%	0.0%
Max % above UB^*	*16.2%	*14.6%	20.5%	17.2%	20.5%	0.0%
# Best UB	0	0	0	0	0	15
Ave LB gap to UB^*	*1.71%	*1.71%	1.43%	1.42%	1.43%	1.61%
Max LB gap to UB^*	*2.39%	*2.40%	2.44%	2.41%	2.42%	2.96%
# Best LB	2	0	5	2	6	0
Root LP time	*3609.6	*3484.0	2587.4	2770.2	2561.0	419.4

* Based on the 5 of the 15 instances with $|I| = 50$.

the CM formulation. The LS formulation is excluded from this comparison because in most cases it does not yield competitive results. The results are generally similar in that

Table 14: Comparison of results with and without nondecreasing constraint.

		General Activities			Nondecreasing Activities		
FC?	$(\|I\|, \|J\|, T)$	LBL	LBL.C	CM	LBLN	LBLN.C	CMN
No	$(50, 20, 10)$	0.47%	0.32%	0.67%	0.21%	0.20%	0.36%
	$(75, 20, 10)$	0.29%	0.19%	0.50%	0.17%	0.17%	0.31%
	$(100, 20, 10)$	0.33%	0.18%	0.54%	0.22%	0.22%	0.31%
	$(50, 20, 15)$	0.70%	0.44%	0.77%	0.41%	0.40%	0.54%
	$(75, 20, 15)$	0.75%	0.40%	0.77%	0.63%	0.52%	0.53%
	$(100, 20, 15)$	0.71%	0.44%	0.85%	0.49%	0.42%	0.40%
	$(50, 20, 20)$	1.88%	1.25%	1.28%	1.14%	1.36%	0.86%
	$(75, 20, 20)$	1.50%	0.86%	1.18%	0.93%	1.26%	0.67%
	$(100, 20, 20)$	1.92%	1.26%	1.36%	1.26%	1.64%	0.50%
Yes	$(50, 20, 10)$	0.60%	0.56%	0.91%	0.44%	0.43%	0.70%
	$(75, 20, 10)$	0.51%	0.52%	0.82%	0.57%	0.53%	0.70%
	$(100, 20, 10)$	0.52%	0.47%	0.81%	0.58%	0.62%	0.67%
	$(50, 20, 15)$	1.01%	1.01%	1.36%	1.24%	1.21%	1.22%
	$(75, 20, 15)$	0.89%	0.83%	1.17%	1.05%	1.04%	1.00%
	$(100, 20, 15)$	0.79%	0.74%	1.39%	0.94%	0.93%	0.89%
	$(50, 20, 20)$	1.86%	1.85%	2.96%	9.94%	9.19%	1.96%
	$(75, 20, 20)$	2.31%	2.60%	4.17%	14.34%	13.15%	1.63%
	$(100, 20, 20)$	3.34%	3.31%	3.98%	10.73%	9.55%	1.24%

most of the instances could be solved to within a reasonable optimality gap within two hours. One difference is that adding the new valid inequalities to the LBL formulation without the nondecreasing condition is very helpful, whereas they are not helpful when the nondecreasing constraint is present. A possible explanation for this is that the LBLN formulation has already been strengthened somewhat using the nondecreasing restriction, so that the incremental strengthening obtained from adding more valid inequalities does not outweigh the increased time to solve the linear programming relaxations. Another difference is that the concave formulation performs better relative to the LBL formulation in the case of nondecreasing activities, particularly for the instances with long planning horizon T. This can be explained partially because the convex hull of the concave formulation is given by a single additional inequality in the nondecreasing case, whereas many cuts have to be added to obtain a strong relaxation for the concave formulation in the general activities

case. Adding these additional cuts can somewhat offset the compactness advantage of the concave formulation. In addition, the formulations with the nondecreasing restriction have more rows due to the nondecreasing constraints, and hence, given the same two hour time limit, it will be relatively more important to be using a more compact formulation.

2.8 Concluding Remarks

We have studied a strategic planning model which addresses the question of when to install technology in an environment in which technology is improving over time. A natural formulation of this model leads to a mixed-integer program with bilinear objective. We have developed a series of progressively more compact formulations that can be used to solve large-scale instances. Some of the results we have developed to strengthen these formulations have been implemented in practice and have led to significant reductions in computation time.

Our computational study indicates that a simple linearization of the bilinear objective yields a formulation with very weak bounds, and hence fails to solve even small instances. However, strengthening this formulation by using ideas from lot sizing enables small instances to be solved to optimality quickly, and reasonable optimality bounds to be obtained for large instances. Studying the mixed-integer bilinear program directly leads to a new mixed-integer linear formulation which introduces significantly fewer linearization variables. Using this formulation leads to reduced computation times for the small instances, and improved optimality gaps within the specified time limit for large instances. In particular, this more compact formulation allows better feasible solutions to be found. Implementation of valid inequalities we have developed for this formulation can also lead to improved computational times. Finally, we have presented a specialized branch-and-cut algorithm which solves an extremely compact concave minimization formulation, and hence can be used to generate good solutions and optimality bounds for large-scale instances. In particular, for very large-scale instances, just solving the relaxation of the mixed-integer linear formulations may take prohibitively long, so that using this formulation may be the only viable option.

In this study, we began by assuming only that the activity levels are constrained by an upper bound, common over all time periods. However, our approach can be used when more general constraints on the activity levels are present. One example of such a constraint, which was present in a variant of the application that motivated this work, is that the activities are restricted to be nondecreasing over time, and we have seen in Section 2.6 that the formulations can be strengthened by using this additional restriction. The computational results indicate that using these strengthened formulations leads to significantly reduced computation times. Another plausible example of an activity constraint is a ramping restriction on the activity levels, which would state that once an activity is begun, it must be performed within certain levels over time. Our approach can still be used to yield formulations and valid inequalities in this case. However, as in the nondecreasing special case, it may be possible to make use of these additional restrictions on the activities over time to yield stronger formulations. This is an area for future work.

A major limitation of our model is that it assumes all data is deterministic. Considering that the problem we study is a strategic planning problem with a long planning horizon, it is unrealistic to assume the data for future periods is known with certainty. In particular, data such as the customer demands and costs of not meeting demand (e.g., prices of the product that would have been sold) would be uncertain at the time the decisions are being made. Perhaps more significantly, the improvements in technology which determine the start-time dependent variable costs are likely to be challenging to predict. Therefore, it would be valuable to study extensions to this model which consider this uncertainty. This will be a very challenging extension, due to the multi-period nature of the problem and the difficulty in even modeling technology improvements over time. Thus, this problem could serve as an excellent motivation for the development of new modeling approaches and optimization methods for solving large-scale problems under uncertainty.

CHAPTER III

NEW APPROACHES FOR OPTIMIZATION WITH PROBABILISTIC CONSTRAINTS

3.1 Introduction

Consider an optimization problem with a probabilistic or chance constraint

$$\min \left\{ cx : x \in X,\ P\left\{ G(x,\xi) \leq \mathbf{0} \right\} \geq 1 - \epsilon \right\} \tag{PCP}$$

where $X \subset \mathbf{R}^n$ represents a deterministic feasible region, $c \in \mathbf{R}^n$, ξ is a random vector taking values in $\mathbf{R}^d$, $G : \mathbf{R}^n \times \mathbf{R}^d \to \mathbf{R}^m$ is a given constraint mapping and ϵ is a risk parameter chosen by the decision maker, typically near zero, e.g., $\epsilon = 0.01$ or $\epsilon = 0.05$. Problem PCP is also referred to as a *probabilistic program*. In PCP we enforce a single probabilistic constraint over *all* rows in the constraints $G(x,\xi) \leq \mathbf{0}$, rather than requiring that each row independently be satisfied with high probability. Such a constraint is known as a *joint probabilistic constraint*, and is appropriate in a context in which it is important to have all constraints satisfied simultaneously and there may be dependence between random variables in different rows.

Problems with joint probabilistic constraints have been extensively studied; see [57] for background and an extensive list of references. Probabilistic constraints have been used in various applications including supply chain management [43], production planning [49], optimization of chemical processes [35, 36] and surface water quality management [67].

Unfortunately, probabilistic programs are still largely intractable except for a few very special cases. There are two primary reasons for this intractability. First, in general, for a given $x \in X$, the quantity $P\left\{ G(x,\xi) \leq \mathbf{0} \right\}$ is hard to compute, as it requires multi-dimensional integration, and hence just checking feasibility of a solution is difficult. Second, the feasible region defined by a probabilistic constraint generally is not convex. We propose to address the difficulty in checking feasibility by solving a *sample approximation* problem based on a finite number of Monte Carlo samples of ξ. In addition, we propose to solve the

resulting non-convex sample approximation problem by mixed-integer programming.

In Section 3.3 we study how a sample approximation problem can be used to generate feasible solutions and optimality bounds for general probabilistic programs. We show that a sample approximation problem with risk level larger than the nominal risk level ϵ will yield a lower bound to the true optimal value with probability approaching one exponentially fast. This leads to an a priori estimate on the sample size required to have high confidence that the sample approximation will yield a lower bound. We also discuss alternative means of generating lower bounds, which can be used regardless of the sample size used. We then provide conditions under which solving a sample approximation problem with a risk level smaller than ϵ will yield feasible solutions to the original problem with high probability. Once again, we obtain a priori estimates on the sample size required to obtain high confidence that the sample approximation problem will yield a feasible solution to the original problem.

A key difference between our sample approximation scheme and existing approximations (e.g. [17, 51, 52]) is that we allow the risk level in the sample approximation problem to be positive, that is, we do not require that all sampled constraint sets be satisfied. Instead, the constraint sets which will be satisfied can be chosen optimally. The disadvantage of this scheme is that the sample approximation problem with positive risk level has a non-convex feasible region, and hence can be difficult to solve despite having a simplified probabilistic structure. Thus, in Section 3.4, we study how the sample approximation problem can be solved by mixed-integer programming. In particular, we study the special case in which only the right-hand side ξ is random, leading to strong formulations which can be used to solve instances with many rows and very many scenarios. Our approach in developing these formulations is to consider the relaxation obtained from a single row of the probabilistic constraint. This yields a system similar to the *mixing set* introduced by Günlük and Pochet [32], subject to an additional knapsack inequality. We derive strong valid inequalities for this system by first using the knapsack inequality to "pre-process" the mixing set and then applying the mixing inequalities of [32]; see also [3, 31]. We also derive an extended formulation, equivalent to one given by Miller and Wolsey in [48]. Making further use

of the knapsack inequality, we derive more general classes of valid inequalities for both the original and extended formulations. If all scenarios are equally likely, the knapsack inequality reduces to a cardinality restriction. In this case, we characterize the convex hull of feasible solutions to the extended formulation for the single row case. Although these results are motivated by the application to PCP, they can be used in any problem in which a mixing set appears along with a knapsack constraint.

The remainder of this chapter is organized as follows. We begin in Section 3.2 by discussing how our work relates to previous work. In Section 3.3 we present and analyze the sample approximation scheme. In Section 3.4 we study how the sample approximation problem can be solved in the special case in which only the right-hand side is random. We present results of a preliminary computational study of the use of the sample approximation scheme in Section 3.5. In Section 3.6 we present computational results using the strengthened formulations to solve the sample approximation problem. We close with concluding remarks and directions for future research in Section 3.7.

3.2 Relation to Previous Work

We begin by discussing results which are related to the sample approximation scheme we propose in Section 3.3. For solving the sample approximation problem we consider the case in which only the right-hand side is random, and thus we discuss methods which have been used to solve this special case of PCP. Finally, we discuss prior work that has used integer programming to solve problems with probabilistic constraints.

3.2.1 Approximations of Probabilistic Constraints

The idea of using sample approximation problems in stochastic programming has a long history, and this approach is often referred to as *Sample Average Approximation* (SAA). Usually, this approach is applied to solve problems of the form

$$\min_{x \in X} \mathbb{E}\big[F(x, \xi)\big] \tag{61}$$

in which the expected value of a function depending on the random vector ξ is to be optimized, with the decision variables varying in a deterministic feasible region. In the

SAA scheme, random samples $\xi^1, \ldots, \xi^N$ are taken of ξ and the objective in (61) is replaced by the sample average objective, leading to the approximation problem

$$\min_{x \in X} \frac{1}{N} \sum_{i=1}^{N} F(x, \xi^i). \tag{62}$$

In [20] it is shown that under fairly mild conditions for any $\beta > 0$, the probability that the distance between the optimal value (and solution) of (62) and the true optimal value (and solution) of (61) exceeds β converges to zero exponentially fast in N. In [65] the authors consider the SAA approach when X is a convex set, $F(x, \xi)$ is convex in x for all ξ and either (61) has a unique minimum or ξ has finite distribution and (61) corresponds to a two-stage stochastic linear program. Under these conditions, they show the optimal solution to (62) is an optimal solution of (61) with probability approaching one exponentially fast in N. In [41] similar results are derived for the case in which the feasible region is finite. In addition, [41] presents a statistical method for generating optimality bounds by solving multiple SAA problems. In the context of two-stage stochastic programs, these results are extended in [2] to the case in which the second stage may have integer variables and the first stage feasible region is not necessarily finite. A significant computational study of the SAA method in these contexts has been conducted in [45].

Results for the SAA approach applied to problems with random quantities in the constraints are less common. In [64], a class of problems is studied which includes problems with expected value constraints, i.e.

$$\min_{x \in X} \Big\{ \mathbb{E}\big[F(x, \xi)\big] : \mathbb{E}\big[H(x, \xi)\big] \geq b \Big\}. \tag{63}$$

A probabilistic constraint $P\big\{G(x, \xi) \leq 0\big\} \geq 1 - \epsilon$ can be cast as an expected value constraint by letting $H(x, \xi) = \mathbb{I}\big(G(x, \xi) \leq 0\big)$, where $\mathbb{I}(\cdot)$ is the indicator function taking value one when $\cdot$ is true and zero otherwise. Thus, $\mathbb{E}[H(x, \xi)] = P\big\{G(x, \xi) \leq 0\big\}$. Unfortunately, in [64], it is assumed that H is continuous in x, and hence these results are not applicable for probabilistic constraints. This is also true for the recent results presented in [8], where convergence to a (possibly local optimal) solution which satisfies second order optimality conditions is studied in the case where H may be non-convex. In [4] a model with expected value constraints in which the function H is not necessarily continuous is considered, and

hence their analysis does apply to the case of probabilistic constraints. However, they consider only the case in which the feasible region is finite, and they only discuss the theoretical rate of convergence. In contrast, we begin with a similar analysis for the finite feasible region case, but then extend the analysis to a number of significantly more general settings. In addition, we separate the analysis of when the sample approximation will be likely to yield a lower bound, and when it will be likely to yield feasible solutions. This separate analysis allows for the development of methods which yield optimality statements which hold with high probability.

Vogel [70] considers convergence properties of the sample approximation we use for probabilistic programs. When only the right-hand side is random with continuous distribution, it is shown that the probability that the distance between the sample feasible region and true feasible region is larger than any positive threshold decreases exponentially fast with the size of the sample. However, the convergence rate has poor dependence on the dimension of the random vector, implying that the number of samples required to yield a reasonable approximation would have to grow exponentially in this dimension. Better convergence is demonstrated for the case of random right-hand side with discrete distribution. For the general case, linear convergence is demonstrated in the case of continuous distributions. Our analysis of the sample approximation problem extends these results by improving on the convergence rates and by analyzing what happens when we allow the sample approximation problem to have different risk level than the nominal risk level ϵ. This allows the sample approximation problem to be used to generate feasible solutions and optimality bounds.

Recently, a number of approaches have been proposed to find approximate solutions to probabilistic programs; the common theme among these is that they all seek "safe" or conservative approximations which can be solved efficiently. That is, they propose approximation problems which are convex and yield solutions which are feasible, or at least highly likely to be feasible, to the original probabilistic program. Approaches of this type include: the scenario approximation method introduced by Calafiore and Campi [16, 17] and significantly extended by Nemirovski and Shapiro [51]; the Bernstein approximation scheme of Nemirovski and Shapiro [52]; and robust optimization e.g., [11, 15, 27]. The scenario

approximation methods are most similar to the sample approach we propose in that they solve an approximation problem based on independent Monte Carlo samples of the random vector. For example, if the problem has the form

$$\min_{x \in X} \{ f(x) : P\{G(x, \xi) \leq 0\} \geq 1 - \epsilon \} \tag{64}$$

the scenario approximation of [16, 17] takes samples $\xi^1, \ldots, \xi^N$ and solves the approximate problem

$$\min_{x \in X} \{ f(x) : G(x, \xi^i) \leq 0 \quad i = 1, \ldots, N \}. \tag{65}$$

That is, the scenario approximation enforces *all* of the constraints corresponding to the samples taken. When $X \subseteq \mathbf{R}^n$ is a convex set, f is convex and G is convex in x for each ξ, they show that the scenario approximation problem will yield a feasible solution to (64) with probability at least $1 - \delta$ for

$$N \geq \frac{2}{\epsilon} \log \left(\frac{1}{\delta} \right) + 2n + \frac{2n}{\epsilon} \log \left(\frac{2}{\epsilon} \right). \tag{66}$$

In addition, under the stated convexity assumptions, the scenario approximation problem remains a convex program. An advantage of this approach relative to the approximations [11, 15, 27, 52] is that the only assumption that is made on ξ is that samples can be taken from it. In [51], it is shown that by taking random samples from a modified distribution, the number of scenarios, N, required to yield a feasible solution to (64) grows *logarithmically* with $1/\epsilon$, as compared to the linear growth given in (66).

The conservative approximations, when applicable, are attractive because they allow efficient generation of feasible solutions. In particular, they can yield feasible solutions when the probabilistic constraint is "hard," that is, with ϵ very small, such as $\epsilon = 10^{-6}$ or even $\epsilon = 10^{-12}$. However, in a context in which ϵ is not so small, such as $\epsilon = 0.05$ or $\epsilon = 0.01$, the probabilistic constraint is more likely to represent a "soft" constraint, one which the decision-maker would like to have satisfied, but is willing to allow a nontrivial chance that it will be violated if doing so would significantly decrease the cost of the implemented solution. In this latter context, it would be desirable to obtain solutions which are feasible to the probabilistic constraint *along with* an assurance that the solutions are not too much

more costly than the lowest cost solution attaining the same risk level. In this way, the decision-maker can be confident they are choosing from solutions on the efficient frontier between the competing objectives of cost and risk. Unfortunately, with one exception, the recently proposed conservative approximations say very little in terms of how conservative the solutions are. In particular, it is not possible to make a statement about how much worse the objective is relative to the optimal value at a fixed risk level ϵ. The exception to this is [52], in which a statistical scheme for finding a lower bound based on solving multiple scenario approximations of the form (65) is presented, although the effectiveness of this bounding approach is not thoroughly investigated.

We propose a sample approximation scheme which is based on Monte Carlo samples of the random vector ξ, but in contrast to the scenario approximation approach, we do not require all sets of constraints to be satisfied. Instead, we solve an approximation problem which is itself an optimization problem with probabilistic constraints, but in this case the probability distribution is taken to be the empirical distribution obtained from the sample. The advantage of this approach is that it allows the generation of optimality guarantees with high probability, and allows these guarantees to be improved with increasing sample size. In addition, we expect it to yield solutions which are less conservative than those given by the conservative approximation schemes. The major disadvantage of our sample approximation is that even in the nice case when X is convex, f is convex, and $G(x, \xi)$ is convex in x the sample approximation problem will not be convex. Additionally, our sample approximation scheme will *only* be applicable when ϵ is not too small, since for very small ϵ the number of samples required to obtain a reasonable approximation would be intractably large.

Finally, we note that in the context of generating feasible solutions for (64), our sample approximation scheme includes as a special case the scenario approximation of [16, 17] in which the constraints corresponding all samples are enforced. In this special case, we obtain results very similar to those in [17] in terms of how many samples should be used to yield a solution feasible to (64) with high probability. However, our analysis is quite different from the analysis of [17], and in particular, requires a significantly different set of assumptions.

In some cases our assumptions are more stringent, but there are also a number of cases in which our assumptions apply and those of [17] do not, most notably if the feasible region X is not convex, as in the case of a mixed-integer program.

3.2.2 Probabilistic Constraints with Random Right-hand Side

When we turn to solving the sample approximation problem, we consider the special case in which randomness appears only in the right-hand side of the probabilistic constraints. Despite its restrictiveness, this special case has received considerable attention in the literature, see, e.g., [18, 21, 57]. In the case of continuous random variables, the primary focus has been on special cases in which the feasible region turns out to be convex. In [56, 57], Prékopa demonstrates that if the probability measure associated with the random right-hand side is *logarithmically concave*, then the associated probabilistic constraint defines a convex set. Furthermore, it is demonstrated that many continuous distributions indeed satisfy this property. Because of this, methods to solve general convex programming problems can be adapted to solve the probabilistic programming problem in this case. However, the work required to obtain reasonable estimates of the function values and gradient evaluations required for these methods grows very quickly with the dimension of the random vector, since these estimates require integration in multiple dimensions. As a result, these methods are limited to problems with relatively small dimension of the random right-hand side.

When the random right-hand side vector is discrete, the majority of past work has attempted to make use of the concept of p-efficient points. The set of p-efficient points basically corresponds to the set of non-dominated solutions of the probabilistically constrained program, and has been shown to be finite when the random right-hand side is discrete (but not necessarily finite), although the size is generally exponential in the dimension of the vector. A number of methods have been developed that attempt to solve such problems by exploiting this structure, see for example [13, 21, 57]. These methods have proven successful for solving problems with discrete distributions in which the dimension of the random vector is not too large. Unfortunately, as the dimension increases, the problem of generating these p-efficient points can become intractable. If the random vector consists of independent and

bounded component random variables, a binary integer program can be solved to generate p-efficient points as needed, although it increases in size with the dimension of the random vector and the number of possible values each component random variable can take on. As noted in [21], generation of p-efficient points in the case of dependent random variables will require new specialized algorithms.

In [14], the *probabilistic set covering problem* (PSC) is introduced. PSC is a set covering problem in which the right-hand side is random and a probabilistic constraint is enforced. In [14], the authors propose to solve PSC by enumeration, explicit or implicit, of the p-efficient points. More recently, in [63], the authors propose a formulation which also uses the ideas of p-efficient points, but exploits the possible existence of independent sub-vectors of the right-hand side. Because the sub-vectors have much smaller dimension, the number of p-efficient points for each sub-vector is manageable. This formulation is possible because the random right-hand side vector can only take values zero or one. In [62], Saxena proposes a delayed constraint generation scheme for this formulation of PSC which can possibly be used when the size of the sub-vectors gets large.

Another approach for solving probabilistic programs when only the right-hand side is random has been given in [18], in which a branch and bound algorithm is proposed. This algorithm is based on monotonicity arguments and can in theory be applied if the right-hand side random vector is discrete or continuous. However, for general distributions the inherent difficulty of checking feasibility persists, limiting the applicability to cases in which the dimension of right-hand side vector is small.

3.2.3 MIP Approach to Probabilistic Constraints

The idea to use mixed-integer programming (MIP) to solve linear programs with probabilistic constraints when the distribution of ξ is finite has been explored by Ruszczyński in [61]. Using the concept of *dominance* between the different possible outcomes of ξ, the author proposes to strengthen the MIP by adding precedence constraints and combining these with the knapsack inequality appearing in the formulation to develop valid inequalities, based on the idea of an *induced cover* and lifting. These techniques allow the solution of moderate

size instances in this general setting which allows randomness in the constraint matrix. Unfortunately, as the number of random components increases, the occurrence of dominance between scenarios will likely decrease, so that these techniques may not be sufficient to solve larger instances. In our analysis of solving the sample approximation problem, we begin with the same MIP formulation presented in [61], but we take advantage of the assumption that only the right-hand side is random to obtain strong formulations for that special case.

3.3 Sample Approximations

We now study how Monte Carlo sampling can be used to generate probabilistically constrained problems with finite distribution which can be used to approximate problems with general distributions. We begin by considering the general problem

$$z_\epsilon^* = \min\{f(x) : x \in X_\epsilon\} \tag{P_ϵ}$$

where

$$X_\epsilon = \Big\{x \in X : P\{G(x,\xi) \leq \mathbf{0}\} \geq 1-\epsilon\Big\}.$$

Here $X \subseteq \mathbf{R}^n$ represents the deterministic feasible region, $f(x)$ is the real valued objective function, and $G : \mathbf{R}^n \times \mathbf{R}^d \to \mathbf{R}^m$ is a constraint mapping depending on x and a random vector $\xi : \Omega \to \mathbf{R}^d$ defined on a probability space (Ω, Σ, P). We let Ξ denote the *support* of ξ, that is, Ξ is the smallest closed set such that $P\{\xi \in \Xi\} = 1$. We assume z_ϵ^* exists and is finite. For example, if X is compact and $G(x,\xi)$ is affine in x for each $\xi \in \Xi$, then X_ϵ is closed [34] and hence compact, and so if $f(x)$ is continuous then an optimal solution exists whenever $X_\epsilon \neq \emptyset$. Furthermore, we take as an assumption the measurability of any event S taken under probability, such as the event $\{G(x,\xi) \leq \mathbf{0}\}$ for each $x \in X$.

If X is a polyhedron, $f(x) = cx$, $G(x,\xi) = \xi - Tx$ $(d = m)$ then we obtain the probabilistically constrained linear program with random right-hand side

$$\min\Big\{cx : x \in X, P\{Tx \geq \xi\} \geq 1-\epsilon\Big\}. \tag{PCLPR}$$

We can also model a two-stage problem in which we make a decision x and wish to guarantee that with probability at least $1-\epsilon$ there is a feasible recourse decision y satisfying $Wy \geq$

$H(x,\xi)$, where W is an m by l matrix, and $H : \mathbf{R}^n \times \mathbf{R}^d \to \mathbf{R}^m$. This is accomplished by letting $G : \mathbf{R}^n \times \mathbf{R}^d \to \mathbf{R}$ be defined by

$$G(x,\xi) = \min_{\mu,y}\{\mu : Wy + \mu\mathbf{e} \geq H(x,\xi), \mu \geq -1\}$$

where $\mathbf{e} \in \mathbf{R}^m$ is a vector of all ones. Indeed, $G(x,\xi) \leq \mathbf{0}$ if and only if there exists $y \in \mathbf{R}^l$ and $\mu \leq 0$ such that $Wy + \mu\mathbf{e} \geq H(x,\xi)$, which occurs if and only if there exists $y \in \mathbf{R}^l$ such that $Wy \geq H(x,\xi)$.

Due to the general difficulty in calculating $P\big\{G(x,\xi) \leq \mathbf{0}\big\}$ for a given $x \in X$, we seek to approximate P_ϵ by solving a sample approximation problem. We let $\xi^1, \ldots, \xi^N$ be independent Monte Carlo samples of the random vector ξ. Then, for fixed $\alpha \in [0,1)$ the sample approximation problem is defined to be

$$\hat{z}_\alpha^N = \min\big\{f(x) : x \in X_\alpha^N\big\} \tag{P_α^N}$$

where

$$X_\alpha^N = \Big\{x \in X : \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\big(G(x,\xi^i) \leq \mathbf{0}\big) \geq 1-\alpha\Big\}.$$

We adopt the convention that if $X_\alpha^N = \emptyset$ then $\hat{z}_\alpha^N = +\infty$, whereas if P_α^N is unbounded, we take $\hat{z}_\alpha^N = -\infty$. We assume that, except for these two cases, P_α^N has an optimal solution. This assumption is satisfied, for example, if X is compact, $f(x)$ is continuous and $G(x,\xi)$ is continuous in x for each $\xi \in \Xi$, since then X_α^N is the union of finitely many compact sets (in this case $\hat{z}_\alpha^N = -\infty$ is also not possible). If $\alpha = 0$, the sample approximation problem P_0^N corresponds to the *scenario approximation* of probabilistic constraints, studied in [17] and [51]. Our goal is to establish statistical relationships between problems P_ϵ and P_α^N for $\alpha \geq 0$. We first consider when P_α^N yields lower bounds for P_ϵ, then consider when P_α^N yields feasible solutions for P_ϵ.

3.3.1 Lower Bounds

We now establish a bound on the probability that P_α^N yields a lower bound for P_ϵ. Let

$$\rho(\alpha,\epsilon,N) = \sum_{i=0}^{\lfloor \alpha N \rfloor} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}.$$

$\rho(\alpha, \epsilon, N)$ represents the probability of having at most $\lfloor \alpha N \rfloor$ "successes" in N independent trials, in which the probability of a success in each trial is ϵ.

Lemma 20. *Assume* P_ϵ *has an optimal solution. Then,*

$$P\big\{\hat{z}_\alpha^N \le z_\epsilon^*\big\} \ge \rho(\alpha, \epsilon, N).$$

Proof. Let $x^* \in X_\epsilon$ be an optimal solution to P_ϵ. Then, $P\big\{G(x^*, \xi^i) \not\le \mathbf{0}\big\} \le \epsilon$ for each i. Hence, if we call the event $\{G(x^*, \xi^i) \not\le \mathbf{0}\}$ a success, then the probability of a success in trial i is $\bar{\phi}(x^*) := P\big\{G(x^*, \xi^i) \not\le \mathbf{0}\big\} \le \epsilon$. By definition of X_α^N, $x^* \in X_\alpha^N$ if and only if

$$\begin{aligned} \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\big(G(x^*, \xi^i) \le \mathbf{0}\big) \ge 1 - \alpha \quad &\Leftrightarrow \quad \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\big(G(x^*, \xi^i) \not\le \mathbf{0}\big) \le \alpha \\ &\Leftrightarrow \quad \sum_{i=1}^{N} \mathbb{I}\big(G(x^*, \xi^i) \not\le \mathbf{0}\big) \le \lfloor \alpha N \rfloor. \end{aligned}$$

Hence, $P\big\{x^* \in X_\alpha^N\big\}$ is the probability of having at most $\lfloor \alpha N \rfloor$ successes in N trials. Also, if $x^* \in X_\alpha^N$ then $\hat{z}_\alpha^N \le z_\epsilon^*$. Thus,

$$P\big\{\hat{z}_\alpha^N \le z_\epsilon^*\big\} \ge P\big\{x^* \in X_\alpha^N\big\} = \rho(\alpha, \bar{\phi}(x^*), N) \ge \rho(\alpha, \epsilon, N)$$

since $\rho(\alpha, \epsilon, N)$ is decreasing in ϵ. □

For example, if $\alpha = 0$ as in previously studied scenario approximation [17, 51], then we obtain $P\big\{\hat{z}_\alpha^N \le z_\epsilon^*\big\} \ge \rho(0, \epsilon, N) = (1 - \epsilon)^N$. For this choice of α, it becomes very unlikely that the sample approximation P_α^N will yield a lower bound as N gets large. For $\alpha > \epsilon$ we see different behavior: the sample approximation yields a lower bound with probability approaching one exponentially fast as N increases. The proof is based on Hoeffding's inequality.

Theorem 21 (Hoeffding's Inequality [37])**.** *Let* $Y_1, \ldots, Y_N$ *be independent random variables with* $P\big\{Y_i \in [a_i, b_i]\big\} = 1$ *where* $a_i \le b_i$ *for* $i = 1, \ldots, N$*. Then, if* $t > 0$

$$P\Big\{\sum_{i=1}^{N}(Y_i - \mathbb{E}[Y_i]) \ge tN\Big\} \le \exp\Big\{-\frac{2N^2t^2}{\sum_{i=1}^{N}(b_i - a_i)^2}\Big\}.$$

Theorem 22. *Let* $\alpha > \epsilon$ *and assume* P_ϵ *has an optimal solution. Then,*

$$P\big\{\hat{z}_\alpha^N \le z_\epsilon^*\big\} \ge 1 - \exp\big\{-2N(\alpha - \epsilon)^2\big\}.$$

Proof. Let x^* be an optimal solution to P_ϵ. As in the proof of Lemma 20, if $x^* \in X_\alpha^N$ then $\hat{z}_\alpha^N \leq z_\epsilon^*$. For $i = 1, \ldots, N$ let Y_i be a random variable taking value 1 if $G(x^*, \xi^i) \not\leq \mathbf{0}$ and 0 otherwise. Then, $P\{Y_i \in [0,1]\} = 1$ and $E[Y_i] \leq \epsilon$. Hence,

$$\begin{aligned} P\{\hat{z}_\alpha^N > z_\epsilon^*\} \leq P\{x^* \notin X_\alpha^N\} &= P\{\frac{1}{N}\sum_{i=1}^N Y_i > \alpha\} \\ &\leq P\{\frac{1}{N}\sum_{i=1}^N (Y_i - E[Y_i]) > \alpha - \epsilon\} \\ &\leq \exp\Big\{-\frac{2N^2(\alpha-\epsilon)^2}{N}\Big\} = \exp\{-2N(\alpha-\epsilon)^2\} \end{aligned}$$

where the first inequality follows since $E[Y_i] \leq \epsilon$ and the second inequality follows from Hoeffding's inequality. □

Theorem 22 indicates that by taking a risk parameter $\alpha > \epsilon$ in our sample approximation problem, we will obtain a lower bound to the true optimal value with probability approaching one exponentially fast as N increases. Stated another way, suppose we solve a sample approximation problem P_α^N with $\alpha = \epsilon$. Then for any $\gamma > 0$ such that $\gamma < \epsilon$, the optimal value of this problem, $\hat{z}_\epsilon^N$ will be a lower bound to the optimal value of $\mathrm{P}_{\epsilon-\gamma}$ with probability approaching one exponentially fast with N. If γ is small this states that the optimal solution to the sample problem will have cost no worse than any solution that is "slightly less risky" than the nominal risk level ϵ.

Theorem 22 immediately yields a method for generating lower bounds with specified confidence $1 - \delta$, where $\delta \in (0,1)$. If we select $\alpha > \epsilon$ and

$$N \geq \frac{1}{2(\alpha-\epsilon)^2}\log\Big(\frac{1}{\delta}\Big)$$

then Theorem 22 ensures that $\hat{z}_\alpha^N \leq z_\epsilon^*$ with probability at least $1 - \delta$. Indeed, with this choice of α and N, we have

$$P\{\hat{z}_\alpha^N > z_\epsilon^*\} \leq \exp\{-2N(\alpha-\epsilon)^2\} \leq \exp\Big\{-\log\Big(\frac{1}{\delta}\Big)\Big\} = \delta.$$

Because $1/\delta$ is taken under logarithm, we can obtain a lower bound with high confidence, i.e. with δ very small, without increasing the sample size N too large. On the other hand,

the required sample size grows quadratically with $1/(\alpha - \epsilon)$ and hence will be large for α very close to ϵ.

Lemma 20 can also be used to obtain lower bounds with specified confidence, using the bounding procedure proposed by Nemirovski and Shapiro [52]. They restrict $\alpha = 0$ in the sample approximation, but the technique can be applied in exactly the same way when $\alpha > 0$, and it is likely this can make the bounding technique significantly more powerful. The idea is as follows. Take M sets of N independent samples of ξ, given by $\xi^{i,j}$ for $j = 1, \ldots, M$ and $i = 1, \ldots, N$ and for each j solve the associated sample approximation problem

$$\hat{z}^N_{\alpha,j} = \min\{f(x) : x \in X^N_{\alpha,j}\}$$

where

$$X^N_{\alpha,j} = \Big\{x \in X : \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\big(G(x, \xi^{i,j}) \leq \mathbf{0}\big) \geq 1 - \alpha\Big\}.$$

We then rearrange the values $\{\hat{z}^N_{\alpha,j}\}_{j=1}^M$ to obtain the *order statistics* $\hat{z}^N_{\alpha,[j]}$ for $j = 1, \ldots, M$ satisfying $\hat{z}^N_{\alpha,[1]} \leq \cdots \leq \hat{z}^N_{\alpha,[M]}$. Then, a lower bound which is valid with specified confidence $1 - \delta$ can be obtained as follows.

Theorem 23. *Let $\delta \in (0,1)$, $\alpha \in [0,1)$, and N, L and M be positive integers such that $L \leq M$ and*

$$\sum_{i=0}^{L-1} \binom{M}{i} \rho(\alpha, \epsilon, N)^i \big(1 - \rho(\alpha, \epsilon, N)\big)^{M-i} \leq \delta. \tag{67}$$

Then,

$$P\{\hat{z}^N_{\alpha,[L]} \leq z^*_\epsilon\} \geq 1 - \delta.$$

Proof. We show $P\{\hat{z}^N_{\alpha,[L]} > z^*_\epsilon\} \leq \delta$. Note that $\hat{z}^N_{\alpha,[L]} > z^*_\epsilon$ if and only if less than L of the values $\hat{z}^N_{\alpha,j}$ satisfy $\hat{z}^N_{\alpha,j} \leq z^*_\epsilon$. Thus, calling the event $\{\hat{z}^N_{\alpha,j} \leq z^*_\epsilon\}$ a success, the event $\hat{z}^N_{\alpha,[L]} > z^*_\epsilon$ occurs if and only if there are less than L successes in M trials, in which the probability of a success is $\eta := P\{\hat{z}^N_{\alpha,j} \leq z^*_\epsilon\}$. The result then follows since $\eta \geq \rho(\alpha, \epsilon, N)$ by Lemma 20 and so

$$\sum_{i=0}^{L-1} \binom{M}{i} \eta^i (1-\eta)^{M-i} \leq \sum_{i=0}^{L-1} \binom{M}{i} \rho(\alpha, \epsilon, N)^i \big(1 - \rho(\alpha, \epsilon, N)\big)^{M-i} \leq \delta$$

by (67). □

An interesting special case of Theorem 23 is obtained by taking $L = 1$. In this case, we are taking as our lower bound the minimum of the optimal values obtained from solving the M sample approximation problems. To have confidence $1 - \delta$ that the lower bound is truly a lower bound, we should choose M such that

$$\big(1 - \rho(\alpha, \epsilon, N)\big)^M \leq \delta. \tag{68}$$

With the choice of $L = 1$, let us consider how large M should be with $\alpha = 0$ and with $\alpha = \epsilon$. With $\alpha = 0$, we obtain $\rho(0, \epsilon, N) = (1 - \epsilon)^N$. Hence, to have confidence $1 - \delta$ to obtain a lower bound, we should take

$$M \geq \log\left(\frac{1}{\delta}\right) / \log\left(\frac{1}{1 - (1 - \epsilon)^N}\right). \tag{69}$$

Using the inequality $\log(1 + x) \leq x$ for $x > 0$ we have

$$\log\left(\frac{1}{1 - (1 - \epsilon)^N}\right) = \log\left(1 + \frac{(1 - \epsilon)^N}{1 - (1 - \epsilon)^N}\right) \leq \frac{(1 - \epsilon)^N}{1 - (1 - \epsilon)^N}.$$

Hence, when $\alpha = 0$, we should take

$$M \geq \log\left(\frac{1}{\delta}\right) \frac{1 - (1 - \epsilon)^N}{(1 - \epsilon)^N}.$$

Thus, for fixed $\epsilon \in (0, 1)$, the required M grows exponentially in N. For example, using (69), if $\delta = 0.001$ and $\epsilon = 0.01$, then for $N = 250$ we need $M \geq 82$, for $N = 500$ we need $M \geq 1,048$, and for $N = 750$ we need $M \geq 12,967$. If $\delta = 0.001$ and $\epsilon = 0.05$, then for $N = 50$ we should take $M \geq 87$, for $N = 100$ we should take $M \geq 1,160$, and for $N = 150$ we must already have $M \geq 15,157$! Thus, to keep M reasonably small, we must keep N small, but this will weaken the lower bound obtained in each sample.

Now suppose we take $L = 1$ and $\alpha = \epsilon$. Then, for N "large enough" (e.g. $N\epsilon \geq 10$), we have $\rho(\epsilon, \epsilon, N) \approx 1/2$. Indeed, $\rho(\epsilon, \epsilon, N)$ is the probability that a binomial random variable with success probability ϵ and N trials is at most $\lfloor \epsilon N \rfloor$. With N large enough relative to ϵ, this probability can be approximated by the probability that a random variable with Normal distribution having mean ϵN does not exceed $\lfloor \epsilon N \rfloor$. Because the median of the

normal distribution equals the mean, we obtain $\rho(\epsilon, \epsilon, N) \gtrsim 1/2$. Thus, with $L = 1$ and $\alpha = \epsilon$, we should choose M such that $(1/2)^M \leq \delta$, or

$$M \geq \log_2 \left(\frac{1}{\delta} \right).$$

Note that this bound is *independent of* N *and* ϵ. For example, for $\delta = 0.001$, we should take $M \geq 10$. The independence of N has the advantage that we can take N to be as large as is computationally tractable, which will tend to make each of the optimal values $\hat{z}^N_{\epsilon,j}$ closer to the true optimal z^*_ϵ, and hence make the lower bound $\min_j\{\hat{z}^N_{\epsilon,j}\}$ tighter.

3.3.2 Feasible Solutions

We now consider conditions under which an optimal solution to P^N_α, if one exists, is feasible to P_ϵ. The idea is that if we take the risk parameter α in P^N_α to be smaller than ϵ, then for N large enough the feasible region of P^N_α will be a subset of the feasible region of P_ϵ, so that any optimal solution to P^N_α must be feasible to P_ϵ. Unlike the case for lower bounds, we will need to make additional assumptions to assure P^N_α yields a feasible solution with high probability.

We begin by assuming that the feasible region X is finite. Note, however, that $|X|$ may be exponentially large, for example X could be the feasible region of a bounded integer program. We then show how this assumption can be relaxed and replaced with some milder assumptions.

3.3.2.1 Finite X

Theorem 24. *Suppose X is finite and $\alpha \in [0, \epsilon)$. Then,*

$$P\big\{X^N_\alpha \subseteq X_\epsilon\big\} \geq 1 - |X \setminus X_\epsilon| \exp\big\{-2N(\epsilon - \alpha)^2\big\}.$$

Proof. Consider any $x \in X \setminus X_\epsilon$, i.e. $x \in X$ with $P\big\{G(x, \xi) \leq \mathbf{0}\big\} < 1 - \epsilon$. We want to estimate the probability that $x \in X^N_\alpha$. For $i = 1, \ldots, N$ define the random variable Y_i by $Y_i = 1$ if $G(x, \xi^i) \leq \mathbf{0}$ and $Y_i = 0$ otherwise. Then, $E[Y_i] = P\big\{G(x, \xi^i) \leq \mathbf{0}\big\} < 1 - \epsilon$ and $P\big\{Y_i \in [0, 1]\big\} = 1$. Observing that $x \in X^N_\alpha$ if and only if $(1/N)\sum_{i=1}^N Y_i \geq 1 - \alpha$ and

applying Hoeffding's inequality, we obtain

$$P\{x \in X_\alpha^N\} = P\{\frac{1}{N}\sum_{i=1}^N Y_i \geq 1-\alpha\} \leq P\{\sum_{i=1}^N (Y_i - E[Y_i]) \geq N(\epsilon-\alpha)\}$$
$$\leq \exp\{-2N(\epsilon-\alpha)^2\}.$$

Then,

$$P\{X_\alpha^N \subseteq X_\epsilon\} = P\{\exists x \in X_\alpha^N \text{ s.t. } P\{G(x,\xi) \leq \mathbf{0}\} < 1-\epsilon\}$$
$$\leq \sum_{x \in X \setminus X_\epsilon} P\{x \in X_\alpha^N\}$$
$$\leq |X \setminus X_\epsilon| \exp\{-2N(\epsilon-\alpha)^2\}.$$

□

For fixed $\alpha < \epsilon$ and $\delta \in (0,1)$, Theorem 24 shows that if we take

$$N \geq \frac{1}{2(\epsilon-\alpha)^2} \log\left(\frac{|X \setminus X_\epsilon|}{\delta}\right)$$

then if P_α^N is feasible, it will yield a feasible solution to P_ϵ with probability at least $1-\delta$. If $|X| \leq U^n$, we can take

$$N \geq \frac{1}{2(\epsilon-\alpha)^2} \log\left(\frac{1}{\delta}\right) + \frac{n}{2(\epsilon-\alpha)^2} \log(U). \tag{70}$$

Note that N grows linearly with the dimension n of the feasible region, and logarithmically with $1/\delta$, so that the confidence of generating a feasible solution can be made large without requiring N to be too large. However, the quadratic dependence on $\epsilon - \alpha$ implies that this a priori estimate of how large N should be will grow quite large for α near ϵ. If we take $\alpha = 0$, we obtain improved dependence of N on ϵ.

Theorem 25. *Suppose X is finite and $\alpha = 0$. Then,*

$$P\{X_0^N \subseteq X_\epsilon\} \geq 1 - |X \setminus X_\epsilon|(1-\epsilon)^N.$$

Proof. With $\alpha = 0$, if $x \in X$ satisfies $P\{G(x,\xi) \leq \mathbf{0}\} < 1-\epsilon$, then $x \in X_0^N$ if and only if $G(x,\xi^i) \leq \mathbf{0}$ for each $i = 1,\ldots,N$, and hence $P\{x \in X_0^N\} < (1-\epsilon)^N$. The claim then follows just as in the proof of Theorem 24. □

When $\alpha = 0$, to obtain confidence $1 - \delta$ that P_α^N will yield a feasible solution to P_ϵ whenever P_α^N is feasible, we should take

$$N \geq \log^{-1}\left(\frac{1}{1-\epsilon}\right) \log\left(\frac{|X \setminus X_\epsilon|}{\delta}\right).$$

If $|X| \leq U^n$, then it is sufficient to take

$$N \geq \frac{1}{\epsilon} \log\left(\frac{1}{\delta}\right) + \frac{n}{\epsilon} \log U \tag{71}$$

where we have used the inequality $\log(1/(1-\epsilon)) \geq \epsilon$. Hence, with $\alpha = 0$, the required sample size again grows linearly in n, but now also linearly with $1/\epsilon$. Note the similarity between the bound (71) and the bound of Campi and Calafiore [17],

$$N \geq \frac{2}{\epsilon} \log\left(\frac{1}{\delta}\right) + 2n + \frac{2n}{\epsilon} \log\left(\frac{2}{\epsilon}\right)$$

which also exhibits linear dependence in n and (nearly) linear dependence in $1/\epsilon$. This is interesting considering the significantly different assumptions used for the analysis. In [17] it is assumed that X is a convex set and $G(x, \xi)$ is a convex function of x for every possibly value of ξ. In contrast, we make the strong assumption that X is finite, but require no other assumptions on the form of the random constraint $G(x, \xi) \leq \mathbf{0}$.

3.3.2.2 Random right-hand side

We now show how the assumption that X is finite can be relaxed when the probabilistic constraint involves randomness only in the right-hand side. Thus, in this section we assume $G(x, \xi) = \xi - g(x)$ where $g : \mathbf{R}^n \to \mathbf{R}^m$, and $\Xi \subseteq \mathbf{R}^m$. Let the cumulative distribution function of ξ be $F(y) = P\{\xi \leq y\}$ for $y \in \mathbf{R}^m$. Then, the feasible region of the probabilistically constrained problem with random right-hand side is

$$\bar{X}_\epsilon = \Big\{x \in X : F(g(x)) \geq 1 - \epsilon\Big\}.$$

The feasible region of the sample approximation problem for $\alpha \in [0, 1)$ is

$$\bar{X}_\alpha^N = \Big\{x \in X : \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\big(g(x) \geq \xi^i\big) \geq 1 - \alpha\Big\}.$$

We first consider the case that ξ has a finite distribution, that is, $\Xi = \{\xi^1, \ldots, \xi^K\}$. Note that K may be very large, for example $K = U^m$ for a positive integer U. Next, for $j = 1, \ldots, m$ define $\Xi_j = \{\xi_j^k : k = 1, \ldots, K\}$ and finally let $C = \prod_{j=1}^m \Xi_j$.

Theorem 26. *Suppose Ξ has a finite distribution and let $\alpha \in [0, \epsilon)$. Then,*

$$P\big\{\bar{X}_\alpha^N \subseteq \bar{X}_\epsilon\big\} \geq 1 - |C| \exp\big\{-2N(\epsilon - \alpha)^2\big\}.$$

Proof. Let $C_\epsilon = \{y \in C : F(y) \geq 1 - \epsilon\}$ and

$$C_\alpha^N = \Big\{y \in C : \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\big(y \geq \xi^i\big) \geq 1 - \alpha\Big\}.$$

Because C is a finite set, we can apply Theorem 24 to obtain

$$P\big\{C_\alpha^N \subseteq C_\epsilon\big\} \geq 1 - |C| \exp\big\{-2N(\epsilon - \alpha)^2\big\}. \tag{72}$$

Now, let $x \in \bar{X}_\alpha^N$, so that $x \in X$ and $\sum_{i=1}^N \mathbb{I}\big(g(x) \geq \xi^i\big) \geq N(1-\alpha)$. Define $\bar{y} \in C$ by

$$\bar{y}_j = \max\{y_j \in \Xi_j : y_j \leq g_j(x)\} \qquad j = 1, \ldots, m$$

so that by definition, $\bar{y} \leq g(x)$. Next, note that if $g(x) \geq \xi^i$ for some i, then also $\bar{y} \geq \xi^i$ since $\xi^i \in C$. Hence, $\sum_{i=1}^N \mathbb{I}\big(\bar{y} \geq \xi^i\big) \geq N(1-\alpha)$ and so $\bar{y} \in C_\alpha^N$. Hence, when $C_\alpha^N \subseteq C_\epsilon$, $F(\bar{y}) \geq 1 - \epsilon$ and because $\bar{y} \leq g(x)$, also $F(g(x)) \geq 1 - \epsilon$ and so $x \in \bar{X}_\epsilon$. Since $x \in \bar{X}_\alpha^N$ was arbitrary, this shows that when $C_\alpha^N \subseteq C_\epsilon$, $\bar{X}_\alpha^N \subseteq \bar{X}_\epsilon$ and the result follows from (72). □

If, for example, $|\Xi_j| \leq U$ for each j, then $|C| \leq U^m$ so to obtain confidence $1 - \delta$ that $\bar{X}_\alpha^N \subseteq \bar{X}_\epsilon$ it is sufficient to take

$$N \geq \frac{1}{2(\epsilon - \alpha)^2} \log\left(\frac{1}{\delta}\right) + \frac{m}{2(\epsilon - \alpha)^2} \log U. \tag{73}$$

The difference between this bound and (70) is that (73) depends linearly on m, the dimension of ξ, whereas (70) depends linearly on n, the dimension of x.

As in the case of Theorem 25, if we take $\alpha = 0$, we can obtain the stronger convergence result

$$P\big\{\bar{X}_0^N \subseteq \bar{X}_\epsilon\big\} \geq 1 - |C|(1 - \epsilon)^N.$$

The assumption in Theorem 26 that Ξ is finite can be relaxed if we assume $\bar{X}_\epsilon \subseteq \bar{X}(l, u) := \{x \in X : l \leq g(x) \leq u\}$ for some $l, u \in \mathbf{R}^m$. This assumption is not very strict. Indeed, if we define $l \in \mathbf{R}^m$ by

$$l_j = \min\{l \in \mathbf{R} : F_j(l) \geq 1 - \epsilon\}$$

where F_j is the marginal distribution of ξ_j for $j = 1, \ldots, m$ then $g(x) \geq l$ for any $x \in \bar{X}_\epsilon$. This holds because if $g_j(x) < l_j$ for some j, then $P\{g(x) \geq \xi\} \leq P\{g_j(x) \geq \xi_j\} = F_j(g_j(x)) < 1 - \epsilon$ by definition of l_j and hence $x \notin \bar{X}_\epsilon$. Furthermore, if X is compact and $g(x)$ is continuous in x, then if we define $u \in \mathbf{R}^m$ by

$$u_j = \max\{g_j(x) : x \in X\} \qquad j = 1, \ldots, m$$

then each u_j is finite, and by definition, $g(x) \leq u$ for any $x \in \bar{X}$. Under the assumption that $\bar{X}_\epsilon \subseteq \bar{X}(l, u)$ the assumption that Ξ is finite can be replaced by the assumption that $\Xi \cap \{y \in \mathbf{R}^m : l \leq y \leq u\}$ is finite, leading to a result similar to Theorem 26, with a nearly identical proof.

Alternatively, when $\bar{X}_\epsilon \subseteq \bar{X}(l, u)$, we can obtain a similar result if ξ has a Lipschitz continuous cumulative distribution function F on $[l, u] = \{y \in \mathbf{R}^m : l \leq y \leq u\}$. That is, we assume there exists $L > 0$ such that

$$|F(y) - F(y')| \leq L\|y - y'\|_\infty \qquad \forall y, y' \in [l, u]$$

where $\|y\|_\infty = \max\{|y_j| : j = 1, \ldots, m\}$. Under the assumption that $\bar{X}_\epsilon \subseteq \bar{X}(l, u)$ we add the constraints $l \leq g(x) \leq u$ to the sample approximation problem to obtain

$$\bar{X}_\alpha^N(l, u) = \Big\{x \in \bar{X}(l, u) : \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\big(g(x) \geq \xi^i\big) \geq 1 - \alpha\Big\}.$$

We define $D = \max\{u_j - l_j : j = 1, \ldots, m\}$. Then we have

Theorem 27. *Suppose $\bar{X}_\epsilon \subseteq \bar{X}(l, u)$ and F is Lipschitz continuous with constant L. Let $\alpha \in [0, \epsilon)$ and $\beta \in (0, \epsilon - \alpha)$. Then,*

$$P\big\{\bar{X}_\alpha^N(l, u) \subseteq \bar{X}_\epsilon\big\} \geq 1 - \lceil DL/\beta \rceil^m \exp\big\{-2N(\epsilon - \alpha - \beta)^2\big\}.$$

Proof. Let $K = \lceil DL/\beta \rceil$ and define $Y_j = \{l_j + (u_j - l_j)i/K : i = 1, \ldots, K\}$ for $j = 1, \ldots, m$ and

$$Y = \prod_{j=1}^{m} Y_j,$$

so that $|Y| = K^m$ and that for any $y \in [l, u]$ there exists $y' \in Y$ such that $y' \geq y$ and $\|y - y'\|_\infty \leq \beta/L$. Indeed, for a given $y \in [l, u]$ such a y' can be obtained by letting

$$y'_j = \min\{w \in Y_j : w \geq y_j\} \qquad j = 1, \ldots, m.$$

With this definition of y', we have $y' \geq y$ and

$$|y'_j - y_j| = y'_j - y_j \leq (u_j - l_j)/K \leq D/K \leq \beta/L \qquad j = 1, \ldots, m.$$

Next, let $Y_{\epsilon-\beta} = \{y \in Y : F(y) \geq 1 - \epsilon + \beta\}$ and

$$Y_\alpha^N = \Big\{y \in Y : \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\big(y \geq \xi^i\big) \geq 1 - \alpha\Big\}. \tag{74}$$

Since Y is finite and $\alpha < \epsilon - \beta$, we can apply Theorem 24 to obtain

$$P\big\{Y_\alpha^N \subseteq Y_{\epsilon-\beta}\big\} \geq 1 - |Y| \exp\big\{-2N(\epsilon - \alpha - \beta)^2\big\}.$$

Now, let $x \in \bar{X}_\alpha^N(l, u)$ and let $y' \in Y$ be such that $y' \geq g(x)$ and $\|y' - g(x)\|_\infty \leq \beta/L$. By Lipschitz continuity of F, this implies

$$F(y') - F(g(x)) \leq L\|y' - g(x)\|_\infty \leq \beta. \tag{75}$$

Because x satisfies $\sum_{i=1}^N \mathbb{I}\big(g(x) \geq \xi^i\big) \geq N(1-\alpha)$ and $y' \geq g(x)$, we have $\sum_{i=1}^N \mathbb{I}\big(y' \geq \xi^i\big) \geq N(1-\alpha)$ and hence $y' \in Y_\alpha^N$. Thus, using (75), when $Y_\alpha^N \subseteq Y_{\epsilon-\beta}$ occurs,

$$F(g(x)) \geq F(y') - \beta \geq (1 - \epsilon + \beta) - \beta = 1 - \epsilon.$$

Since $x \in \bar{X}_\alpha^N(l, u)$ was arbitrary, $Y_\alpha^N \subseteq Y_{\epsilon-\beta}$ implies $\bar{X}_\alpha^N(l, u) \subseteq \bar{X}_\epsilon$ and the result follows from (74). □

To obtain confidence at least $1 - \delta$ that $\bar{X}_\alpha^N(l, u) \subseteq \bar{X}_\epsilon$ it is sufficient to take

$$N \geq \frac{1}{2(\epsilon - \alpha - \beta)^2} \log\left(\frac{1}{\delta}\right) + \frac{m}{2(\epsilon - \alpha - \beta)^2} \log\Big\lceil \frac{DL}{\beta} \Big\rceil.$$

Note that for fixed $\epsilon > 0$ and $\alpha \in [0, \epsilon)$, β is a free parameter which can be chosen in $(0, \epsilon - \alpha)$. If, for example, we take $\beta = (\epsilon - \alpha)/2$ we obtain

$$N \geq \frac{2}{(\epsilon - \alpha)^2} \log\left(\frac{1}{\delta}\right) + \frac{2m}{(\epsilon - \alpha)^2} \log\Big\lceil \frac{2DL}{\epsilon - \alpha} \Big\rceil.$$

Once again, if $\alpha = 0$, similar arguments can be used to conclude that if

$$N \geq \frac{2}{\epsilon} \log\left(\frac{1}{\delta}\right) + \frac{2m}{\epsilon} \log\Big\lceil \frac{2DL}{\epsilon} \Big\rceil$$

then $P\big\{\bar{X}_0^N(l, u) \subseteq \bar{X}_\epsilon\big\} \geq 1 - \delta$.

3.3.2.3 Lipschitz continuous G

We now turn to the problem of using a sample approximation problem to generate feasible solutions to P_ϵ when X is not necessarily finite, and $G(x,\xi)$ does not necessarily have the form $G(x,\xi) = g(x) - \xi$. In this section, we assume for simplicity of exposition that G takes values in $\mathbf{R}$. This is without loss of generality, since if $\bar{G} : \mathbf{R}^n \times \mathbf{R}^d \to \mathbf{R}^m$ we can define $G : \mathbf{R}^n \times \mathbf{R}^d \to \mathbf{R}$ by $G(x,\xi) = \max\{\bar{G}_j(x,\xi) : j = 1,\ldots,m\}$ and the constraints $G(x,\xi) \leq 0$ and $\bar{G}(x,\xi) \leq \mathbf{0}$ are equivalent. In this section, we shall make the following Lipschitz continuity assumption on G.

Assumption 1. There exists $L > 0$ such that

$$|G(x,\xi) - G(x',\xi)| \leq L\|x - x'\|_\infty \qquad \forall x, x' \in X \text{ and } \forall \xi \in \Xi.$$

It is important that the Lipschitz constant L is independent of $\xi \in \Xi$, and this condition makes Assumption 1 rather stringent. There are, however, interesting cases in which the assumption does hold. For example, if Ξ is finite (with possibly huge cardinality) and $G(x,\xi)$ is Lipschitz continuous with Lipschitz constant $L(\xi)$ for each $\xi \in \Xi$, then Assumption 1 holds with $L = \max\{L(\xi) : \xi \in \Xi\}$. Alternatively, if Ξ is compact and $G(x,\xi) = \max\{T_j(\xi)x : j = 1,\ldots,m\}$ and $T_j : \Xi \to \mathbf{R}^n$ is continuous in ξ for each j, then Assumption 1 holds with

$$L = \sup_{\xi\in\Xi}\big\{\max\{\|T_j(\xi)\|_\infty : j = 1,\ldots,m\}\big\}.$$

To generate feasible solutions for this general case, we will also need to modify the sample approximation problem somewhat. In addition to taking a risk level α less than the nominal risk level ϵ, we will require that at least $(1-\alpha)N$ of the constraints be satisfied *strictly*. That is, for a fixed $\gamma > 0$, we define the sample approximation feasible region to be

$$X^N_{\alpha,\gamma} = \Big\{x \in X : \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\big(G(x,\xi) + \gamma \leq 0\big) \geq 1 - \alpha\Big\}.$$

Finally, we will assume that X is bounded, and let $D = \sup\{\|x - x'\|_\infty : x, x' \in X\}$ be the diameter of X.

Theorem 28. *Suppose X is bounded with diameter D and Assumption 1 holds. Let $\alpha \in$*

$[0, \epsilon)$, $\beta \in (0, \epsilon - \alpha)$ *and* $\gamma > 0$. *Then,*

$$P\big\{X^N_{\alpha,\gamma} \subseteq X_\epsilon\big\} \geq 1 - \lceil 1/\beta \rceil \lceil 2LD/\gamma \rceil^n \exp\big\{-2N(\epsilon - \alpha - \beta)^2\big\}.$$

Proof. For $x \in X$, let $\phi(x) = P\big\{G(x, \xi) \leq 0\big\}$. Let $J = \lceil 1/\beta \rceil$ and for $j = 1, \ldots, J-1$, define

$$X_j = \Big\{x \in X : \frac{j-1}{J} \leq \phi(x) < \frac{j}{J}\Big\}$$

and let $X_J = \{x \in X : (J-1)/J \leq \phi(x) \leq 1\}$. Next, we claim that for each j there exists a finite set $Z^\gamma_j \subseteq X_j$ such that $|Z^\gamma_j| \leq \lceil 2LD/\gamma \rceil^n$ and for all $x \in X_j$ there exists $z \in Z^\gamma_j$ such that $\|x - z\|_\infty \leq \gamma/L$. Indeed, because $X_j \subseteq X$ and X is bounded with diameter D, there exists a finite set $Y \subseteq \mathbf{R}^n$ with $|Y| \leq \lceil 2LD/\gamma \rceil^n$ such that for all $x \in X$ there exists $y \in Y$ such that $\|x - y\|_\infty \leq \gamma/2L$. For any $y \in \mathbf{R}^n$ and $\eta > 0$, define $B(y, \eta) = \{x \in \mathbf{R}^N : \|y - x\|_\infty \leq \eta\}$. Now, let $Y'_j = \{y \in Y : X_j \cap B(y, \gamma/2L) \neq \emptyset\}$, and for $y \in Y'_j$ select an arbitrary $x_y \in X_j \cap B(y, \gamma/2L)$. Then, let

$$Z^\gamma_j = \bigcup_{y \in Y'_j} x_y.$$

By definition, $Z^\gamma_j \subseteq X_j$ and $|Z^\gamma_j| \leq \lceil 2LD/\gamma \rceil^n$. In addition, for any $x \in X_j$, there exists y such that $x \in B(y, \gamma/2L)$ and because for this y, $X_j \cap B(y, \gamma/2L) \neq \emptyset$ there exists $x_y \in Z^\gamma_j$ such that $\|x_y - y\|_\infty \leq \gamma/2L$. Hence,

$$\|x_y - x\|_\infty \leq \|x_y - y\|_\infty + \|y - x\|_\infty \leq \gamma/L.$$

Now, define

$$Z^\gamma = \bigcup_{j=1}^{J} Z^\gamma_j$$

and observe that $|Z^\gamma| \leq J\lceil 2LD/\gamma \rceil^n$. Next, define $Z^\gamma_{\epsilon-\beta} = \big\{x \in Z^\gamma : P\big\{G(x, \xi) \leq 0\big\} \geq 1 - \epsilon + \beta\big\}$ and

$$Z^{\gamma,N}_\alpha = \Big\{x \in Z^\gamma : \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\big(G(x, \xi^i) \leq 0\big)\Big\}.$$

Since Z^γ is finite and $\alpha < \epsilon - \beta$ we can apply Theorem 24 to obtain

$$P\big\{Z^{\gamma,N}_\alpha \subseteq Z^\gamma_{\epsilon-\beta}\big\} \geq 1 - \lceil 1/\beta \rceil \lceil 2LD/\gamma \rceil^n \exp\big\{-2N(\epsilon - \alpha - \beta)^2\big\}. \tag{76}$$

Now consider an arbitrary $x \in X^N_{\alpha,\gamma}$. Let $j \in \{1, \ldots, J\}$ be such that $x \in X_j$. By definition of Z^γ_j there exists $z \in Z^\gamma_j$ such that $\|x - z\|_\infty \leq \gamma/L$. By definition of X_j and because $Z^\gamma_j \subseteq X_j$, we have $|\phi(x) - \phi(z)| \leq \beta$. In addition, Assumption 1 implies $|G(x, \xi^i) - G(z, \xi^i)| \leq \gamma$. Hence, if $G(x, \xi^i) + \gamma \leq 0$ then $G(z, \xi^i) \leq 0$ and, because x satisfies $\sum_{i=1}^N \mathbb{I}\big(G(x, \xi^i) + \gamma \leq 0\big) \geq N(1 - \alpha)$, it follows that z satisfies $\sum_{i=1}^N \mathbb{I}\big(G(z, \xi^i) \leq 0\big) \geq N(1 - \alpha)$. Thus $z \in Z^{\gamma,N}_\alpha$ and so if $Z^{\gamma,N}_\alpha \subseteq Z^\gamma_{\epsilon-\beta}$ then $\phi(z) \geq 1 - \epsilon + \beta$. Thus, $\phi(x) \geq \phi(z) - \beta \geq 1 - \epsilon$ when $Z^{\gamma,N}_\alpha \subseteq Z^\gamma_{\epsilon-\beta}$. Since $x \in X^N_{\alpha,\gamma}$ was arbitrary, $Z^{\gamma,N}_\alpha \subseteq Z^\gamma_{\epsilon-\beta}$ implies $X^N_{\alpha,\gamma} \subseteq X_\epsilon$ and the result follows from (76). □

Once again, for fixed ϵ and $\alpha < \epsilon$, β is a free parameter to be chosen in $(0, \epsilon - \alpha)$. If we choose, for example $\beta = (\epsilon - \alpha)/2$, then we can assure $X^N_{\alpha,\gamma} \subseteq X_\epsilon$ with confidence at least $1 - \delta$ by taking

$$N \geq \frac{2}{(\epsilon - \alpha)^2}\Big[\log\left(\frac{1}{\delta}\right) + n \log\Big\lceil\frac{2LD}{\gamma}\Big\rceil + \log\Big\lceil\frac{2}{\epsilon - \alpha}\Big\rceil\Big].$$

Additionally, if $\alpha = 0$, similar arguments show that $X^N_{0,\gamma} \subseteq X_\epsilon$ occurs with probability at least $1 - \delta$ if

$$N \geq \frac{2}{\epsilon}\Big[\log\left(\frac{1}{\delta}\right) + n \log\Big\lceil\frac{2LD}{\gamma}\Big\rceil + \log\Big\lceil\frac{2}{\epsilon}\Big\rceil\Big].$$

Regardless of whether $\alpha = 0$ or $\alpha > 0$ the term $1/\gamma$ is taken under log, and hence γ can be made very small without significantly increasing the required sample size, suggesting that modifying the sample approximation problem to require at least $(1 - \alpha)N$ of the sampled constraints to be satisfied with slack at least γ need not significantly alter the feasible region.

3.3.2.4 A Posteriori Feasibility Checking

The results of Sections 3.3.2.1 - 3.3.2.3 demonstrate that with appropriately constructed sample approximation problems, the probability that the resulting feasible region will be a subset of the true feasible region X_ϵ approaches one exponentially fast. This gives strong theoretical support for using these sample approximations to yield solutions feasible to X_ϵ. These results yield *a priori* estimates on how large the sample size N should be to have high confidence the sample approximation feasible region will be a subset of X_ϵ. However, these a priori estimates are likely to yield required sample sizes which are very large, and hence

the sample approximation problems will still be impractical to solve. This is particularly true if $\alpha > 0$ and $\epsilon - \alpha$ is small. However, typically in sampling approximation results such as these, the a priori estimates of the required sample size are very conservative, and in fact much smaller sample sizes are sufficient. See [45] for a computational demonstration of this phenomenon for the case of Sample Average Approximation applied to two-stage stochastic linear programs. Thus, a natural alternative to using the sample size suggested by the a priori estimates is to solve a sample approximation problem with a smaller sample to yield a candidate solution $\hat{x} \in X$, and then conduct an *a posteriori* check to see whether $P\{G(\hat{x}, \xi) \leq \mathbf{0}\} \geq 1 - \epsilon$. A simple method for conducting an a posteriori analysis of the risk of a candidate solution is to take a single very large Monte Carlo sample $\xi^1, \ldots, \xi^{N'}$ and count how many times $G(\hat{x}, \xi^i) \leq \mathbf{0}$ holds. Bounds on the true risk $P\{G(\hat{x}, \xi) \leq \mathbf{0}\}$ which hold with high confidence can then be constructed, and if N' is very large, these bounds should be tight. This approach will not work well if the allowed risk ϵ is extremely small, but on the other hand, we do not expect the sample approximation approach to be practical in this case anyway. Of course, if good estimates of $P\{G(\hat{x}, \xi) \leq \mathbf{0}\}$ can be obtained efficiently by some other method, for example if $G(x, \xi) = \xi - g(x)$ and ξ consists of independent components, then this other method should be used for a posteriori feasibility checking.

3.4 Solving the Sample Approximation

We now turn to solving the sample approximation problem in the case of a probabilistic programming problem with random right-hand side given by

$$\begin{array}{ll} \min & cx \\ \text{s.t.} & P\{g(x) \geq \xi\} \geq 1 - \epsilon \\ & x \in X. \end{array} \tag{PCPR}$$

Here $X \subseteq \mathbf{R}^n$ represents a deterministic feasible region, ξ is a random vector taking values in $\mathbf{R}^m$, $g : \mathbf{R}^n \to \mathbf{R}^m$, $\epsilon \in (0, 1)$ (typically small) and $c \in \mathbf{R}^n$. An interesting special case of PCPR is the probabilistically constrained linear program with random right-hand side (PCLPR), in which X is a polyhedron and $g(x) = Tx$ for an $m \times n$ matrix T. We

assume that ξ has finite support, that is there exist vectors, $\xi^i \in \mathbf{R}^m, i = 1, \ldots, N$ such that $P\{\xi = \xi^i\} = \pi_i$ for each i where $\pi_i > 0$ and $\sum_{i=1}^N \pi_i = 1$. Note that in the sample approximation problem, we would have $\pi_i = 1/N$ for all i. We begin by considering general probabilities here because much of the analysis applies to this case. We refer to the possible outcomes as scenarios. We assume without loss of generality that $\xi^i \geq 0$ and $\pi_i \leq \epsilon$ for each i. We also define the set $\mathcal{N} = \{1, \ldots, N\}$.

Before proceeding, we note that PCPR is NP-hard even when the constraint $x \in X$ is not present, all outcomes are equally likely and g is very simple.

Theorem 29. *PCPR is NP-hard, even in the special case in which $\pi_i = 1/N$ for all $i \in \mathcal{N}$, the constraints $x \in X$ are not present, $g(x) = x$, and $c = (1, \ldots, 1) \in \mathbf{R}^m$.*

Proof. Let $K = \lceil (1 - \epsilon)N \rceil$. Then, under the stated conditions PCPR can be written as

$$\min_{I \subseteq \mathcal{N}} \left\{ \sum_{j=1}^{m} \max_{i \in I} \{\xi_j^i\} : |I| \geq K \right\}.$$

We show that the associated decision problem:

> (DPCLP) Given non-negative integers ξ_j^i for $i = 1, \ldots, N,\ j = 1, \ldots, m$, $K \leq N$ and B, is there an $I \subseteq \mathcal{N}$ such that $|I| \geq K$ and $\sum_{j=1}^m \max_{i \in I} \left\{\xi_j^i\right\} \leq B$?

is NP-complete by reduction from the NP-complete problem CLIQUE. Consider an instance of CLIQUE given by graph $G = (V, E)$, in which we wish to decide whether there exists a clique of size C. We construct an instance of DPCLP by letting $\{1, \ldots, m\} = V$, $\mathcal{N} = E$, $B = C$, $K = C(C-1)/2$ and $\xi_j^i = 1$ if edge i is incident to node j and $\xi_j^i = 0$ otherwise. The key observation is that for any $I \subseteq E$, and $j \in V$,

$$\max_{i \in I} \{\xi_j^i\} = \begin{cases} 1 & \text{if some edge } i \in I \text{ is incident to node } j \\ 0 & \text{otherwise.} \end{cases}$$

Hence, if there exists a clique of size C in G then we have a subgraph of G consisting of C nodes and $C(C-1)/2$ edges. Thus there exists $I \subseteq \mathcal{N}$ with $|I| = C(C-1)/2 = K$ and

$$\sum_{j=1}^{m} \max_{i \in I} \{\xi_j^i\} = C = B$$

and the answer to DPCLP is yes.

Conversely, if the answer to DPCLP is yes, there exists $I \subseteq E$ of size at least $K = C(C-1)/2$ such that the number of nodes incident to I is at most $B = C$. This can only happen if I defines a clique of size C. □

3.4.1 The MIP Formulation

We now formulate PCPR as a mixed-integer program [61]. To do so, we introduce for each $i \in \mathcal{N}$, a binary variable z_i, where $z_i = 0$ guarantees that $g(x) \geq \xi^i$. Observe that because $\epsilon < 1$ we must have $g(x) \geq \xi^i$ for at least one $i \in \mathcal{N}$, and because $\xi^i \geq 0$ for all i, this implies $g(x) \geq 0$ in every feasible solution of PCPR. Then, letting $v = g(x)$, we obtain the MIP formulation of PCPR

$$
\begin{aligned}
(PMIP) \quad \min \quad & cx \\
\text{s.t. } & x \in X,\ g(x) - v = 0 && (77)\\
& v + \xi^i z_i \geq \xi^i \qquad i = 1, \ldots, N && (78)\\
& \sum_{i=1}^{N} \pi_i z_i \leq \epsilon && (79)\\
& x \geq 0, \quad z \in \{0,1\}^N
\end{aligned}
$$

where (79) is equivalent to the probabilistic constraint

$$\sum_{i=1}^{N} \pi_i (1 - z_i) \geq 1 - \epsilon.$$

If X represents the feasible region of a mixed-integer linear program and $g(x) = Tx$ for a $m \times n$ matrix T, then PMIP is a mixed-integer *linear* program, otherwise, PMIP is a mixed-integer nonlinear program.

3.4.2 Strengthening the Formulation

We begin by considering how the formulation PMIP can be strengthened when the probabilities π_i are general. In Section 3.4.2.2 we present results specialized to the case when all π_i are equal.

3.4.2.1 General Probabilities

Our approach is to strengthen PMIP by ignoring (77) and finding strong formulations for the set

$$F := \left\{(v,z) \in \mathbf{R}_+^m \times \{0,1\}^N : (78), (79)\right\}. \tag{80}$$

Note that

$$F = \bigcap_{j=1}^{m} \left\{(v,z) : (v_j, z) \in G_j\right\},$$

where for $j = 1, \ldots, m$

$$G_j = \left\{(v_j,z) \in \mathbf{R}_+ \times \{0,1\}^N : (79), \quad v_j + \xi_j^i z_i \geq \xi_j^i \quad i = 1, \ldots, N\right\}.$$

Thus, a natural first step in developing a strong formulation for F is to develop a strong formulation for each G_j. In particular, note that if an inequality is facet-defining for $\text{conv}(G_j)$, then it is also facet-defining for $\text{conv}(F)$. This follows because if an inequality valid for G_j is supported by $N+1$ affinely independent points in $\mathbf{R}^{N+1}$, then because this inequality will not have coefficients on v_i for any $i \neq j$, the set of supporting points can trivially be extended to a set of $N+m$ affinely independent supporting points in $\mathbf{R}^{N+m}$ by appropriately setting the v_i values for each $i \neq j$.

The above discussion leads us to consider the generic set

$$G = \left\{(y,z) \in \mathbf{R}_+ \times \{0,1\}^N : (79), \quad y + h_i z_i \geq h_i \quad i = 1, \ldots, N\right\} \tag{81}$$

obtained by dropping the index j and setting $y = v_j$ and $h_i = \xi_j^i$ for each i. We assume without loss of generality that $h_1 \geq h_2 \geq \cdots \geq h_N$. The relaxation of G obtained by dropping (79) is a *mixing set* given by

$$P = \left\{(y,z) \in \mathbf{R}_+ \times \{0,1\}^N : y + h_i z_i \geq h_i \quad i = 1, \ldots, N\right\}.$$

This set has been extensively studied, in varying degrees of generality, by Atamtürk et. al [3], Günlük and Pochet [32], Guan et. al [31] and Miller and Wolsey [48]. The *star inequalities* of [3] given by

$$y + \sum_{j=1}^{l} (h_{t_j} - h_{t_{j+1}}) z_{t_j} \geq h_{t_1} \quad \forall T = \{t_1, \ldots, t_l\} \subseteq \mathcal{N}, \tag{82}$$

where $t_1 < t_2 < \cdots < t_l$ and $h_{t_{l+1}} := 0$ are valid for P. Furthermore, these inequalities can be separated in polynomial time, are facet-defining for P when $t_1 = 1$, and are sufficient to define the convex hull of P [3, 31, 32].

We can tighten these inequalities for G by using the knapsack constraint (79). In particular, let $p := \max\{k : \sum_{i=1}^{k} \pi_i \leq \epsilon\}$. Then, from the knapsack constraint, we cannot have $z_i = 1$ for all $i = 1, \ldots, p+1$ and thus we have $y \geq h_{p+1}$. This also implies that the mixed-integer constraints in G are redundant for $i = p+1, \ldots, N$. Thus, we can replace the inequalities $y + h_i z_i \geq h_i$ for $i = 1, \ldots, N$ in the definition of G by the inequalities

$$y + (h_i - h_{p+1}) z_i \geq h_i \quad i = 1, \ldots, p. \tag{83}$$

That is, we have

$$G = \left\{ (y, z) \in \mathbf{R}_+ \times \{0,1\}^N : (79), (83) \right\}. \tag{84}$$

In addition to yielding a tighter relaxation, the description (84) of G is also more compact. In typical applications, ϵ is near 0, suggesting $p << N$. When applied for each j in the set F, if p is the same for all rows, this would yield a formulation with $mp << mN$ rows.

By applying the star inequalities to (84) we obtain

Theorem 30. *The inequalities*

$$y + \sum_{j=1}^{l} (h_{t_j} - h_{t_{j+1}}) z_{t_j} \geq h_{t_1} \quad \forall T = \{t_1, \ldots, t_l\} \subseteq \{1, \ldots, p\} \tag{85}$$

with $t_1 < \ldots < t_l$ and $h_{t_{l+1}} := h_{p+1}$, are valid for G. Moreover, (85) *is facet-defining for* $\mathrm{conv}(G)$ *if and only if $h_{t_1} = h_1$.*

Proof. The result follows directly from Proposition 3.4 and Theorem 3.5 of [3] after appropriate reformulation. See also [31, 32]. However, since our formulation differs somewhat, we give a self-contained proof. To prove (85) is valid, let $(y, z) \in G$ and let $j^* = \min\left\{ j \in \{1, \ldots, l\} : z_{t_j} = 0 \right\}$. Then $y \geq h_{t_{j^*}}$. Thus,

$$y + \sum_{j=1}^{l} (h_{t_j} - h_{t_{j+1}}) z_{t_j} \geq h_{t_{j*}} + \sum_{j=1}^{j^*-1} (h_{t_j} - h_{t_{j+1}}) = h_{t_1}.$$

If $h_{t_1} < h_1$, then a stronger inequality can be obtained by including index 1 in the set T, proving that this is a necessary condition for (85) to be facet-defining. Consider the

following set of points: $(h_1, e_i), i \in N \setminus T$, $(h_i, \sum_{j=1}^{i-1} e_j), i \in T$ and $(h_{p+1}, \sum_{j=1}^{p} e_j)$, where e_j is the j^{th} unit vector in $\mathbf{R}^n$. It is straightforward to verify that these $n+1$ feasible points satisfy (85) at equality and are affinely independent, completing the proof. □

We refer to the inequalities (85) as the *strengthened star inequalities*. Because the strengthened star inequalities are just the star inequalities applied to a strengthened mixing set, separation can be accomplished using an algorithm for separation of star inequalities [3, 31, 32].

3.4.2.2 Equal Probabilities

We now consider the case in which $\pi_i = 1/N$ for all $i \in \mathcal{N}$. Thus $p = \max\{k : \sum_{i=1}^{k} 1/N \leq \epsilon\} = \lfloor N\epsilon \rfloor$ and the knapsack constraint (79) becomes

$$\sum_{i=1}^{N} z_i \leq N\epsilon$$

which, by integrality on z_i, can be strengthened to the simple cardinality restriction

$$\sum_{i=1}^{N} z_i \leq p. \tag{86}$$

Thus, the feasible region (84) becomes

$$G' = \left\{(y, z) \in \mathbf{R}_+ \times \{0,1\}^N : (83), (86)\right\}.$$

Although the strengthened star inequalities are not sufficient to characterize the convex hull of G', we now show that it is possible to separate over $\text{conv}(G')$ in polynomial time. To obtain this result we first show that for any $(\gamma, \alpha) \in \mathbf{R}^{N+1}$, the problem

$$\min\left\{\gamma y + \alpha z : (y, z) \in G'\right\} \tag{87}$$

is easy to solve. For $k = 1, \ldots, p$ let

$$\mathcal{S}_k = \{S \subseteq \{k, \ldots, N\} : |S| \leq p-k+1\}$$

and

$$S_k^* \in \operatorname*{arg\,min}_{S \in \mathcal{S}_k} \left\{\sum_{i \in S} \alpha_i\right\}.$$

Also, let $S_{p+1}^* = \emptyset$ and $k^* \in \arg\min\{\gamma h_k + \sum_{i \in S_k^*} \alpha_i : k = 1, \ldots, p+1\}$.

Lemma 31. *If $\gamma < 0$, then* (87) *is unbounded. Otherwise, an optimal solution to* (87) *is given by $y = h_{k^*}$ and $z_i = 1$ for $i \in S^*_{k^*} \cup \{1, \ldots, k^* - 1\}$ and $z_i = 0$ otherwise.*

Proof. Problem (87) is unbounded when $\gamma < 0$ because $(1, \mathbf{0})$ is a feasible direction for G'. Now suppose $\gamma \geq 0$. We consider all feasible values of y, $y \geq h_{p+1}$. First, if $y \geq h_1$, then the z_i can be set to any values satisfying (86), and hence it would yield the minimum objective to set $z_i = 1$ if and only if $i \in S^*_1$ and to set $y = h_1$ since $\gamma \geq 0$. For any $k \in \{2, \ldots, p+1\}$, if $h_{k-1} > y \geq h_k$ then we must set $z_i = 1$ for $i = 1, \ldots, k-1$. The minimum objective in this case is then obtained by setting $y = h_k$ and $z_i = 1$ for $i = 1, \ldots, k-1$ and $i \in S^*_k$. The optimal solution to (87) is then obtained by considering y in each of these ranges. □

Using Lemma 31, we can optimize over G' by first sorting the values of α_i in increasing order, then finding the sets S^*_k by considering at most $p-k+1$ of the smallest values in this list for each $k = 1, \ldots, p+1$. Subsequently finding the index k^* yields an optimal solution defined by Lemma 31. This yields an obvious algorithm with complexity $O(N \log N + p^2) = O(N^2)$. It follows that we can separate over $\text{conv}(G')$ in polynomial time. We begin by characterizing the set of valid inequalities for G'.

Theorem 32. *Any valid inequality for G' with nonzero coefficient on y can be written in the form*

$$y \geq \beta + \sum_{i=1}^{n} \alpha_i z_i. \tag{88}$$

Furthermore, (88) *is valid for G' if and only if there exists (σ, ρ) such that*

$$\beta + \sum_{i=1}^{k-1} \alpha_i + (p-k+1)\sigma_k + \sum_{i=k}^{N} \rho_{ik} \leq h_k \quad k = 1, \ldots, p+1 \tag{89}$$

$$\alpha_i - \sigma_k - \rho_{ik} \leq 0 \qquad i = k, \ldots, N,\ k = 1, \ldots, p+1 \tag{90}$$

$$\sigma \geq 0, \rho \geq 0. \tag{91}$$

Proof. First consider a generic inequality of the form $\gamma y \geq \beta + \sum_{i=1}^{N} \alpha_i z_i$. Since $(1, \mathbf{0})$ is a feasible direction for G', we know this inequality is valid for G' only if $\gamma \geq 0$. Thus, if a valid inequality for G' has nonzero coefficient γ on y, then $\gamma > 0$, and so we can scale the inequality such that $\gamma = 1$, thus obtaining the form (88). Now, since any extreme point

of conv(G') is an optimal solution to (87) for some $(\gamma', \alpha') \in \mathbf{R}^{N+1}$, we know by Lemma 31 that the extreme points of conv(G') are contained in the set of feasible points given by $y = h_k$, $z_i = 1$ for $i = 1, \ldots, k-1$ and $i \in S$, and $z_i = 0$ otherwise, for all $S \in \mathcal{S}_k$ and $k = 1, \ldots, p+1$. This fact, combined with the fact that $(1, \mathbf{0})$ is the *only* feasible direction for G', implies inequality (88) is valid for G' if and only if

$$\beta + \sum_{i=1}^{k-1} \alpha_i + \max_{S \in \mathcal{S}_k} \sum_{i \in S} \alpha_i \leq h_k \quad k = 1, \ldots, p+1. \tag{92}$$

Note that

$$\begin{aligned} \max_{S \in \mathcal{S}_k} \sum_{i \in S} \alpha_i \quad = \quad & \max_{\omega} \quad \sum_{i=k}^{n} \omega_{ik} \alpha_i \\ & \text{s.t.} \ \sum_{i=k}^{N} \omega_{ik} \leq p-k+1 \qquad (93) \\ & \qquad 0 \leq \omega_{ik} \leq 1 \qquad i = k, \ldots, N \\ = \quad & \min_{\sigma, \rho} \quad (p-k+1)\sigma_k + \sum_{i=k}^{n} \rho_{ik} \\ & \text{s.t.} \quad \sigma_k + \rho_{ik} \geq \alpha_i \qquad i = k, \ldots, N \qquad (94) \\ & \qquad \sigma_k \geq 0, \rho_{ik} \geq 0 \qquad i = k, \ldots, N \end{aligned}$$

by linear programming duality since (93) is feasible and bounded and its optimal solution is integral. It follows that condition (92) is satisfied and hence (88) is valid for G' if and only if there exists (σ, ρ) such that the system (89) - (91) is satisfied. □

Using Theorem 32 we can separate over conv(G') by solving a polynomial size linear program.

Corollary 33. *Suppose* (y^*, z^*) *satisfy* $z^* \in Z := \left\{ z \in [0,1]^N : \sum_{i=1}^{N} z_i \leq p \right\}$. *Then,* $(y^*, z^*) \in \text{conv}(G')$ *if and only if*

$$y^* \geq LP^* = \max_{\alpha, \beta, \sigma, \rho} \left\{ \beta + \sum_{i=1}^{N} \alpha_i z_i^* : (89) - (91) \right\} \tag{95}$$

where LP^* *exists and is finite. Furthermore, if* $y^* < LP^*$ *and* (α^*, β^*) *is optimal to* (95), *then* $y \geq \beta^* + \sum_{i=1}^{N} \alpha^* z_i$ *is a valid inequality for* G' *which is violated by* (y^*, z^*).

Proof. By Theorem 32, if $y^* \geq LP^*$, then (y^*, z^*) satisfies all valid inequalities for G' which have nonzero coefficient on y. But, $z^* \in Z$ and integrality of the set Z imply (y^*, z^*) also satisfies all valid inequalities which have a zero coefficient on y, showing that $(y^*, z^*) \in \text{conv}(G')$. Conversely, if $y^* < LP^*$, then the optimal solution to (95) defines a valid inequality of the form (88) which is violated by (y^*, z^*).

We next argue that (95) has an optimal solution. First note that it is feasible since we can set $\beta = h_{p+1}$ and all other variables to zero. Next, because $z^* \in Z$, and Z is an integral polytope, we know there exists sets $S_j,\ j \in J$ for some finite index set J, and $\lambda \in \mathbf{R}_+^{|J|}$ such that $\sum_{j\in J} \lambda_j = 1$ and $z^* = \sum_{j\in J} \lambda_j z^j$ where $z_i^j = 1$ if $i \in S_j$ and 0 otherwise. Hence,

$$\beta + \sum_{i=1}^{N} \alpha_i z_i^* = \beta + \sum_{j\in J} \lambda_j \sum_{i\in S_j} \alpha_i = \sum_{j\in J} \lambda_j (\beta + \sum_{i\in S_j} \alpha_i) \leq \sum_{j\in J} \lambda_j h_1 = h_1$$

where the inequality follows from (92) for $k = 1$ which is satisfied whenever $(\alpha, \beta, \sigma, \rho)$ is feasible to (89) - (91). Thus, the objective is bounded, and so (95) has an optimal solution. □

Although (95) yields a theoretically efficient way to separate over $\text{conv}(G')$, it still may be too expensive to solve a linear program to generate cuts. We would therefore prefer to have an explicit characterization of a class or classes of valid inequalities for G' with an associated combinatorial algorithm for separation. The following theorem gives an example of one such class, which generalizes the strengthened star inequalities.

Theorem 34. *Let $m \in \{1, \ldots, p\}$, $T = \{t_1, \ldots, t_l\} \subseteq \{1, \ldots, m\}$ and $Q = \{q_1, \ldots, q_{p-m}\} \subseteq \{p+1, \ldots, N\}$. For $m < p$, define $\Delta_1^m = h_{m+1} - h_{m+2}$ and*

$$\Delta_i^m = \max \left\{ \Delta_{i-1}^m, h_{m+1} - h_{m+i+1} - \sum_{j=1}^{i-1} \Delta_j^m \right\} \quad i = 2, \ldots, p-m.$$

Then, with $h_{t_{l+1}} := h_{m+1}$,

$$y + \sum_{j=1}^{l} (h_{t_j} - h_{t_{j+1}}) z_{t_j} + \sum_{j=1}^{p-m} \Delta_j^m (1 - z_{q_j}) \geq h_{t_1} \tag{96}$$

is valid for G' and facet-defining for $\text{conv}(G')$ if and only if $h_{t_1} = h_1$.

Proof. First note that if $m = p$, we recover the strengthened star inequalities. Now, let $m < p$ and T, Q satisfy the conditions of the theorem and let $(y, z) \in G'$ and $S = \{i \in \mathcal{N} : z_i = 1\}$. Suppose first there exists $t_j \in T \setminus S$ and let $j^* = \min\{j \in \{1, \dots, l\} : t_j \notin S\}$. Then, $z_{t_{j^*}} = 0$ and so $y \geq h_{t_{j^*}}$. Hence,

$$\begin{aligned} y + \sum_{j=1}^{l} (h_{t_j} - h_{t_{j+1}}) z_{t_j} &\geq h_{t_{j^*}} + \sum_{j=1}^{j^*-1} (h_{t_j} - h_{t_{j+1}}) \\ &= h_{t_1} \geq h_{t_1} - \sum_{j=1}^{p-m} \Delta_j^m (1 - z_{q_j}) \end{aligned}$$

since $\Delta_j^m \geq 0$ for all j.

Next, suppose $T \subseteq S$. Now let $k = \sum_{i \in Q} (1 - z_i)$ so that, because $|Q| = p - m$, $0 \leq k \leq p - m$ and $\sum_{i \in Q} z_i = p - m - k$. Because $Q \subseteq \{p + 1, \dots, N\}$, we know $\sum_{i=1}^{p} z_i + \sum_{i \in Q} z_j \leq p$ and hence $\sum_{i=1}^{p} z_i \leq k + m$. It follows that $y \geq h_{k+m+1}$. Next, note that by definition, $\Delta_1^m \leq \Delta_2^m \leq \cdots \leq \Delta_{p-m}^m$. Thus

$$\begin{aligned} \sum_{j=1}^{p-m} \Delta_j^m (1 - z_{q_j}) \geq \sum_{j=1}^{k} \Delta_j^m &= \Delta_k^m + \sum_{j=1}^{k-1} \Delta_j^m \\ &\geq (h_{m+1} - h_{m+k+1} - \sum_{j=1}^{k-1} \Delta_j^m) + \sum_{j=1}^{k-1} \Delta_j^m \\ &= h_{m+1} - h_{m+k+1}. \end{aligned} \tag{97}$$

Using (97), $y \geq h_{k+m+1}$ and the fact that $T \subseteq S$ we have

$$\begin{aligned} y + \sum_{j=1}^{l} (h_{t_j} - h_{t_{j+1}}) z_{t_j} &\geq h_{k+m+1} + \sum_{j=1}^{l} (h_{t_j} - h_{t_{j+1}}) \\ &= h_{k+m+1} + h_{t_1} - h_{m+1} \geq h_{t_1} - \sum_{j=1}^{p-m} \Delta_j^m (1 - z_{q_j}) \end{aligned}$$

completing the proof of validity.

The proof that $h_{t_1} = h_1$ is a necessary condition for (96) to be facet-defining is exactly the same as that given for the star inequalities in the proof of Theorem 30.

Consider the following set of points in G':

$$(h_1, e_i + \sum_{j \in Q} e_j) \qquad i \in \mathcal{N} \setminus T \setminus Q \tag{98}$$

$$(h_i, \sum_{j=1}^{i-1} e_j + \sum_{j \in Q} e_j) \qquad i \in T \tag{99}$$

$$(h_{m+l(k)+1}, \sum_{j=1}^{l(k)+m} e_j + \sum_{j=l(k)}^{k-1} e_{q_j} + \sum_{j=k+1}^{p-m} e_{q_j}) \quad k = 1, \ldots, p-m \tag{100}$$

$$(h_{m+1}, \sum_{j=1}^{m} e_j + \sum_{j \in Q} e_j) \tag{101}$$

where

$$l(k) := \max \left\{ l = 1, \ldots, k : \Delta_k^m = h_{m+1} - h_{m+l+1} - \sum_{j=1}^{l-1} \Delta_j^m \right\}.$$

First note that $l(k)$ is well-defined since if $\Delta_k^m \neq h_{m+1} - h_{m+k+1} - \sum_{j=1}^{k-1} \Delta_j^m$ then $\Delta_k^m = \Delta_{k-1}^m$. Inductively, if $\Delta_k^m \neq h_{m+1} - h_{m+l+1} - \sum_{j=1}^{l-1} \Delta_j^m$ for some $l < k$, then $\Delta_k^m = \Delta_{l-1}^m$. Since, by definition $\Delta_1^m = h_{m+1} - h_{m+2}$ we know we must have $\Delta_k^m = h_{m+1} - h_{m+l+1} - \sum_{j=1}^{l-1} \Delta_j^m$ for some $l \leq k$. Verification that the points (98),(99) and (101) satisfy (96) at equality is straightforward. For point k in the set of points (100), the left-hand side of (96) evaluates to:

$$h_{m+l(k)+1} + (h_1 - h_{m+1}) + \sum_{j=1}^{l(k)-1} \Delta_j^m + \Delta_k^m = h_1$$

by definition of $l(k)$, and hence points (100) satisfy (96) at equality. Verification that the $N+1$ points given in (98)-(101) are affinely independent is straightforward, concluding the proof. □

Example 1. Let $N = 10$ and $\epsilon = 0.4$ so that $p = 4$ and suppose $h_{1-5} = \{20, 18, 14, 11, 6\}$.

The formulation of G' for this example is

$$\begin{aligned} y + 14z_1 &\geq 20 \\ y + 12z_2 &\geq 18 \\ y + 8z_3 &\geq 14 \\ y + 5z_4 &\geq 11 \\ \sum_{i=1}^{10} z_i \leq 4, \quad z_i \in \{0,1\} \quad i &= 1, \ldots, 10. \end{aligned}$$

Let $m = 2$, $T = \{1, 2\}$ and $Q = \{5, 6\}$. Then, $\Delta_1^2 = 3$ and $\Delta_2^2 = \max\{3, 8 - 3\} = 5$ so that (96) yields

$$y + 2z_1 + 4z_2 + 3(1 - z_5) + 5(1 - z_6) \geq 20. \qquad \square$$

Separation of inequalities (96) can be accomplished by a simple modification to the routine for separating the strengthened star inequalities.

3.4.3 A Strong Extended Formulation

3.4.3.1 General Probabilities

Let

$$FS = \left\{(y, z) \in \mathbf{R}_+ \times [0,1]^N : (79), (85)\right\}.$$

FS represents the polyhedral relaxation of G, augmented with the strengthened star inequalities. Note that the inequalities (83) are included in FS by taking $T = \{i\}$, so that enforcing integrality in FS would yield a valid formulation for the set G. Our aim is to develop a reasonably compact extended formulation which is equivalent to FS. To do so, we introduce variables $w_1, \ldots, w_p$ and let

$$EG = \left\{(y, z, w) \in \mathbf{R}_+ \times \{0,1\}^{N+p} : (102) - (105)\right\}$$

where

$$w_i - w_{i+1} \geq 0 \quad i = 1, \ldots, p \tag{102}$$

$$z_i - w_i \geq 0 \quad i = 1, \ldots, p \tag{103}$$

$$y + \sum_{i=1}^{p} (h_i - h_{i+1}) w_i \geq h_1 \tag{104}$$

$$\sum_{i=1}^{N} \pi_i z_i \leq \epsilon \tag{105}$$

and $w_{p+1} := 0$. The variables w_i can be interpreted as deciding whether or not scenario i is satisfied for the single row under consideration. The motivation for introducing these variables is that because they are specific to the single row under consideration, the ordering on the h_i values implies that the inequalities (102) can be safely added. Note that this is not the case for the original variables z_i for $i \in \mathcal{N}$ since they are common to all rows in the formulation. The inequalities (103) ensure that if a scenario is infeasible for the single row under consideration, then it is infeasible overall. Because of the inequalities (102), the inequalities (83) used in the description (84) of G can be replaced by the single inequality (104). We now show that EG is a valid formulation for G.

Theorem 35. $\text{Proj}_{(y,z)}(EG) = G$.

Proof. First, suppose $(y, z, w) \in EG$. Let $l \in \{1, \ldots, p+1\}$ be such that $w_i = 1,\ i = 1, \ldots, l-1$ and $w_i = 0,\ i = l, \ldots, p$. Then, $y \geq h_1 - (h_1 - h_l) = h_l$. For $i = 1, \ldots, l-1$ we have also $z_i = 1$ and hence,

$$y + (h_i - h_{p+1}) z_i \geq h_l + (h_i - h_{p+1}) \geq h_i$$

and for $i = l, \ldots, N$ we have $y + (h_i - h_{p+1}) z_i \geq h_l \geq h_i$ which establishes that $(y, z) \in G$. Now, let $(y, z) \in G$ and let $l = \min\{i : z_i = 0\}$. Then, $y + (h_l - h_{p+1}) z_l = y \geq h_l$. Let $w_i = 1,\ i = 1, \ldots, l-1$ and $w_i = 0,\ i = l, \ldots, p$. Then, $z_i \geq w_i$ for $i = 1, \ldots, p$, w_i are non-increasing, and $y \geq h_l = h_1 - \sum_{i=1}^{p} (h_i - h_{i+1}) w_i$ which establishes $(y, z, w) \in EG$. □

An interesting result is that the linear relaxation of this extended formulation is as strong as having all strengthened star inequalities in the original formulation. A similar

type of result has been proved in [48]. Let EF be the polyhedron obtained by relaxing integrality in EG.

Theorem 36. $\text{Proj}_{(y,z)}(EF) = FS$.

Proof. First suppose $(y, z) \in FS$. We show there exists $w \in \mathbf{R}^p_+$ such that $(y, z, w) \in EF$. For $i = 1, \ldots, p$ let $w_i = \min\{z_j : j = 1, \ldots, i\}$. By definition, $1 \geq w_1 \geq w_2 \geq \cdots w_p \geq 0$ and $z_i \geq w_i$ for $i = 1, \ldots, p$. Next, let $T := \{i = 1, \ldots, p : w_i = z_i\} = \{t_1, \ldots, t_l\}$, say. By construction, we have $w_i = w_{t_j}$ for $i = t_j, \ldots, t_{j+1} - 1$, $j = 1, \ldots, l$ ($t_{p+1} := p + 1$). Thus,

$$\sum_{i=1}^{p}(h_i - h_{i+1})w_i = \sum_{j=1}^{l}(h_{t_j} - h_{t_{j+1}})w_{t_j} = \sum_{j=1}^{l}(h_{t_j} - h_{t_{j+1}})z_{t_j}$$

implying that $y + \sum_{i=1}^{p}(h_i - h_{i+1})w_i \geq h_1$ as desired.

Now suppose $(y, z, w) \in EF$. Let $T = \{t_1, \ldots, t_l\} \subseteq \{1, \ldots, p\}$. Then,

$$\begin{aligned} y + \sum_{j=1}^{l}(h_{t_j} - h_{t_{j+1}})z_{t_j} &\geq y + \sum_{j=1}^{l}(h_{t_j} - h_{t_{j+1}})w_{t_j} \\ &\geq y + \sum_{j=1}^{l}\sum_{i=t_j}^{t_{j+1}-1}(h_i - h_{i+1})w_i \\ &= y + \sum_{i=t_1}^{p}(h_i - h_{i+1})w_i. \end{aligned}$$

But also, $y + \sum_{i=1}^{p}(h_i - h_{i+1})w_i \geq h_1$ and so

$$y + \sum_{i=t_1}^{p}(h_i - h_{i+1})w_i \geq h_1 - \sum_{i=1}^{t_1-1}(h_i - h_{i+1})w_i \geq h_1 - (h_1 - h_{t_1}) = h_{t_1}.$$

Thus, $(y, z) \in FS$. □

Because of the knapsack constraint (105), formulation EF does not characterize the convex hull of feasible solutions of G. We therefore investigate what other valid inequalities exist for this formulation. We introduce the notation

$$f_k := \sum_{i=1}^{k}\pi_i, \quad k = 0, \ldots, p.$$

Theorem 37. *Let $k \in \{1, \ldots, p\}$ and let $S \subseteq \{k, \ldots, N\}$ be such that $\sum_{i\in S}\pi_i \leq \epsilon - f_{k-1}$. Then,*

$$\sum_{i\in S}\pi_i z_i + \sum_{i\in\{k,\ldots,p\}\setminus S}\pi_i w_i \leq \epsilon - f_{k-1} \tag{106}$$

is valid for EG.

Proof. Let $l = \max\{i : w_i = 1\}$ so that $z_i = w_i = 1$ for $i = 1, \ldots, l$ and hence $\sum_{i=l+1}^{N} \pi_i z_i \leq \epsilon - f_l$. Suppose first $l < k$. Then, $\sum_{i \in \{k,\ldots,p\} \setminus S} \pi_i w_i = 0$ and the result follows since, by definition of the set S, $\sum_{i \in S} \pi_i \leq \epsilon - f_{k-1}$. Next, suppose $l \geq k$. Then,

$$\sum_{i \in S} \pi_i z_i \leq \sum_{i \in S \cap \{k,\ldots,l\}} \pi_i z_i + \sum_{i=l+1}^{N} \pi_i z_i \leq \sum_{i \in S \cap \{k,\ldots,l\}} \pi_i + \epsilon - f_l$$

and also $\sum_{i \in \{k,\ldots,p\} \setminus S} \pi_i w_i = \sum_{i \in \{k,\ldots,l\} \setminus S} \pi_i$. Thus,

$$\sum_{i \in S} \pi_i z_i + \sum_{i \in \{k,\ldots,p\} \setminus S} \pi_i w_i \leq \sum_{i \in S \cap \{k,\ldots,l\}} \pi_i + \epsilon - f_l + \sum_{i \in \{k,\ldots,l\} \setminus S} \pi_i = \epsilon - f_{k-1}.$$

□

3.4.3.2 Equal Probabilities

Now, consider the case in which $\pi_i = 1/N$ for $i = 1, \ldots, N$. Then the extended formulation becomes

$$EG' = \left\{ (y, z, w) \in \mathbf{R}_+ \times \{0,1\}^{N+p} : (86) \text{ and } (102) - (104) \right\}.$$

The inequalities (106) become

$$\sum_{i \in S} z_i + \sum_{i \in \{k,\ldots,p\} \setminus S} w_i \leq p - k + 1 \qquad \forall S \in \mathcal{S}_k,\ k = 1, \ldots, p. \tag{107}$$

Example 2. (Example 1 continued.) The extended formulation EG' is given by

$$w_1 \geq w_2 \geq w_3 \geq w_4$$

$$z_i \geq w_i \quad i = 1, \ldots, 4$$

$$y + 2w_1 + 4w_2 + 3w_3 + 5w_4 \geq 20$$

$$\sum_{i=1}^{10} z_i \leq 4,\ z \in \{0,1\}^{10},\ w \in \{0,1\}^4.$$

Let $k = 2$ and $S = \{4, 5, 6\}$. Then (107) becomes

$$z_4 + z_5 + z_6 + w_2 + w_3 \leq 3.$$

□

Next we show that in the equal probabilities case, the inequalities (107) together with the inequalities defining EG' are sufficient to define the convex hull of the extended formulation EG'. Let

$$EH' = \left\{ (y, z, w) \in \mathbf{R}_+ \times [0,1]^{N+p} : (86), (102) - (104) \text{ and } (107) \right\}$$

be the linear relaxation of the extended formulation, augmented with this set of valid inequalities.

Theorem 38. $EH' = \text{conv}(EG')$.

Proof. That $EH' \supseteq \text{conv}(EG')$ is immediate by validity of the extended formulation and the inequalities (107).

To prove $EH' \subseteq \text{conv}(EG')$ we first show that it is sufficient to prove that the polytope

$$H = \left\{(z, w) \in [0,1]^{N+p} : (86), (102), (103) \text{ and } (107)\right\}$$

is integral. Indeed, suppose H is integral, and let $(y, z, w) \in EH'$. Then, $(z, w) \in H$, and hence there a exists a finite set of integral points $(z^j, w^j), j \in J$, each in H, and a weight vector $\lambda \in \mathbf{R}_+^{|J|}$ with $\sum_{j\in J} \lambda_j = 1$ such that $(z, w) = \sum_{j\in J} \lambda_j (z^j, w^j)$. For each $j \in J$ define $y^j = h_1 - \sum_{i=1}^{p} (h_i - h_{i+1}) w_i^j$ so that $(y^j, z^j, w^j) \in EG'$ and also

$$\sum_{j\in J} \lambda_j y^j = h_1 - \sum_{i=1}^{p} (h_i - h_{i+1}) w_i \leq y.$$

Thus, there exists $\mu \geq 0$ such that $(y, z, w) = \sum_{j\in J} \lambda_j (y^j, z^j, w^j) + \mu(1, \mathbf{0})$ where each $(y^j, z^j, w^j) \in EG'$ and $(1, \mathbf{0})$ is a feasible direction for EG', which establishes that $(y, z, w) \in \text{conv}(EG')$.

We now move to proving the integrality of H, or equivalently, that $H = \text{conv}(H^I)$ where $H^I = H \cap \{0,1\}^{N+p}$. Thus, if $(z, w) \in H$, we aim to prove $(z, w) \in \text{conv}(H^I)$. We do this in two steps. First we establish a sufficient condition for $(z, w) \in \text{conv}(H^I)$, and then show that if $(z, w) \in H$ it satisfies this condition.

A sufficient condition for $(z, w) \in \text{conv}(H^I)$.

First observe that the feasible points of H^I are given by $w_j = 1$ for $j = 1, \ldots, k-1$ and $w_j = 0$ for $j = k, \ldots, p$ and

$$z_j = \begin{cases} 1 & j = 1, \ldots, k-1 \text{ and } j \in S \\ 0 & j \in \{k, \ldots, N\} \setminus S \end{cases}$$

for all $S \in \mathcal{S}_k$ and $k = 1, \ldots, p+1$. Thus, an inequality

$$\sum_{j=1}^{N} \alpha_j z_j + \sum_{j=1}^{p} \gamma_j w_j - \beta \leq 0 \tag{108}$$

is valid for $\text{conv}(H^I)$ if and only if

$$\sum_{j=1}^{k-1} (\alpha_j + \gamma_j) + \max_{S \in \mathcal{S}_k} \sum_{j \in S} \alpha_j - \beta \leq 0 \qquad k = 1, \ldots, p+1. \tag{109}$$

Representing the term $\max \left\{ \sum_{j \in S} \alpha_j : S \in \mathcal{S}_k \right\}$ as a linear program and taking the dual, as in (93) and (94) in the proof of Theorem 32, we obtain that (109) is satisfied and hence (108) is valid if and only if the system of inequalities

$$\sum_{j=1}^{k-1} (\alpha_j + \gamma_j) + \sum_{j=k}^{N} \rho_{jk} + (p-k+1)\sigma_k - \beta \leq 0 \tag{110}$$

$$\alpha_j - \sigma_k - \rho_{jk} \leq 0 \quad j = k, \ldots, N \tag{111}$$

$$\sigma_k \geq 0, \ \rho_{jk} \geq 0 \quad j = k, \ldots, N \tag{112}$$

has a feasible solution for $k = 1, \ldots, p+1$. Thus, $(w, z) \in \text{conv}(H^I)$ if and only if

$$\max_{\alpha, \beta, \gamma, \sigma, \rho} \left\{ \sum_{j=1}^{N} \alpha_j z_j + \sum_{j=1}^{p} \gamma_j w_j - \beta : (110) - (112), \ k = 1, \ldots, p+1 \right\} \leq 0.$$

For $k = 1, \ldots, p+1$, associate with (110) the dual variable δ_k, and with (111) the dual variables η_{jk} for $j = k, \ldots, N$. Then, applying Farkas' lemma to (110) - (112) and the condition

$$\sum_{j=1}^{N} \alpha_j z_j + \sum_{j=1}^{p} \gamma_j w_j - \beta > 0$$

we obtain that $(w, z) \in \text{conv}(H^I)$ if and only if the system

$$\sum_{k=j+1}^{p+1} \delta_k = w_j \quad j = 1, \ldots, p \tag{113}$$

$$\sum_{k=j+1}^{p+1} \delta_k + \sum_{k=1}^{j} \eta_{jk} = z_j \quad j = 1, \ldots, p \tag{114}$$

$$\sum_{k=1}^{p+1} \eta_{jk} = z_j \quad j = p+1, \ldots, N \tag{115}$$

$$(p-k+1)\delta_k - \sum_{j=k}^{n} \eta_{jk} \geq 0 \quad k = 1, \ldots, p+1 \tag{116}$$

$$\delta_k - \eta_{jk} \geq 0 \quad j = k, \ldots, N,\ k = 1, \ldots, p+1 \tag{117}$$

$$\sum_{k=1}^{p+1} \delta_k = 1 \tag{118}$$

$$\delta_k \geq 0, \eta_{jk} \geq 0 \quad j = k, \ldots, N,\ k = 1, \ldots, p+1 \tag{119}$$

has a feasible solution, where constraints (113) are associated with variables γ, (114) and (115) are associated with α, (116) are associated with σ, (117) are associated with ρ, and (118) is associated with β. Noting that (113) and (118) imply $\delta_k = w_{k-1} - w_k$ for $k = 1, \ldots, p+1$, with $w_0 := 1$ and $w_{p+1} := 0$, we see that $(w, z) \in \text{conv}(H^I)$ if and only if $w_{k-1} - w_k \geq 0$ for $k = 1, \ldots, p+1$ and the system

$$\sum_{k=1}^{\min\{j,p+1\}} \eta_{jk} = \theta_j \quad j = 1, \ldots, n \tag{120}$$

$$\sum_{j=k}^{n} \eta_{jk} \leq (p-k+1)(w_{k-1} - w_k) \quad k = 1, \ldots, p+1 \tag{121}$$

$$0 \leq \eta_{jk} \leq w_{k-1} - w_k \quad j = k, \ldots, N,\ k = 1, \ldots, p+1 \tag{122}$$

has a feasible solution, where $\theta_j = z_j - w_j$ for $j = 1, \ldots, p$ and $\theta_j = z_j$ for $j = p+1, \ldots, N$.

Verification that $(z, w) \in H$ implies $(z, w) \in \text{conv}(H^I)$.

Let $(z, w) \in H$ and consider a network G with node set given by $V = \{u, v, r_k \text{ for } k = 1, \ldots, p+1,\ m_j \text{ for } j \in \mathcal{N}\}$. This network has arcs from u to r_k with capacity $(p-k+1)(w_{k-1} - w_k)$ for all $k = 1, \ldots, p+1$, arcs from r_k to m_j with capacity $w_{k-1} - w_k$ for all $j = k, \ldots, N$ and $k = 1, \ldots, p+1$, and arcs from m_j to v with capacity θ_j

for all $j \in \mathcal{N}$. An example of this network with $N = 4$ and $p = 2$ is given in Figure 2. The labels on the arcs in this figure represent the capacities. For the arcs from nodes r_k to nodes m_j, the capacity depends only on the node r_k, so only the first outgoing arc from each r_k is labeled. It is easy to check that if this network has a flow from u to v of value $\sum_{j\in\mathcal{N}} \theta_j$, then the system (120) - (122) has a feasible solution. We will show that $(z, w) \in H$ implies the minimum $u - v$ cut in the network is at least $\sum_{j\in\mathcal{N}} \theta_j$, and by the max-flow min-cut theorem, this guarantees a flow of this value exists, proving that $(z, w) \in \text{conv}(H^I)$.

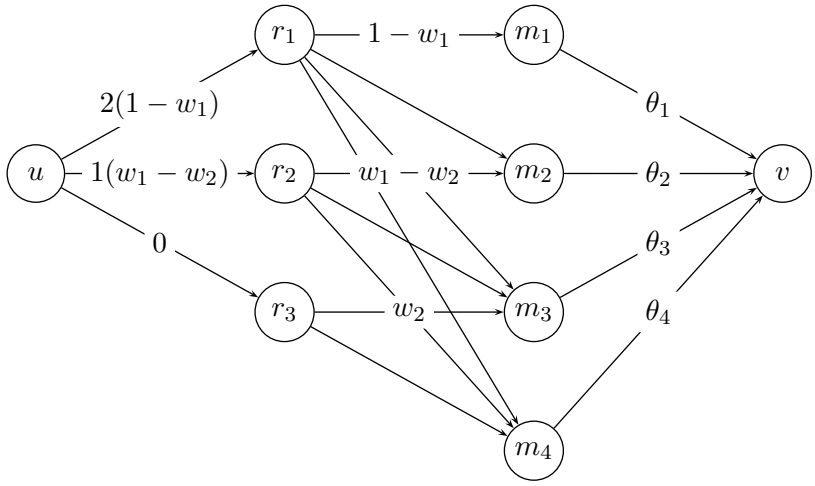

Figure 2: Example of network G with $p = 2$ and $N = 4$.

Now, consider a minimum $u - v$ cut in the network G, defined by a node set $U \subset V$ with $u \in U$ and $v \notin U$. Let $S = \{m_j : j \in \mathcal{N} \setminus U\}$. Note that if $r_k \notin U$ we obtain an arc in the cut, from u to r_k, with capacity $(p - k + 1)(w_{k-1} - w_k)$, whereas if $r_k \in U$, we obtain a set of arcs in the cut, from r_k to m_j for $j \in S$ such that $j \geq k$, with total capacity

$$\sum_{j\in S\cap\{k,\ldots,N\}} (w_{k-1} - w_k) = |S \cap \{k, \ldots, N\}|(w_{k-1} - w_k).$$

Thus, because $w_{k-1} \geq w_k$ we can assume that in this minimum $u-v$ cut $r_k \in U$ if and only if $|S\cap\{k, \ldots, N\}| < p-k+1$. Hence, if we let $l = \min\{k = 1, \ldots, p+1 : |S\cap\{k, \ldots, N\}| \geq p-k+1\}$ then we can assume $r_k \in U$ for $1 \leq k < l$ and $r_k \notin U$ for $l \leq k \leq p + 1$.

We now show that $S \subseteq \{l, \ldots, N\}$. Indeed, suppose $j < l$. If $j \in S$ then the cut includes arcs from r_k to m_j with capacity $(w_{k-1} - w_k)$ for all $1 \leq k \leq j$ yielding a total capacity of $1 - w_j$. If $j \notin S$, then the cut includes an arc from m_j to v with capacity $\theta_j = z_j - w_j$.

Because $z_j \leq 1$, this implies we can assume that in this minimum $u - v$ cut if $j < l$, then $j \notin S$.

Now suppose that $l = 1$, which occurs if $|S| \geq p$. Then the value of the minimum cut is given by

$$\sum_{k=1}^{p+1}(p-k+1)(w_{k-1}-w_k) + \sum_{j\notin S}\theta_j = p - \sum_{k=1}^{p} w_k + \sum_{j\notin S}\theta_j$$
$$\geq (p - \sum_{j\in S} z_j) + \sum_{j\in\mathcal{N}}\theta_j \geq \sum_{j\in\mathcal{N}}\theta_j$$

since $\sum_{j\in\mathcal{N}} z_j \leq p$. Thus, in this case, the value of the minimum cut is at least $\sum_{j\in\mathcal{N}}\theta_j$. So now assume $l > 1$. In this case, we claim that $|S| = p - l + 1$. Indeed, if not, then $|S| > p - l + 1$, and so $|S \cap \{l-1, \ldots, N\}| \geq p - (l-1) - 1$, contradicting the minimality in the definition of l since $l - 1$ also satisfies the condition in the definition.

The capacity of this minimum $u - v$ cut is

$$C = \sum_{k=1}^{l-1}(p-k+1)(w_{k-1}-w_k) + \sum_{k=l}^{p+1}|S|(w_{k-1}-w_k) + \sum_{j\in\mathcal{N}\setminus S}\theta_j.$$

Since,

$$\sum_{k=1}^{l-1}(p-k+1)(w_{k-1}-w_k) = \sum_{k=l}^{p}\sum_{j=k}^{p}(w_{k-1}-w_k) = \sum_{j=l}^{p}(w_{l-1}-w_j)$$

it follows that

$$C = (p-l+1)w_{l-1} - \sum_{k=l}^{p} w_k + (1-w_{l-1})|S| + \sum_{j\in\mathcal{N}\setminus S}\theta_j$$
$$= (p-l+1) - \sum_{k=l}^{p} w_k + \sum_{j\in\mathcal{N}\setminus S}\theta_j \geq \sum_{j\in\mathcal{N}}\theta_j$$

by (107) for $k = l$ since $S \subseteq \{l, \ldots, N\}$ and $|S| = p - l + 1$. □

We close this section by noting that inequalities (107) can be separated in polynomial time. Indeed, suppose we wish to separate the point (z^*, w^*). Then separation can be accomplished by calculating

$$V_k^* = \max_{S\in\mathcal{S}_k}\left\{\sum_{i\in S} z_i^* + \sum_{i\in\{k,\ldots,p\}\setminus S} w_i^*\right\} = \max_{S\in\mathcal{S}_k}\left\{\sum_{i\in S}\theta_i^*\right\} + \sum_{i=k}^{p} w_i^*$$

for $k = 1, \ldots, p$ where $\theta_i^* = z_i^* - w_i^*$ for $i = 1, \ldots, p$ and $\theta_i^* = z_i^*$ for $i = p+1, \ldots, N$. If $V_k^* > p-k+1$ for any k, then a violated inequality is found. Hence, a trivial separation algorithm is to first sort the values θ_i^* in non-increasing order, then for each k, find the maximizing set $S \in \mathcal{S}_k$ by searching this list. This yields an algorithm with complexity $O(N \log N + p^2) = O(N^2)$. However, the complexity can be improved to $O(N \log N)$ as follows. Start by storing the p largest values of θ_i^* over $i \in \{p+1, \ldots, N\}$ in a heap, and define $V_{p+1}^* = 0$. Then, for $k = p, \ldots, 1$ do the following. First insert θ_k^* into this heap. Next remove the largest value, say $\theta_{\max}^*$, from the heap and finally calculate V_k^* by

$$V_k^* = V_{k+1}^* + \max\{\theta_{\max}^*, 0\} + w_k^*.$$

The initial heap construction is accomplished with complexity $O(N \log N)$, and the algorithm then proceeds through p steps, each requiring insertion into a heap and removal of the maximum value from a heap, which can each be done with $O(\log p)$ complexity, yielding overall complexity of $O(N \log N)$. For general probabilities π_i, (heuristic) separation of inequalities (106) can be accomplished by (heuristically) solving p knapsack problems.

3.5 Sample Approximation Experiments

We conducted experiments to test the effectiveness of the sample approximation approach for yielding good feasible solutions and lower bounds. In particular, our aim is to determine whether using $\alpha > 0$ in the sample approximation can yield better solutions than when using $\alpha = 0$ as in the scenario approximation approach of [16, 51]. In addition, we test whether reasonable lower bounds which are valid with high probability can be obtained. We first conducted tests on a probabilistic version of the classical set covering problem, which has been studied recently in [14, 63, 62]. This problem has both finite feasible region and finite distribution (although both are exponentially large) so that for generating feasible solutions, the stronger Theorems 24 and 26 apply. These results are given in Section 3.5.1. We also conducted tests on a probabilistic version of the transportation problem. For this problem, the feasible region is continuous and we also use a joint normal distribution for the right-hand side vector, so that Theorem 27 applies. These results are presented in Section 3.5.2.

Note that although Theorem 22 provides support for using the sample approximation scheme to generate lower bounds, we will use Theorem 23 to actually obtain lower bounds which are valid with high confidence, because it can be used regardless of how many scenarios N are used (with the possible drawback that using smaller N will yield weaker lower bounds). Similarly, Theorems 24, 26, and 27 support the use of sample approximation to yield feasible solutions, but we do not use these Theorems to guide our choice of α and N. Indeed, the bounds implied by these theorems would suggest using N which is far too large to be able to solve the approximation problem. Instead, we experiment with different values of α and N, and perform an *a posteriori* test on each solution generated to determine whether it is feasible (with high confidence).

3.5.1 Probabilistic Set Cover Problem

The Probabilistic Set Cover Problem is given by

$$\min\{cx : P\{Ax \geq \xi\} \geq 1 - \epsilon, x \in \{0,1\}^n\} \tag{PSC}$$

where $c \in \mathbf{R}^n$ is the cost vector, A is an $m \times n$ zero-one matrix and ξ is a random vector taking values in $\{0,1\}^m$. We conducted tests on a single instance of PSC, with two values of ϵ, 0.05 and 0.1.

Test Instance

Following [14], we based our tests on a deterministic set covering instance, scp41, of the OR library [10], which has $m = 200$ rows and $n = 1000$ columns. Also following [14], the random vector ξ is assumed to consist of 20 independent sub-vectors, with each sub-vector having size $k = 10$ following the *circular* distribution. The circular distribution is defined by parameters $\lambda_j \in [0,1]$ for $j = 1, \ldots, k$. First, Bernoulli random variables Y_j for $j = 1, \ldots, k$ are generated independently with $P\{Y_j = 1\} = \lambda_i$. Then, the random sub-vector is defined by $\xi_j = \max\{Y_j, Y_{j+1}\}$ for $j < k$ and by $\xi_k = \max\{Y_1, Y_k\}$. We use the same parameters as in [14], which are given in the preprint [12]. Because of the simple form of this distribution, given a solution x, it is possible to calculate exactly $P\{Ax \geq \xi\}$. Thus, when a solution is obtained from a sample approximation problem, we test a posteriori whether it is feasible at

a given risk level by exactly calculating $P\{Ax \geq \xi\}$. To illustrate this calculation, we show how to calculate the probability for a single sub-vector, that is $P\{\xi_j \leq y_j,\ j = 1, \ldots, k\}$. Then, with $y = Ax$, the overall probability $P\{Ax \geq \xi\}$ is calculated as the product of the probabilities for each sub-vector. Let $J = \{1 \leq j \leq k : y_j = 0\}$. Then,

$$P\{\xi_j \leq y_j,\ j = 1, \ldots, k\} = P\{\xi_j = 0, \forall j \in J\} = P\{Y_j = 0, \forall j \in J^+\} = \prod_{j \in J^+} (1 - \lambda_j)$$

where $J^+ = \cup_{j \in J}\{j, (j+1) \mod k\}$. Although in this test, calculation of the distribution function is easy, we stress that this is not a necessary condition to use the sample approximation, it is only necessary that sampling from the distribution can be done efficiently.

Solving the Sample Approximation

To solve the sample approximation problem of the PSC, we used a MIP formulation which is equivalent to the extended formulation studied in Section 3.4.3. The formulation is not exactly the same, since because the random right-hand side can take on only two values, it can be simplified somewhat. Let the scenarios obtained in the sample of size N be denoted by ξ^i for $i = 1, \ldots, N$, where each $\xi^i \in \{0,1\}^m$. Then, the formulation we use is

$$\begin{aligned}
\min \quad & cx \\
\text{s.t.} \quad & Ax \geq y \\
& y_j + z_i \geq 1 \qquad \forall i, j \text{ s.t. } \xi^i_j = 1 && (123) \\
& \sum_{i=1}^{N} z_i \leq p && (124) \\
& x \in \{0,1\}^n,\ z \in \{0,1\}^N,\ y \in \{0,1\}^m
\end{aligned}$$

where $p = \lfloor \alpha N \rfloor$. We could relax the intregrality restriction on the y variables, but we found that leaving this restriction and also placing higher branching priority on these variables significantly improved performance when solving with CPLEX 9.0. The intuition behind this is that if we fix $y_j = 1$, then we are enforcing the constraint $A^j x \geq 1$, and on the other hand, if we fix $y_j = 0$, then any scenario i for which $\xi^i_j = 1$ will be fixed to 1, and constraint (124) will quickly become binding. We also found that some simple preprocessing

of the formulation significantly helped solution times. If, for a row j, $\sum_i \xi_j^i > p$, then we cannot have $y_j = 0$, and so we fixed $y_j = 1$, and the corresponding inequalities (123) for j were not included. For our tests this happened very often. After this preprocessing, for each j there will be at most p inequalities in (123), so that these inequalities add at most mp rows and $O(mp)$ nonzeros to the formulation. Using this formulation, we found the sample approximation problems could be solved quickly, in all cases in less than ten seconds, and usually much less. However, this may be due to the particular distribution used, which yielded sample approximation problems in which very many of the y_j variables could be fixed to one, and thus this should not be taken as a study of the effectiveness of this formulation in general. Rather, we are interested here only in the properties of the solutions generated by the sample approximation problems.

Feasible Solutions

We first tested the effectiveness of the sample approximation approach for generating feasible solutions. To do so, we varied the risk level of the approximation problem, α, and sample size, N. For each combination of α and N we generated and solved 10 sample approximation problems. Table 15 gives statistics of the solutions generated for the PSC instance with $\epsilon = 0.05$, and Table 16 gives the same for the PSC instance with $\epsilon = 0.1$. For each combination of α and N, we report statistics on the risk of the generated solutions, where for a solution x, the risk is $P\{Ax \not\geq \xi\}$, as well as on the costs of the *feasible* solutions generated, i.e. those solutions which have risk less than 0.05 and 0.1 respectively. For the risk of the solutions, we report the average, minimum, maximum and sample standard deviation over the 10 solutions. For the solutions costs, we report first how many solutions were feasible, then report the average, minimum, maximum and sample standard deviation of the cost taken over these solutions.

We first discuss results for the case of nominal risk level $\epsilon = 0.05$. When using $\alpha = 0$, the best results were obtained with N in the range of 80-130, and these are the results we report. With $\alpha = 0$, as N increases more constraints are being enforced, which leads to smaller feasible region of the approximation and higher likelihood that the optimal solution of the

Table 15: Solution results for PSC sample problems with $\epsilon = 0.05$.

		Solution Risk				Feasible Solutions Cost				
α	N	Ave	Min	Max	σ	#	Ave	Min	Max	σ
0.000	80	0.068	0.010	0.115	0.035	3	416.3	**406**	425	9.6
	90	0.067	0.010	0.108	0.031	3	414.0	**406**	418	6.9
	100	0.072	0.010	0.141	0.042	3	420.7	418	424	3.1
	110	0.029	0.000	0.074	0.023	9	422.9	413	429	5.8
	120	0.028	0.000	0.114	0.038	8	425.1	418	429	4.8
	130	0.021	0.000	0.044	0.018	10	426.0	409	429	6.3
0.045	1000	0.051	0.030	0.060	0.010	4	405.3	404	409	2.5
	3000	0.044	0.039	0.051	0.004	8	404.4	404	405	0.5
	5000	0.043	0.039	0.050	0.004	10	404.4	404	405	0.5
	7500	0.043	0.039	0.046	0.003	10	404.3	404	405	0.5
	10000	0.042	0.039	0.046	0.003	10	404.6	404	405	0.5
0.050	1000	0.058	0.044	0.070	0.009	1	404.0	404	404	***
	3000	0.052	0.046	0.059	0.004	3	404.0	404	404	0.0
	5000	0.050	0.046	0.055	0.003	4	404.0	404	404	0.0
	7500	0.049	0.044	0.051	0.003	5	404.0	404	404	0.0
	10000	0.047	0.044	0.051	0.003	9	404.0	404	404	0.0

Table 16: Solution results for PSC sample problems with $\epsilon = 0.1$.

		Solution Risk				Feasible Solutions Cost				
α	N	Ave	Min	Max	σ	#	Ave	Min	Max	σ
0.000	60	0.122	0.073	0.206	0.038	3	421.0	407	428	12.1
	70	0.089	0.030	0.206	0.051	7	415.7	407	425	6.8
	80	0.068	0.010	0.115	0.035	8	414.9	406	425	6.3
	90	0.067	0.010	0.108	0.031	9	414.6	**400**	428	8.3
	100	0.072	0.010	0.141	0.042	6	418.3	408	424	5.8
	110	0.029	0.000	0.074	0.023	10	423.0	413	429	5.5
0.095	1000	0.102	0.093	0.112	0.007	4	387.3	387	388	0.5
	3000	0.093	0.083	0.103	0.006	9	388.2	387	390	1.3
	5000	0.094	0.089	0.103	0.004	8	387.6	387	390	1.2
	7500	0.093	0.085	0.103	0.004	9	387.6	387	390	1.0
	10000	0.090	0.083	0.093	0.005	10	387.9	387	390	1.4
0.100	1000	0.107	0.093	0.120	0.008	1	387.0	387	387	***
	3000	0.098	0.093	0.112	0.006	6	387.0	387	387	0.0
	5000	0.100	0.093	0.112	0.005	4	387.0	387	387	0.0
	7500	0.099	0.093	0.103	0.004	5	387.0	387	387	0.0
	10000	0.097	0.093	0.103	0.004	7	387.0	387	387	0.0

approximation is feasible at the nominal risk level. However, the smaller feasible region also causes the cost to increase, so that increasing N more would yield overly conservative solutions. We also conducted tests with $\alpha = 0.045$ and $\alpha = 0.05$, and for these values of α we used a significantly larger sample size. The best feasible solution found using $\alpha = 0$ had cost 406, and the average cost of the feasible solutions found was significantly greater than this. When $\alpha > 0$, a feasible solution with cost 404 was found in one of the ten runs for *every* combination of α and N. When $\alpha = \epsilon = 0.05$ was used, *every* feasible solution found had cost 404. Even when $\alpha = 0.045$ was used, in which case feasible solutions were found more consistently, the average cost of the feasible solutions was still less than the *minimum* cost of a feasible solution found using $\alpha = 0$.

For $\epsilon = 0.1$, we observed similar results. In this case, when using $\alpha = 0$, the best results were obtained with N in the range of 60-110. The best solution found using $\alpha = 0$ had cost 400, whereas the best solution found using $\alpha > 0$ was 387, which was obtained by one of the ten runs for *every* combination of $\alpha > 0$ and N.

In terms of the variability of the risks and costs of the solutions generated, using $\alpha > 0$ and a much larger sample size yielded solutions with much lower variability than when using $\alpha = 0$ and small sample size. This is not surprising since using a larger sample size naturally should reduce variability. On the other hand, constraining the sample approximation to have $\alpha = 0$ prohibits the use of a larger sample size, as the solutions produced then become overly conservative.

Lower Bounds

We next discuss the results for obtaining lower bounds for the PSC. We used the procedure of Theorem 23 with $\alpha = \epsilon$ and $M = 10$. We use the same 10 sample approximation problems as when generating feasible solutions. As argued after Theorem 23, with $\alpha = \epsilon$, we have $\rho(\alpha, \epsilon, N) = \rho(\epsilon, \epsilon, N) \gtrsim 1/2$. Then, if we take $L = 1$ the test of Theorem 23 yields a lower bound with confidence 0.999. Taking $L = 1$ corresponds to taking the minimum optimal value over all the $M = 10$ runs (not just over the ones which yielded feasible solutions). More generally, we can take $L \in \{1, \ldots, 10\}$ yielding a lower bound with confidence

at least

$$1 - \sum_{i=0}^{L-1} \binom{10}{i} \rho(\epsilon, \epsilon, N)^i \big(1 - \rho(\epsilon, \epsilon, N)\big)^{10-i} \gtrsim 1 - \sum_{i=0}^{L-1} \binom{10}{i} (1/2)^{10}$$

to obtain possibly "tighter" lower bounds of which we are less confident.

Table 17: Lower bounds for PSC sample problems with $\alpha = \epsilon = 0.05$.

	LB with confidence at least:				Gap with confidence at least:			
N	0.999	0.989	0.945	0.828	0.999	0.989	0.945	0.828
1000	395	396	397	398	2.2%	2.0%	1.7%	1.5%
3000	399	400	400	400	1.2%	1.0%	1.0%	1.0%
5000	400	400	400	400	1.0%	1.0%	1.0%	1.0%
7500	400	400	400	400	1.0%	1.0%	1.0%	1.0%
10000	400	404	404	404	1.0%	0.0%	0.0%	0.0%

The results obtained using varying values of N and $\epsilon = \alpha = 0.05$ are given in Table 17. The gaps reported are the percent by which the lower bound is below the best feasible solution (404, obtained by any of the tested combinations of $\alpha > 0$ and N). Thus, for example, by solving 10 problems with sample size $N = 1000$ we obtained a feasible solution of cost 404, and lower bound of 397, which is valid with probability at least 0.989. Solving 10 sample problems with sample size $N = 5000$ also yields a feasible solution of cost 404, but a lower bound of 400, valid with probability 0.999. By solving 10 sample problems with $N = 10000$, we obtain a feasible solution of cost 404, and confidence at least 0.989 that this is actually the optimal solution to PSC at risk level $\epsilon = 0.05$.

Table 18 yields the lower bound results obtained with $\epsilon = \alpha = 0.1$ and varying sample size N. Solving 10 sample problems with $N = 1000$ we obtained a feasible solution of cost 387, and can say with confidence 0.989 that the optimal solution is at most 1.55% less costly than this solution. Using $N = 10000$, we obtain a feasible solution of the same cost, but a lower bound which states that with confidence at least 0.999 the optimal solution has cost at most one less (0.26%) than this solution.

3.5.2 Probabilistic Transportation Problem

We next tested the sampling approach on a probabilistic version of the classical transportation problem, which we call the Probabilistic Transportation Problem (PTP). In this

Table 18: Lower bounds for PSC sample problems with $\alpha = \epsilon = 0.1$.

	LB with confidence at least:				Gap with confidence at least:			
N	0.999	0.989	0.945	0.828	0.999	0.989	0.945	0.828
1000	381	382	384	385	1.55%	1.29%	0.78%	0.52%
3000	382	385	386	386	1.29%	0.52%	0.26%	0.26%
5000	382	385	386	386	1.29%	0.52%	0.26%	0.26%
7500	385	385	386	386	0.52%	0.52%	0.26%	0.26%
10000	386	386	386	387	0.26%	0.26%	0.26%	0.00%

problem, we have a set of suppliers I and a set of customers D with $|D| = m$. The suppliers have limited capacity M_i for $i \in I$. There is a transportation cost c_{ij} for shipping a unit of product from supplier $i \in I$ to customer $j \in D$. The customer demands are random and are represented by a random vector $\tilde{d}$ taking values in $\mathbf{R}^m$. We assume we must choose the shipment quantities before the customer demands are known. We enforce the probabilistic constraint

$$P\{\sum_{i \in I} x_{ij} \geq \tilde{d}_j, j = 1, \ldots, m\} \geq 1 - \epsilon \tag{125}$$

where $x_{ij} \geq 0$ is the amount shipped from supplier $i \in I$ to customer $j \in D$. The objective is to minimize distribution costs subject to (125), and the supply capacity constraints

$$\sum_{j \in D} x_{ij} \leq M_i, \quad \forall i \in I.$$

Test Instances

We conducted our tests on an instance with 40 suppliers and 50 customers. The supply capacities and cost coefficients were randomly generated using normal and uniform distributions respectively. The demand is assumed to have a joint normal distribution. The mean vector and covariance matrix were randomly generated. We considered two cases for the covariance matrix: a low variance and a high variance case. In the low variance case, the standard deviation of the one dimensional marginal random demands is 10% of the mean on average. In the high variance case, the covariance matrix of the low variance case is multiplied by 25, yielding standard deviations of the one dimensional marginal random demands being 50% of the mean on average. In both cases, we consider a single risk level

$\epsilon = 0.05$.

We remark that for this particular choice of distribution, the feasible region defined by the probabilistic constraint is convex [56]. However, the dimension of the random vector $\tilde{d}$ is $m = 50$, and so evaluating $P\{y \geq \tilde{d}\}$ for a single vector $y \in \mathbf{R}^m$ is difficult. Hence, applying variations of standard convex programming techniques will not likely be efficient. However, generating random samples from the joint normal distribution is easy so that generating (non-convex) sample approximation problems can be accomplished.

Once a sample approximation is solved yielding solution $\hat{x}$, we use a single very large sample ($N' = 250,000$), to estimate $P\{\hat{y} \geq \tilde{d}\}$ where $\hat{y} \in \mathbf{R}^m$ is the vector given by $\hat{y}_j = \sum_{i \in I} \hat{x}_{ij}$ for $j \in D$. Letting $d^1, \ldots, d^{N'}$ be the realizations of this large sample, we calculate $\sum_{i=1}^{N'} \mathbb{I}(\hat{y} \geq d^i)$ and use the normal approximation to the binomial distribution to construct an upper bound $\hat{\alpha}$ on the true solution risk $P\{\hat{y} \geq \tilde{d}\}$, which is valid with confidence 0.999. Henceforth for this experiment, if we say a solution is feasible at risk level ϵ, we mean $\hat{\alpha} \leq \epsilon$, and so it is feasible at this risk level with confidence 0.999. We used such a large sample to get a good estimate of the true risk of the solutions generated, but we note that because this sample was so large, generating this sample and calculating $\sum_{i=1}^{N'} \mathbb{I}(\hat{y} \geq d^i)$ often took longer than solving the sample approximation itself.

Solving the Sample Approximation

We solved the sample approximation problem using the formulation PMIP, augmented with the strengthened star inequalities. In Section 3.6 we present computational results of this and other formulations for solving the probabilistic transportation problem with finite distribution, so we will not discuss this more here. However, we should mention that in contrast to the probabilistic set cover problem, solving the sample approximation problem with the largest sample size we consider ($N = 10000$) and largest α (0.05) takes a nontrivial amount of time, in some cases as long as 30 minutes. On the other hand, for $N = 5000$, the worst case was again $\alpha = 0.05$ and usually took less than 4 minutes to solve.

Low Variance Instance

Table 19: Solution results for low variance PTP sample problems with $\epsilon = 0.05$.

		Solution Risk				Feasible Solutions Cost				
α	N	Ave	Min	Max	σ	#	Ave	Min	Max	σ
0.000	900	0.048	0.036	0.066	0.011	7	2.0266	2.0199	2.0320	0.0045
	950	0.047	0.039	0.055	0.005	6	2.0244	**2.0185**	2.0291	0.0041
	1000	0.045	0.040	0.051	0.004	8	2.0253	2.0185	2.0300	0.0039
	1500	0.033	0.025	0.043	0.005	10	2.0336	2.0245	2.0406	0.0053
0.030	5000	0.049	0.045	0.050	0.002	6	2.0098	2.0075	2.0114	0.0013
	7500	0.045	0.041	0.047	0.002	10	2.0112	2.0094	2.0136	0.0015
	10000	0.042	0.041	0.044	0.001	10	2.0129	2.0112	2.0145	0.0010
0.031	5000	0.050	0.047	0.051	0.002	4	2.0096	2.0090	2.0106	0.0007
	7500	0.046	0.043	0.049	0.002	10	2.0104	2.0086	2.0126	0.0015
	10000	0.043	0.042	0.045	0.001	10	2.0120	2.0104	2.0138	0.0010
0.032	5000	0.051	0.047	0.053	0.002	3	2.0089	2.0083	2.0097	0.0007
	7500	0.047	0.045	0.050	0.002	9	2.0096	2.0077	2.0117	0.0014
	10000	0.044	0.043	0.046	0.001	10	2.0111	2.0096	2.0129	0.0010
0.033	5000	0.052	0.049	0.054	0.002	2	2.0080	2.0073	2.0088	0.0011
	7500	0.048	0.045	0.051	0.002	7	2.0092	2.0075	2.0107	0.0012
	10000	0.045	0.044	0.047	0.001	10	2.0103	2.0089	2.0118	0.0009
0.034	5000	0.053	0.050	0.055	0.002	0	***	***	***	***
	7500	0.049	0.046	0.052	0.002	5	2.0089	2.0079	2.0098	0.0008
	10000	0.046	0.045	0.048	0.001	10	2.0095	2.0082	2.0110	0.0008
0.035	5000	0.054	0.051	0.056	0.002	0	***	***	***	***
	7500	0.051	0.047	0.053	0.002	4	2.0083	2.0073	2.0091	0.0008
	10000	0.048	0.045	0.050	0.001	10	2.0086	2.0073	2.0101	0.0008
0.036	5000	0.055	0.053	0.057	0.002	0	***	***	***	***
	7500	0.052	0.049	0.054	0.002	2	2.0079	2.0077	2.0080	0.0002
	10000	0.049	0.047	0.051	0.001	8	2.0080	**2.0066**	2.0093	0.0008

We begin by presenting results for the instance in which the distribution of demand has relatively low variance. For generating feasible solutions, we tested $\alpha = 0$ with various sample size N and report the results for the sample sizes which yielded the best results. Once again, this means we use a relatively small sample size for the case $\alpha = 0$, as compared to the cases with $\alpha > 0$. We tested several values of $\alpha > 0$ and varying sample size. In contrast to the PSC case, we found that taking $\alpha = \epsilon$ or even α close to ϵ did not yield feasible solutions, even with a large sample size. Thus, we report results for many different values of α in the range 0.03 to 0.036. The primary reason for reporting results for this many different values of α is to illustrate that within this range, the results are not extremely sensitive to the choice of α.

Table 20: Lower bounds for low variance PTP sample problems with $\alpha = \epsilon = 0.05$.

	LB with confidence at least:				Gap with confidence at least:			
N	0.999	0.989	0.945	0.828	0.999	0.989	0.945	0.828
1000	1.9755	1.9757	1.9775	1.9782	1.55%	1.54%	1.45%	1.42%
3000	1.9879	1.9892	1.9892	1.9910	0.93%	0.87%	0.87%	0.78%
5000	1.9940	1.9943	1.9948	1.9951	0.63%	0.62%	0.59%	0.57%
7500	1.9954	1.9956	1.9959	1.9963	0.56%	0.55%	0.54%	0.52%
10000	1.9974	1.9977	1.9980	1.9981	0.46%	0.45%	0.43%	0.42%

Table 19 gives the characteristics of the solutions generated for the different values of α and N. We observe that as in the case of the PSC, the *average* cost of the feasible solutions obtained using $\alpha > 0$ is always less than the *minimum* cost of the feasible solutions obtained with $\alpha = 0$. However, for this instance, the minimum cost solution obtained using $\alpha = 0$ is not so significantly worse than the minimum cost solutions using different values of $\alpha > 0$, being between 0.40% and 0.58% more costly. As in the case of the PSC, using $\alpha > 0$ and large N significantly reduced the variability of the risk and cost of the solutions generated.

We next investigated the quality of the lower bounds that can be obtained for PTP by solving sample approximation problems. As in the case of the PSC, we obtained lower bounds by generating and solving 10 sample approximation problems with $\alpha = \epsilon = 0.05$. By taking the lowest value of all the optimal values we obtain a lower bound valid with

confidence 0.999, taking the second smallest yields a lower bound which is valid with confidence 0.989, etc. The results for different values of N are given in Table 20. For reference, the percentage gap between these lower bounds and the best feasible solution found (with cost 2.0066) is also given. Using $N \geq 3000$ we obtain lower bounds that are valid with confidence 0.999 and are within one percent of the best feasible solution, indicating that for this low variance instance, the lower bounding scheme yields good evidence that the solutions we have found are good quality.

High Variance Instance

Table 21: Solution results for high variance PTP sample problems with $\epsilon = 0.05$.

		Solution Risk				Feasible Solutions Cost				
α	N	Ave	Min	Max	σ	#	Ave	Min	Max	σ
0.000	900	0.050	0.035	0.066	0.010	4	3.5068	3.4672	3.5488	0.0334
	950	0.050	0.041	0.058	0.006	6	3.4688	**3.4403**	3.4917	0.0191
	1000	0.045	0.041	0.052	0.004	9	3.4895	3.4569	3.5167	0.0234
	1500	0.030	0.022	0.035	0.005	10	3.5494	3.5205	3.6341	0.0368
0.030	5000	0.050	0.045	0.053	0.002	4	3.4014	3.3897	3.4144	0.0101
	7500	0.046	0.043	0.050	0.002	9	3.4060	3.3920	3.4235	0.0098
	10000	0.043	0.041	0.046	0.001	10	3.4139	3.4001	3.4181	0.0055
0.031	5000	0.051	0.044	0.054	0.003	2	3.4094	3.3991	3.4197	0.0146
	7500	0.047	0.044	0.051	0.002	8	3.4016	3.3958	3.4118	0.0065
	10000	0.044	0.040	0.047	0.002	10	3.4083	3.3983	3.4247	0.0089
0.032	5000	0.052	0.045	0.055	0.003	2	3.4049	3.3945	3.4152	0.0146
	7500	0.048	0.046	0.052	0.002	8	3.3965	3.3910	3.4064	0.0062
	10000	0.045	0.041	0.048	0.002	10	3.4032	3.3934	3.4193	0.0087
0.033	5000	0.053	0.046	0.057	0.003	1	3.4107	3.4107	3.4107	***
	7500	0.049	0.046	0.054	0.002	7	3.3928	3.3865	3.4020	0.0062
	10000	0.046	0.042	0.049	0.002	10	3.3982	3.3885	3.4139	0.0086
0.034	5000	0.054	0.047	0.058	0.003	1	3.4064	3.4064	3.4064	***
	7500	0.050	0.047	0.055	0.002	5	3.3903	**3.3817**	3.3971	0.0056
	10000	0.048	0.044	0.050	0.002	10	3.3934	3.3840	3.4084	0.0085
0.035	5000	0.056	0.048	0.059	0.003	1	3.4022	3.4022	3.4022	***
	7500	0.052	0.049	0.056	0.002	2	3.3903	3.3881	3.3925	0.0031
	10000	0.049	0.045	0.051	0.002	6	3.3936	3.3861	3.4034	0.0072
0.036	5000	0.057	0.049	0.060	0.003	1	3.3979	3.3979	3.3979	***
	7500	0.053	0.050	0.057	0.002	0	***	***	***	***
	10000	0.050	0.046	0.053	0.002	4	3.3927	3.3859	3.3986	0.0054

Table 21 gives the characteristics of the solutions generated for the high variance instance. In this case, the *maximum* cost of a feasible solution generated using any combination of $\alpha > 0$ and N was less than the *minimum* cost of any feasible solution generated using $\alpha = 0$. The minimum cost feasible solution generated with $\alpha = 0$ was between 0.87% and 1.7% more costly than the best feasible solution generated for the different combinations of $\alpha > 0$ and N. Thus, it appears that for the high variance instance, using $\alpha > 0$ in a sample approximation is more important for generating good feasible solutions than for the low variance instance.

Table 22 gives the lower bounds for different confidence levels and sample sizes, as well as the gaps between these lower bounds and the best feasible solution found. In this case, solving 10 instances with sample size $N = 1000$ yields a lower bound that is not very tight, 5.11% from the best solution cost at confidence level 0.999. Increasing the sample size improves the lower bound, but even with $N = 10000$, the gap between the lower bound at confidence 0.999 and the best solution found is 1.83%. Thus, it appears that for the high variance instance, the sample approximation scheme exhibits considerably slower convergence, in terms of the lower bounds, the feasible solutions generated, or both.

Table 22: Lower bounds for high variance PTP sample problems with $\alpha = \epsilon = 0.05$.

	LB with confidence at least:				Gap with confidence at least:			
N	0.999	0.989	0.945	0.828	0.999	0.989	0.945	0.828
1000	3.2089	3.2158	3.2178	3.2264	5.11%	4.91%	4.85%	4.59%
3000	3.2761	3.2775	3.2909	3.2912	3.12%	3.08%	2.69%	2.68%
5000	3.3060	3.3075	3.3077	3.3094	2.24%	2.19%	2.19%	2.14%
7500	3.3083	3.3159	3.3165	3.3169	2.17%	1.95%	1.93%	1.92%
10000	3.3200	3.3242	3.3284	3.3299	1.83%	1.70%	1.58%	1.53%

3.6 MIP Approach for Solving the Sample Approximation

We performed computational tests on the probabilistic transportation problem introduced in Section 3.5.2. We randomly generated instances with the number of suppliers fixed at 40 and varying numbers of customers and scenarios. The supply capacities and cost coefficients were generated using normal and uniform distributions respectively. For the random demands,

we experimented with independent normal, dependent normal and independent Poisson distributions. We found qualitatively similar results in all cases, but the independent normal case yielded the most challenging instances, so for our experiments we focus on this case. For each instance, we first randomly generated the mean and variance of each customer demand. We then generated the number N of scenarios required, independently across scenarios and across customer locations, as Monte Carlo samples with these fixed parameters. In most instances we assumed all scenarios occur with probability $1/N$ (corresponding to the sample approximation problem), but we also did some tests in which the scenarios have general probabilities, which were randomly generated. CPLEX 9.0 was used as the MIP solver and all experiments were done on a computer with two 2.4 Ghz processors (although no parallelism is used) and 2.0 Gb of memory. We set a time limit of one hour. For each problem size we generated 5 random instances and, unless otherwise specified, the computational results reported are averages over the 5 instances.

3.6.1 Comparison of Formulations

In Table 23 we compare the results obtained by solving our instances using

1. formulation PMIP given by (77) - (79),
2. formulation PMIP with strengthened star inequalities (85), and
3. the extended formulation of Sect. 3.4.3, but without (106) or (107).

When the strengthened star inequalities are not used, we still used the improved formulation of G corresponding to (84). Recall that the strengthened star inequalities subsume the rows defining the formulation PMIP; therefore, when using these inequalities we initially added only a small subset of the *mp* inequalities (83) in the formulation. Subsequently separating the strengthened star inequalities as needed guarantees the formulation remains valid. For formulation PMIP without strengthened star inequalities, we report the average optimality gap that remained after the hour time limit was reached, where we define the optimality gap as the difference between the final upper and lower bounds, divided by the upper bound. For the other two formulations, which we refer to as the strong formulations, we report

the geometric average of the time to solve the instances to optimality. We used $\epsilon = 0.05$ and $\epsilon = 0.1$, reflecting the natural assumption that we want to meet demand with high probability.

Table 23: Average solution times for different formulations.

				PMIP	PMIP+Star		Extended
Probabilities	ϵ	m	N	Gap	Cuts	Time(s)	Time(s)
Equal	0.05	100	1000	0.18%	734.8	7.7	1.1
		100	2000	1.29%	1414.2	31.8	4.6
		200	2000	1.02%	1848.4	61.4	12.1
		200	3000	2.56%	2644.0	108.6	12.4
	0.10	100	1000	2.19%	1553.2	34.6	12.7
		100	2000	4.87%	2970.2	211.3	41.1
		200	2000	4.48%	3854.0	268.5	662.2
		200	3000	5.84%	5540.8	812.7	490.4
General	0.05	100	1000	0.20%	931.8	9.0	3.9
		100	2000	1.04%	1806.6	55.2	13.2
	0.10	100	1000	1.76%	1866.0	28.7	52.5
		100	2000	4.02%	3686.2	348.5	99.2

The first observation from Table 23 is that formulation PMIP without the strengthened star inequalities failed to solve these instances within an hour, often leaving large optimality gaps, whereas the instances are solved efficiently using the strong formulations. The number of nodes required to solve the instances for the strong formulations is very small. The instances with equi-probable scenarios were usually solved at the root, and even when branching was required, the root relaxation usually gave an exact lower bound. Branching in this case was only required to find an integer solution which achieved this bound. The instances with general probabilities required slightly more branching, but generally not more than 100 nodes. Observe that the number of strengthened star inequalities that were added is small relative to the number of rows in the formulation PMIP itself. For example, with equi-probable scenarios, $\epsilon = 0.1$, $m = 200$ and $N = 3,000$, the number of rows in PMIP would be $mp = 60,000$, but on average, only $5,541$ strengthened star inequalities were added. Next we observe that in most cases the computation time using the extended formulation is significantly less than the formulation with strengthened star inequalities. Finally, we observe that the instances with general probabilities take somewhat longer to

solve than those with equi-probable scenarios but can still be solved efficiently.

3.6.2 The Effect of Increasing ϵ

The results of Table 23 indicate that the strong formulations can solve large instances to optimality when ϵ is small, which is the typical case. However, it is still an interesting question to investigate how well this approach works for larger ϵ. Note first that we should expect solution times to grow with ϵ if only because the formulation sizes grow with ϵ. However, we observe from the chart in Figure 3 that the situation is much worse than this. This chart shows the root LP solve times and optimality gaps achieved after an hour of computation time for an example instance with equi-probable scenarios, $m = 50$ rows and $N = 1,000$ scenarios at increasing levels of ϵ, using the formulation PMIP with strengthened star inequalities. Root LP solve time here refers to the time until no further strengthened star inequalities could be separated. We see that the time to solve the root linear programs does indeed grow with ϵ as expected, but the optimality gaps achieved after an hour of computation time deteriorate drastically with growing ϵ. This is explained by the increased time to solve the linear programming relaxations *combined with* an apparent weakening of the relaxation bound as ϵ increases.

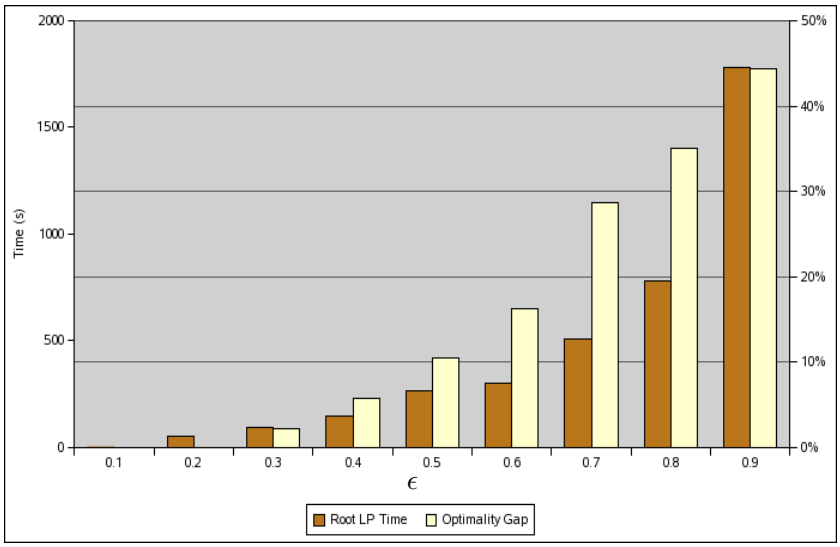

Figure 3: The effect of increasing ϵ.

Table 24: Results with and without inequalities (107).

			Root Gap		Nodes		Time(s) or Gap	
m	ϵ	N	Ext	+(107)	Ext	+(107)	Ext	+(107)
25	0.3	250	1.18%	0.67%	276.9	69.0	121.2	93.9
	0.3	500	1.51%	0.58%	455.0	165.8	750.6	641.3
	0.35	250	2.19%	1.50%	1259.4	409.0	563.2	408.4
	0.35	500	2.55%	1.61%	2297.6	968.8	0.22%	0.06%
50	0.3	500	2.32%	2.00%	991.8	238.6	1.37%	1.41%
	0.3	1000	2.32%	1.75%	28.3	8.5	1.98%	1.66%
	0.35	500	4.10%	3.31%	650.4	92.9	3.03%	2.66%
	0.35	1000	4.01%	3.23%	22.7	6.2	3.58%	3.17%

3.6.3 **Testing Inequalities** (107)

With small ϵ the root relaxation given by the extended formulation is extremely tight, so that adding the inequalities (107) is unlikely to have a positive impact on computation time. However, for larger ϵ, we have seen that formulation PMIP, augmented with the strengthened star inequalities, and hence also the extended formulation, may have a substantial optimality gap. We therefore investigated whether using inequalities (107) in the extended formulation can improve solution time in this case. In Table 24 we present results comparing solution times and node counts with and without inequalities (107) for instances with larger ϵ. We performed these tests on smaller instances since these instances are already hard for these values of ϵ. We observe that adding inequalities (107) at the root can decrease the root optimality gap significantly. For the instances that could be solved in one hour, this leads to a significant reduction in the number of nodes explored, and a moderate reduction in solution time. For the instances which were not solved in one hour, the remaining optimality gap was usually, but not always, lower when the inequalities (107) were used. These results indicate that when ϵ is somewhat larger, inequalities (107) may be helpful on smaller instances. However, they also reinforce the difficulty of the instances with larger ϵ, since even with these inequalities, only the smallest of these smaller instances could be solved to optimality within an hour.

3.7 Concluding Remarks

We have studied a sample approximation scheme for probabilistically constrained optimization problems and demonstrated how this scheme can be used to generate optimality bounds and feasible solutions for very general optimization problems with probabilistic constraints. We have also conducted a preliminary computational study of this approach. This study demonstrates that using sample approximation problems that allow a choice of which sampled constraints to satisfy can yield good quality feasible solutions. In addition, the sample approximation scheme can be used to obtain lower bounds which are valid with high confidence. We found that good lower bounds could be found in the case of finite (but possibly exponential) feasible region and distribution, and also in the case of continuous feasible region and distribution, provided the distribution has reasonably low variance. With continuous feasible region and distribution, if the distribution has high variance the lower bounds were relatively weak. Future work in this area will include conducting more extensive computational tests, and also extending the theory to allow generation of samples which are not necessarily independent and identically distributed. For example, the use of Latin hypercube sampling may yield significantly faster convergence.

The sample approximation problem we have studied is non-convex, and hence may in general be difficult to solve. However, for the special case in which only the right-hand side of the constraints is random, we have studied a mixed-integer programming formulation which appears to be very effective. The MIP formulation we study can be used when the different realizations have general probabilities, not just equal probabilities as in the sample approximation case. In our polyhedral study of the MIP formulation we made use of existing results on a mixed-integer substructure appearing in the formulation known as a *mixing set*. We have also introduced new polyhedral results for the case in which the mixing set additionally has a knapsack restriction. We have conducted computational experiments for the case of a linear program with probabilistic constraints. These results indicate that these formulations are extremely effective on instances in which relatively low risk of violation of the constraints is allowed, which is the typical case. However, instances in which the desired risk level is higher remain difficult to solve, partly due to increased

size of the formulations, but more significantly due to the weakening of the formulation bounds. Moreover, these instances remain difficult even when using the inequalities which characterize the single row relaxation convex hull. This suggests that relaxations which consider multiple rows simultaneously need to be studied to yield valid inequalities which significantly improve the relaxation bounds for these instances.

The sample approximation scheme we have developed is quite general, but so far we have only demonstrated that the sample approximation problem can be solved efficiently in the case that only the right-hand side of the probabilistic constraints is random. Future work will attempt to find methods to solve the sample approximation problem (that is, a probabilistic program with finite distribution) in more general settings. The most obvious extension will be to allow the constraint matrix in a probabilistically constrained linear program to be random. A natural first step in this direction will be to extend results from the *generalized* mixing set [48, 69] to the case in which an additional knapsack constraint is present. An alternative, or perhaps a supplement, to this approach is to apply the ideas of dominance given in [61] to the single row formulation. We say that scenario i dominates another scenario k if the constraints of scenario k are automatically satisfied whenever the constraints of scenario i are satisfied. In such a case, we can add the precedence constraint, $z_i \geq z_k$, to the formulation. The idea given in [61] was to find such dominance relations, and then combine the corresponding precedence constraints with the knapsack inequality to find strong valid inequalities. For moderate size problems it seems unlikely that dominance between scenarios to be common, and hence this approach may not be that attractive. However, suppose we take the approach given in the extended formulation of Section 3.4.3, and for each row j in our formulation we introduce a set of binary variables representing whether or not each scenario is satisfied *for that row*. Now, in applications we may expect that for each row the number of random coefficients may not be very large. Therefore, on a row-by-row basis, we may find that dominance between scenarios is common, and hence adding precedence constraints and valid inequalities based on them combined with the knapsack inequality may be fruitful.

CHAPTER IV

NEW FORMULATIONS FOR STOCHASTIC PROGRAMMING UNDER STOCHASTIC DOMINANCE CONSTRAINTS

4.1 Introduction

Stochastic programming under stochastic dominance constraints is an attractive approach to managing risk in an optimization setting. The idea is to optimize an objective, such as the expected profit, subject to a constraint that a random outcome of interest, such as the actual profit, is "better" in a strong sense than a given reference random outcome. Here, "better" is taken to mean that the random outcome we achieve stochastically dominates the reference outcome. A simple example application is to choose investments so that you maximize the expected return, subject to the constraint that the actual return should stochastically dominate the return from a given index, such as the S&P 500. Stochastic programming under stochastic dominance constraints has recently been studied in [22, 23, 24, 53, 54].

Let W and Y be random variables with distribution functions F and G. The random variable W dominates Y in the first order, written $W \succeq_{(1)} Y$, if

$$F(\eta) \leq G(\eta) \qquad \forall\ \eta \in \mathbf{R}. \tag{126}$$

The random variable W dominates Y in the second order, written $W \succeq_{(2)} Y$, if

$$\mathbb{E}\big[(\eta - W)^+\big] \leq \mathbb{E}\big[(\eta - Y)^+\big] \qquad \forall\ \eta \in \mathbf{R}. \tag{127}$$

If W and Y represent random outcomes for which we prefer larger values, stochastic dominance of W over Y implies a very strong preference for W. In particular, it is known that (see, e.g. [71]) $W \succeq_{(1)} Y$ if and only if

$$\mathbb{E}[h(W)] \geq \mathbb{E}[h(Y)]$$

for all nondecreasing functions $h : \mathbf{R} \to \mathbf{R}$ for which the above expectations exist and are finite. Thus, if $W \succeq_{(1)} Y$, any rational decision maker would prefer W to Y. In addition

$W \succeq_{(1)} Y$ if and only if

$$\mathbb{E}[h(W)] \geq \mathbb{E}[h(Y)]$$

for all nondecreasing and concave functions $h : \mathbf{R} \to \mathbf{R}$ for which the above expectations exist and are finite. Thus, if $W \succeq_{(2)} Y$, any rational and risk-averse decision maker will prefer W to Y.

In this chapter, we present new, computationally attractive formulations for optimization under stochastic dominance constraints. Let $X \subseteq \mathbf{R}^n$ represent a feasible region and $f : \mathbf{R}^n \to \mathbf{R}$ represent an objective we want to maximize. Let Y be a given random variable, which we refer to as the *reference random variable*, and let ξ be a random vector taking values in $\mathbf{R}^m$. Finally, let $g : \mathbf{R}^n \times \mathbf{R}^m \to \mathbf{R}$ be a given mapping which represents a random outcome depending on the decision x and the random vector ξ. We consider the two optimization problems

$$\max_x \left\{ f(x) : x \in X, g(x, \xi) \succeq_{(1)} Y \right\} \tag{FSD}$$

and

$$\max_x \left\{ f(x) : x \in X, g(x, \xi) \succeq_{(2)} Y \right\}. \tag{SSD}$$

We will present formulations for these problems when the random vector ξ and reference random variable Y have finite distributions. That is, we assume ξ can take at most N values, and Y can take at most D values. In particular,

1. We introduce a new linear formulation for SSD which has $O(N + D)$ constraints, as opposed to $O(ND)$ constraints in a previous linear formulation. Computational results indicate that this yields significant improvement in solution time.

2. We introduce a new mixed-integer programming (MIP) formulation for FSD which also has $O(N + D)$ constraints. In addition, the linear relaxation of this formulation is also a formulation of SSD. As a result, the linear programming relaxation of this formulation is equivalent to the SSD relaxation proposed in [53], and shown to be a tight relaxation of FSD in [54].

3. We present a specialized branching rule and heuristics for the new FSD formulation and conduct computational tests which indicate that provably good, and in some cases provably optimal, solutions can be obtained for large instances using this approach.

We do not make any assumptions on the set X or the mapping g in the development of our formulations, but computationally we are interested in the case when X is a polyhedron and $g(x, \xi)$ is affine in x for all possible values of ξ, so that the formulations become linear and linear integer programs, for SSD and FSD respectively.

In [24], it is shown that in some special cases the convex second order dominance constraint yields the convexification of the non-convex first order dominance constraint, and that in all cases, the second order constraint is a relaxation of the first order constraint. Our new formulations further illustrate this close connection by showing that relaxing integrality in our new formulation for FSD yields a formulation for SSD.

In Section 4.2 we review some basic results about stochastic dominance and present previous formulations for FSD and SSD. In Section 4.3 we present our new formulation for SSD and in Section 4.4 we present our new formulation for FSD. In Section 4.5 we present a specialized branching scheme and some heuristics for solving our new formulation of FSD. In Section 4.6 we present some illustrative computational results, and we close with some concluding remarks in Section 4.7.

4.2 Previous Results on Stochastic Dominance

For the purpose of developing formulations for FSD and SSD, it will be sufficient to present conditions which characterize when a random variable W stochastically dominates the reference random variable Y. We will assume the distributions of W and Y are finite and described by

$$\mu\{W = w_i\} = p_i \qquad i \in \mathcal{N} := \{1, \ldots, N\} \tag{128}$$

$$\nu\{Y = y_k\} = q_k \qquad k \in \mathcal{D} := \{1, \ldots, D\} \tag{129}$$

where μ and ν are the probability distributions induced by W and Y respectively. Furthermore, we assume without loss of generality that $y_1 < y_2 < \cdots < y_D$.

Given a formulation which guarantees W stochastically dominates Y, a formulation for FSD or SSD can be obtained by simply enforcing that $g(x,\xi) = W$. Then, if ξ has distribution given by $\mathbb{P}\{\xi = \xi^i\} = p_i$ for $i \in \mathcal{N}$ and we add the constraints

$$u_i = g(x, \xi^i) \qquad i \in \mathcal{N} \tag{130}$$

to the formulation, we will have $g(x,\xi) \succeq_{(1)} Y$ if and only if $W \succeq_{(1)} Y$ and $g(x,\xi) \succeq_{(2)} Y$ if and only if $W \succeq_{(2)} Y$. Henceforth, we will only consider formulations which guarantee stochastic dominance of W over Y, but based on the relation (130), the reader should think of the values u_i as decision variables, whereas the values y_k are fixed.

When the reference random variable Y has finite distribution, the conditions for stochastic dominance can be simplified, as has been observed, for example, in [22, 23]. We let $y_0 \in \mathbf{R}$ be such that $y_0 < y_1$ and introduce the notation $(\cdot)^+ = \max\{0, \cdot\}$.

Lemma 39. *Let W, Y be random variables with distributions given by* (128) *and* (129). *Then, $W \succeq_{(2)} Y$ if and only if*

$$\sum_{i=1}^{N} p_i (y_k - w_i)^+ \leq \sum_{j=1}^{D} q_j (y_k - y_j)^+ \qquad k \in \mathcal{D} \tag{131}$$

and $W \succeq_{(1)} Y$ if and only if

$$P\{W < y_k\} \leq P\{Y \leq y_{k-1}\} \qquad k \in \mathcal{D}. \tag{132}$$

The key simplification is that the infinite sets of inequalities in the definitions (126) and (127) can be reduced to a finite set when Y has a finite distribution.

Second order stochastic dominance constraints are known to define a convex feasible region [22]. In fact, condition (131) can be used to derive a linear formulation (in an extended variable space) for second order stochastic dominance by introducing variables s_{ik} representing the terms $(y_k - w_i)^+$ [22]. Thus, $W \succeq_{(2)} Y$ if and only if there exists $s \in \mathbf{R}_+^{ND}$ such that

$$\begin{aligned} &\sum_{i=1}^{N} p_i s_{ik} \leq \sum_{j=1}^{D} q_j (y_k - y_j)^+ \quad && k \in \mathcal{D} \\ &s_{ik} + w_i \geq y_k && i \in \mathcal{N}, k \in \mathcal{D}. \end{aligned}$$

We refer to this formulation as SDLP. Note that this formulation introduces ND variables and $(N+1)D$ constraints.

It is possible to use the nonsmooth convex constraints (131) directly, yielding a formulation for SSD that does not introduce auxiliary variables and has $O(D)$ constraints, and specialized methods can be used to solve this formulation, see [23]. The advantage of using a linear formulation is that it can be solved directly by readily available linear programming solvers such as the open source solver CLP [26] or the commercial solver Ilog CPLEX [40].

Condition (132) can be used to derive a MIP formulation for first order stochastic dominance [53, 54]. $W \succeq_{(1)} Y$ if and only if there exists β such that

$$\sum_{i=1}^{N} p_i \beta_{ik} \leq \sum_{j=1}^{k-1} q_j \quad k \in \mathcal{D} \tag{133}$$

$$w_i + M_{ik}\beta_{ik} \geq y_k \qquad i \in \mathcal{N}, k \in \mathcal{D} \tag{134}$$

$$\beta_{ik} \in \{0,1\} \quad i \in \mathcal{N}, k \in \mathcal{D}.$$

We refer to this formulation as FDMIP. Here, M_{ik} is sufficiently large to guarantee that if $\beta_{ik} = 1$, then the corresponding constraint (134) will be redundant. For example, if other constraints in the model imply $w_i \geq l_i$, then we can take $M_{ik} = y_k - l_i$ for all i and k. Although this formulation was presented in [53, 54], the authors do not recommend using this formulation for computation, since the linear programming relaxation bounds are too weak. Instead, because first order stochastic dominance implies second order dominance, any formulation for second order dominance is a relaxation of first order dominance, and the authors therefore propose to use the problem SSD as a relaxation for FSD. Thus, they propose to use the formulation SDLP to obtain bounds for FSD, and to improve these bounds using disjunctive cuts [5]. In addition, formulation SDLP is used as a basis for heuristics to find feasible solutions for FSD. It is demonstrated in [54] that the bounds from using SSD as a relaxation of FSD are usually good, and that using their heuristics based on formulation SDLP, they are able to obtain good feasible solutions. However, these results do not yield a convergent algorithm for finding an optimal solution to FSD, and require the solution of SDLP, which we shall see in our computational results can be quite time-consuming using commercial solvers.

We also want to mention the recent work in [29, 30] in which first order and second order stochastic dominance constraints induced by mixed-integer recourse models are considered. In this two-stage model, an initial set of decisions is made, then the random outcome is observed, and a set of mixed-integer recourse decisions can then be made at minimum cost. In their model, a stochastic dominance constraint is enforced on the (random) recourse cost. The formulations they present for the stochastic dominance constraints are based on the formulations FDMIP and SDLP, respectively. It is clear that our more compact and tighter formulations for FSD and SSD can also be applied in this setting, and we expect that doing so would yield improved computational performance. The same authors have applied their approach to risk modeling in power systems with dispersed generation [28].

4.3 A New Formulation for Second Order Stochastic Dominance

We motivate our new formulation for second order stochastic dominance by first considering the case in which $N = D$ and all outcomes are equally likely, that is, $p_i = q_i = 1/N$ for all $i \in \mathcal{N}$. In this case, Dentcheva and Ruszczyński have shown in [24] that the second order stochastic dominance constraint yields the convexification of the first order stochastic dominance constraint. The proof of this result implicitly yields an alternative formulation for second order stochastic dominance which we now describe. This formulation has also been presented in [42] as a test for second order stochastic dominance, but it is clear that it can also be used to implement stochastic dominance constraints in this case. Note that when all outcomes are equally likely and $N = D$, $W \succeq_{(2)} Y$ if and only if

$$\sum_{i=1}^{N} (y_k - w_i)^+ \leq \sum_{i=1}^{N} (y_k - y_i)^+ \qquad k \in \mathcal{N}.$$

Then, by a theorem in [47], it follows that $W \succeq_{(2)} Y$ if and only if w *majorizes* y, that is,

$$\sum_{i=1}^{k} w_{[i]} \geq \sum_{i=1}^{k} y_i \qquad k \in \mathcal{N} \tag{135}$$

where $w_{[i]}$ represents the i^{th} smallest value in the vector w. Now, a theorem of Hardy, Littlewood and Pólya [33] yields that (135) holds if and only if there exists $\pi \in \mathbf{R}_+^{N^2}$ such that

$$\sum_{j=1}^{N} \pi_{ij} y_j \leq w_i \; i \in \mathcal{N}, \quad \sum_{j=1}^{N} \pi_{ij} = 1 \; i \in \mathcal{N}, \quad \text{and} \quad \sum_{i=1}^{N} \pi_{ij} = 1 \; j \in \mathcal{N}. \tag{136}$$

Hence, this yields a formulation for second order dominance which uses $N^2 = ND$ auxiliary variables, but only $3N$ linear constraints. Our aim is to present a formulation for second order dominance between finitely distributed random variables which has general probabilities and allows $N \neq D$. The formulation will not be based on the majorization theory, and instead will follow from

Lemma 40. *Let $y_1 \leq y_2 \leq \cdots \leq y_D$ and suppose $w \in \mathbf{R}$ and $\pi \in \mathbf{R}^D_+$ satisfy*

$$w - \sum_{j=1}^{D} y_j \pi_j \geq 0 \tag{137}$$

$$\sum_{j=1}^{D} \pi_j = 1. \tag{138}$$

Then,

$$(y_k - w)^+ \leq \sum_{j=1}^{k-1} (y_k - y_j)\pi_j \qquad k \in \mathcal{D}.$$

Moreover, if $w \geq y_1$, then there exists $\pi \in \mathbf{R}^D_+$ which satisfies (137), (138) *and*

$$(y_k - w)^+ = \sum_{j=1}^{k-1} (y_k - y_j)\pi_j \qquad k \in \mathcal{D}. \tag{139}$$

Proof. First suppose $w \in \mathbf{R}$ and $\pi \in \mathbf{R}^D_+$ satisfy (137) and (138). Then, for each $k \in \mathcal{D}$

$$\begin{aligned}
(y_k - w)^+ &\leq (y_k - \sum_{j=1}^{D} y_j \pi_j)^+ && \text{by (137)} \\
&= \Big(\sum_{j=1}^{D} (y_k - y_j)\pi_j\Big)^+ && \text{by (138)} \\
&\leq \sum_{j=1}^{D} (y_k - y_j)^+ \pi_j = \sum_{j=1}^{k-1} (y_k - y_j)\pi_j && \text{since } \pi \geq 0.
\end{aligned}$$

Now suppose $w \geq y_1$. If $w \geq y_D$, then π given by $\pi_D = 1$ and $\pi_j = 0$ for $j = 1, \ldots, D-1$ satisfies (137) - (139). If $w < y_D$, let i be the index such that $y_i \leq w < y_{i+1}$ and note that $1 \leq i < D$. Let π be given by $\pi_j = 0$ for $j < i$ and for $j > i+1$, and $\pi_i = (y_{i+1} - w)/(y_{i+1} - y_i)$ and $\pi_{i+1} = (w - y_i)/(y_{i+1} - y_i)$. Then, $\pi \geq 0$ and π satisfies (138). In addition,

$$\sum_{j=1}^{D} \pi_j y_j = \frac{1}{y_{i+1} - y_i}\big(y_i(y_{i+1} - w) + y_{i+1}(w - y_i)\big) = w$$

and so (137) is satisfied. Finally, for $k \leq i$, $\sum_{j=1}^{k-1}(y_k - y_j)\pi_j = 0 = (y_k - w)^+$ and for $k = i+1$,

$$\sum_{j=1}^{i}(y_{i+1} - y_j)\pi_j = (y_{i+1} - y_i)\left(\frac{y_{i+1} - w}{y_{i+1} - y_i}\right) = (y_{i+1} - w)^+$$

and for $k > i+1$,

$$\sum_{j=1}^{k-1}(y_{i+1} - y_j)\pi_j = (y_k - y_i)\left(\frac{y_{i+1} - w}{y_{i+1} - y_i}\right) + (y_k - y_{i+1})\left(\frac{w - y_i}{y_{i+1} - y_i}\right) = (y_k - w)^+$$

and so (139) holds. □

The standard way to represent the quantities $(y_k - w)^+$ in an optimization problem where w is a decision variable to be maximized is to introduce variables s_k for $k \in \mathcal{D}$ and enforce $s_k \geq 0$ and $s_k \geq y_k - w$ for $k \in \mathcal{D}$. This is how the formulation for second order stochastic dominance SDLP is obtained from the condition (131). Lemma 40 yields an alternative means to represent the quantities $(y_k - w)^+$ which also introduces D variables, but introduces only two constraints. We take advantage of this representation to achieve the following compact formulation for SSD.

Theorem 41. *Let W, Y be random variables with distributions given by* (128) *and* (129). *Then $W \succeq_{(2)} Y$ if and only if there exists $\pi \in \mathbf{R}_+^{ND}$ such that*

$$\sum_{j=1}^{D} y_j \pi_{ij} \leq w_i \qquad i \in \mathcal{N} \tag{140}$$

$$\sum_{j=1}^{D} \pi_{ij} = 1 \qquad i \in \mathcal{N} \tag{141}$$

$$\sum_{i=1}^{N} p_i \sum_{j=1}^{k-1}(y_k - y_j)\pi_{ij} \leq \sum_{j=1}^{k-1} q_j(y_k - y_j) \qquad k \in \mathcal{D}. \tag{142}$$

Proof. From Lemma 39, we know that $W \succeq_{(2)} Y$ if and only if (131) holds. Suppose there exists $\pi \in \mathbf{R}_+^{ND}$ such that (140) - (142) hold. Then, by applying Lemma 40 to w_i and $\pi_i = (\pi_{i1}, \ldots, \pi_{iD})$ for each $i \in \mathcal{N}$, we obtain

$$\sum_{i=1}^{N} p_i (y_k - w_i)^+ \leq \sum_{i=1}^{N} p_i \sum_{j=1}^{k-1}(y_k - y_j)\pi_{ij} \leq \sum_{j=1}^{k-1} q_j(y_k - y_j) = \sum_{j=1}^{D} q_j (y_k - y_j)^+$$

by (142) implying (131) holds and hence $W \succeq_{(2)} Y$.

Now suppose $W \succeq_{(2)} Y$ so that (131) holds. Note that (131) for $k = 1$ implies that $w_i \geq y_1$ for each $i \in \mathcal{N}$. Thus, we apply Lemma 40 for each w_i to obtain a vector $\pi_i \in \mathbf{R}^D_+$ such that (140) and (141) hold and

$$(y_k - w_i)^+ = \sum_{j=1}^{k-1} (y_k - y_j)\pi_{ij} \qquad k \in \mathcal{D}.$$

Hence, (131) implies

$$\sum_{i=1}^{N} p_i \sum_{j=1}^{k-1} (y_k - y_j)\pi_{ij} = \sum_{i=1}^{N} p_i (y_k - w_i)^+ \leq \sum_{j=1}^{k-1} q_j (y_k - y_j).$$

□

To use Theorem 41 to obtain a formulation for SSD, we replace w_i with $g(x, \xi^i)$ so that (140) becomes

$$g(x, \xi^i) \geq \sum_{j=1}^{D} y_j \pi_{ij} \qquad i \in \mathcal{N} \tag{143}$$

and thus obtain the formulation given by

$$f^*_{\text{SSD}} = \max_{x,\pi} \left\{ f(x) : \ (141), (142), (143), x \in X, \pi \in \mathbf{R}^{ND}_+ \right\}.$$

We refer to this formulation as cSSD. This formulation introduces ND variables and $O(N + D)$ rows.

For computational purposes, it is beneficial to introduce variables $v \in \mathbf{R}^D$ and replace the D constraints in (142) with the $2D$ constraints

$$v_j - \sum_{i=1}^{N} p_i \pi_{ij} = 0 \qquad j \in \mathcal{D} \tag{144}$$

$$\sum_{j=1}^{k-1} (y_k - y_j) v_j \leq \sum_{j=1}^{k-1} (y_k - y_j) p_j \qquad k \in \mathcal{D}. \tag{145}$$

This equivalent formulation has D additional variables and constraints, although the number of constraints importantly remains $O(N + D)$. The advantage of this version is that (144) and (145) have $O(ND)$ nonzeros, as compared to $O(ND^2)$ nonzeros in (142).

4.4 A New Formulation for First Order Stochastic Dominance

As in the case for second order stochastic dominance, if $N = D$ and all outcomes are equally likely, a formulation for first order stochastic dominance which introduces N^2 (binary) variables and $O(N)$ constraints has been presented in [42]. Once again, we are able to generalize this to the case with general probabilities and $N \neq D$. An immediate advantage of this formulation is that it is compact, and hence the linear programming relaxation can be solved efficiently. In addition, we will see that the bound from this relaxation is exactly the same as that from the second order stochastic dominance relaxation proposed in [53].

Theorem 42. *Let W, Y be random variables with distributions given by* (128) *and* (129). *Then $W \succeq_{(1)} Y$ if and only if there exists $\pi \in \{0,1\}^{ND}$ such that (w, π) satisfy* (140),(141) *and*

$$\sum_{i=1}^{N} p_i \sum_{j=1}^{k-1} \pi_{ij} \leq \sum_{j=1}^{k-1} q_j \qquad k = 2, \ldots, D. \tag{146}$$

Proof. Let us introduce the notation $\Omega_k := \{i \in \mathcal{N} : w_i < y_k\}$ for $k \in \mathcal{D}$. Now, suppose $W \succeq_{(1)} Y$. Then, by condition (132) in Lemma 39 we have

$$\mu(\Omega_k) = \mu\{W < y_k\} \leq \nu\{Y \leq y_{k-1}\} = \sum_{i=1}^{k-1} q_i. \tag{147}$$

Next, for $i \in N$, define $k^*(i) := \max\{k \in \mathcal{D} : i \notin \Omega_k\}$. Note that $k^*(i)$ always exists since (132) for $k = 1$ implies $\mu\{W < y_1\} = 0$, and so for sure $i \notin \Omega_1$ for each i. Next, note that $k^*(i) < k$ if and only if $i \in \Omega_k$. That $k^*(i) < k$ implies $i \in \Omega_k$ is immediate by the definition of $k^*(i)$. On the other hand, if $i \in \Omega_k$, then also $i \in \Omega_j$ for all $j > k$, and so $k^*(i) < k$. Now, for each $i \in \mathcal{N}$, let $\pi_{ij} = 1$ for $j = k^*(i)$ and $\pi_{ij} = 0$ for all $j \neq k^*(i)$. By definition, we have that π satisfies (141). For $k > 1$, $\sum_{j=1}^{k-1} \pi_{ij} = 1$ if and only if $k^*(i) < k$ and so also $\sum_{j=1}^{k-1} \pi_{ij} = 1$ if and only if $i \in \Omega_k$. Hence,

$$\mu(\Omega_k) = \sum_{i=1}^{N} p_i \sum_{j=1}^{k-1} \pi_{ij}. \tag{148}$$

Combining (148) with (147) shows that π satisfies (146), since also any π which satisfies (141) trivially satisfies (146) for $k = D$. Finally, for each $i \in \mathcal{N}$, we have $\sum_{j=1}^{D} y_j \pi_{ij} = y_{k^*(i)} \leq w_i$ since, by definition, $i \notin \Omega_{k^*(i)}$, thus establishing that π satisfies (140).

Now suppose $\pi \in \{0,1\}^{ND}$ satisfies (140),(141) and (146). For $k \in D$, let $\Omega'_k := \{i \in \mathcal{N} : \sum_{j=1}^{k-1} \pi_{ij} = 1\}$. Note that if $i \notin \Omega'_k$ then $\sum_{j=1}^{k-1} \pi_{ij} = 0$ and by (141) $\sum_{j=k}^{D} \pi_{ij} = 1$, and so $\sum_{j=1}^{D} y_j \pi_{ij} \geq y_k$. Using (140) we have that if $i \in \Omega_k$, then

$$\sum_{j=1}^{D} y_j \pi_{ij} \leq w_i < y_k$$

and hence also $i \in \Omega'_k$. Therefore, by (146), we have for $k \in D$,

$$\mu\{W < y_k\} = \mu(\Omega_k) \leq \mu(\Omega'_k) = \sum_{i=1}^{N} \sum_{j=1}^{k-1} \pi_{ij} \leq \sum_{j=1}^{k-1} q_j = \nu\{Y \leq y_{k-1}\}$$

and so $W \succeq_{(1)} Y$ by condition (132) in Lemma 39. □

Using Theorem 42 with $w_i = g(x, \xi^i)$, we obtain the formulation for FSD given by

$$f^*_{\text{FSD}} = \max_{x,\pi} \left\{ f(x) : \ (141), (143), (146), x \in X, \pi \in \{0,1\}^{ND} \right\}.$$

We refer to this formulation as cFSD. As in the new formulation for second order stochastic dominance, for computational purposes it is beneficial to use the equivalent formulation obtained by introducing variables $v \in \mathbf{R}^D$ and replacing the constraints (146) with the constraints (144) and

$$\sum_{j=1}^{k-1} v_j \leq \sum_{j=1}^{k-1} q_j \qquad k \in \mathcal{D}. \tag{149}$$

One advantage of formulation cFSD over FDMIP is the number of constraints is reduced from $O(ND)$ to $O(N + D)$, which means it should be more efficient to solve the linear programming relaxation of cFSD than to solve that of FDMIP. We now consider the relationship between the relaxation of this formulation and *second* order stochastic dominance.

Lemma 43. *Let W, Y be random variables with distributions given by* (128) *and* (129)*, and suppose there exists $\pi \in \mathbf{R}^{ND}_+$ such that (w, π) satisfy* (140),(141) *and* (146)*. Then, $W \succeq_{(2)} Y$.*

Proof. Let $\pi \in \mathbf{R}_+^{ND}$ and (w, π) satisfy (140),(141) and (146). Then,

$$\begin{aligned}
\sum_{i=1}^{N} p_i \sum_{j=1}^{k-1} \pi_{ij}(y_k - y_j) &= \sum_{i=1}^{N} p_i \sum_{j=1}^{k-1} \pi_{ij} \sum_{l=j+1}^{k} (y_l - y_{l-1}) \\
&= \sum_{l=2}^{k} (y_l - y_{l-1}) \sum_{i=1}^{N} p_i \sum_{j=1}^{l-1} \pi_{ij} \\
&\leq \sum_{l=1}^{k} (y_l - y_{l-1}) \sum_{j=1}^{l-1} q_j \qquad \text{by (146)} \\
&= \sum_{j=1}^{k-1} q_j (y_k - y_j).
\end{aligned}$$

and hence π also satisfies (142) which implies $W \succeq_{(2)} Y$ by Theorem 41. □

As a result, we know that the linear programming relaxation of cFSD yields a bound at least as strong as the bound obtained from the second order stochastic dominance relaxation. This leads to the question as to whether the bound from cFSD could be strictly better. The following theorem shows that this is not the case.

Theorem 44. *Let W, Y be random variables with distributions given by* (128) *and* (129)*. Then the linear programming relaxation of* cFSD *yields a valid formulation for second order stochastic dominance. That is, $W \succeq_{(2)} Y$ if and only if there exists $\pi \in \mathbf{R}_+^{ND}$ such that (w, π) satisfy* (140),(141) *and* (146)*.*

Proof. Due to Lemma 43, we need only show the "only if" part, so suppose $W \succeq_{(2)} Y$. By Theorem 41, there exists $\pi \in \mathbf{R}_+^{ND}$ such that (140), (141) and (142) are satisfied. We show that this implies existence of $\pi' \in \mathbf{R}_+^{ND}$ such that (140) and (141) are satisfied, and (142) is satisfied at equality, that is

$$\sum_{i=1}^{N} p_i \sum_{j=1}^{k-1} (y_k - y_j)\pi'_{ij} = \sum_{j=1}^{k-1} q_j (y_k - y_j) \tag{150}$$

for $k \in \mathcal{D}$. Suppose no such π' exists. Then, let $l \geq 2$ be the maximum index such that there exists a solution $\pi' \in \mathbf{R}_+^{ND}$ which satisfies (140), (141), (142) and (150) for $k = 2, \ldots, l-1$ (if $l = 2$ then $\pi' = \pi$ works so such an $l \geq 2$ exists). Next, let $\pi' \in \mathbf{R}_+^{ND}$ be a solution which satisfies (140), (141), (142) and (150) for $k = 2, \ldots, l-1$ and which minimizes $\sum_{i=1}^{N} \sum_{k=l}^{D} \pi_{ik}$. We claim that $\sum_{i=1}^{N} \sum_{k=l}^{D} \pi'_{ik} > 0$. Suppose not. Then (141)

implies $\sum_{j=1}^{l-1} \pi'_{ij} = 1$ for all $i \in \mathcal{N}$. In addition, by definition of l, (150) holds for $k = l-1$. Using this equality satisfied by π' we obtain

$$\begin{aligned}
\sum_{i=1}^{N} p_i \sum_{j=1}^{l-1} (y_l - y_j)\pi'_{ij} &= \sum_{i=1}^{N} p_i \sum_{j=1}^{l-1} (y_l - y_{l-1})\pi'_{ij} + \sum_{i=1}^{N} p_i \sum_{j=1}^{l-2} (y_{l-1} - y_j)\pi'_{ij} \\
&= (y_l - y_{l-1}) \sum_{i=1}^{N} p_i \sum_{j=1}^{l-1} \pi'_{ij} + \sum_{j=1}^{l-2} q_j (y_{l-1} - y_j) \\
&= (y_l - y_{l-1}) \sum_{i=1}^{N} p_i + \sum_{j=1}^{l-2} q_j (y_{l-1} - y_j) \quad \Big(\text{since } \sum_{j=1}^{l-1} \pi'_{ij} = 1\Big) \\
&\geq \sum_{j=1}^{l-1} q_j (y_l - y_{l-1}) + \sum_{j=1}^{l-2} q_j (y_{l-1} - y_j) \\
&= \sum_{i=1}^{l-1} p_i (y_l - y_j)
\end{aligned}$$

contradicting the definition of l because this would imply π' satisfies (150) for $k = l$. Hence, $\sum_{i=1}^{N} \sum_{k=l}^{D} \pi'_{ik} > 0$ and so there exists i' and $k' \geq l$ such that $\pi'_{i'k'} > 0$. Next, let

$$\delta_1 = \max\{\delta : \sum_{i=1}^{N} p_i \sum_{j=1}^{l-1} (y_l - y_j)\pi'_{ij} + p_{i'}(y_l - y_{l-1})\delta \leq \sum_{j=1}^{l-1} q_j (y_l - y_j)\}$$

and $\delta' = \min\{\delta_1, \pi'_{i'k'}\}$. Then $\delta' > 0$. Now, let $\hat{\pi}$ be given by $\hat{\pi}_{ij} = \pi'_{ij}$ for all $i \neq i'$ and all $j \in \mathcal{D}$, $\hat{\pi}_{i'j} = \pi'_{i'j}$ for all $j \notin \{l, k'\}$, $\hat{\pi}_{i'l} = \pi'_{i'l} + \delta'$ and $\hat{\pi}_{i'k'} = \pi'_{i'k'} - \delta' \geq 0$. Then, $\hat{\pi} \geq 0$ and it is easy to see that $\hat{\pi}$ satisfies (140),(141) and (142) since π' satisfies them. In addition $\hat{\pi}$ satisfies (150) for $k = 1, \ldots, l-1$. Finally, we obtain either $\hat{\pi}$ satisfies (150) for $k = l$, contradicting the definition of l, or $\sum_{i=1}^{N} \sum_{k=l}^{D} \hat{\pi}_{ik} < \sum_{i=1}^{N} \sum_{k=l}^{D} \pi'_{ik}$, contradicting the definition of π'. Thus, we have proved our claim that there exists $\pi' \in \mathbf{R}_+^{ND}$ which satisfies (140), (141) and (150) for all $k \in \mathcal{D}$. The proof is completed by observing that if (150) is satisfied for $k \in \mathcal{D}$, then also (146) is satisfied at equality for all k. Indeed, the case $k = 1$ is trivial, and (150) for $k = 2$ yields

$$\sum_{i=1}^{N} p_i (y_2 - y_1)\pi'_{i1} = q_1 (y_2 - y_1)$$

and hence $\sum_{i=1}^{N} p_i \pi'_{ij} = q_1$ since $y_2 > y_1$. If $k > 2$, subtracting (150) for $k-1$ from (150) for k yields

$$\sum_{i=1}^{N} p_i (y_k - y_{k-1}) \sum_{j=1}^{k-1} \pi'_{ij} = \sum_{j=1}^{k-1} q_j (y_k - y_{k-1})$$

and since $y_k > y_{k-1}$ we can divide $y_k - y_{k-1}$ to obtain (146) holds with equality for k as well. □

Next, we illustrate the relationship between the formulation cFSD and FDMIP by presenting a derivation of cFSD based on strengthening FDMIP. In FDMIP, if $\beta_{ik} = 0$, then $w_i \geq y_k$. But, because $y_k > y_{k-1} > \cdots > y_1$, then we also know $w_i \geq y_{k-1} > \cdots > y_1$. Thus, we lose nothing by setting $\beta_{i,k-1} = \cdots = \beta_{i1} = 0$. Hence, we can add the inequalities

$$\beta_{ik} \leq \beta_{i,k+1} \qquad i \in \mathcal{N}, k \in \mathcal{D} \tag{151}$$

and maintain a valid formulation. The inequalities (134) can then be replaced by

$$w_i - \sum_{k=1}^{D} (\beta_{i,k+1} - \beta_{ik}) y_k \geq 0 \qquad i \in \mathcal{N}$$

which together with inequalities (151) ensure that when $\beta_{ik} = 0$, we have $w_i \geq y_k$. We finally obtain the new formulation cFSD by substituting $\pi_{ik} = \beta_{i,k+1} - \beta_{ik}$ for $k \in \mathcal{D}$ and $i \in \mathcal{N}$, where $\beta_{i,D+1} = 1$.

4.5 Branching and Heuristics for FSD

cFSD yields a mixed-integer programming formulation for FSD. Moreover, if X is a polyhedron and $g(x, \xi^i)$ are affine in x for each i, cFSD is a mixed-integer *linear* programming formulation, which can be solved by commercially available solvers such as Ilog CPLEX, which are based on the branch-and-cut methodology. As has been shown in [54], the optimal value of SSD yields a good bound on the optimal value of FSD, and hence the bound obtained from relaxing integrality in cFSD should be good. In addition, because of the compactness of cFSD, this bound can be calculated efficiently. However, we have found that the default settings in Ilog CPLEX do not effectively generate good feasible solutions for cFSD. In addition, the default branching setting does not help to find feasible solutions or effectively improve the relaxation bounds. In this section we show that by exploiting the special structure of this formulation, it is possible to develop specialized branching routines and heuristics which yield improved computational performance.

Throughout this section, we will make use of the problem given by

$$f^*_{\text{cSSD}(l,u)} = \max_{x,\pi} \{f(x) : (141), (142), (143), x \in X, l \leq \pi \leq u\} \qquad (\text{cSSD}(l,u))$$

which is the relaxation of cFSD with the addition of lower and upper bounds $l, u \in \{0,1\}^{ND}$ on the π variables. For example, the root relaxation of cFSD, cSSD, is obtained by setting $l_{ij} = 0$ and $u_{ij} = 1$ for all $i \in \mathcal{N}, j \in \mathcal{D}$. More generally, l and u may be the bounds on π at a node in the branch-and-bound tree obtained by branching, or bounds which are enforced during a heuristic.

4.5.1 Branching for FSD

Standard variable branching for mixed-integer programming would select a fractional variable π_{ij}, and then create two branches, one with π_{ij} fixed to one and one with π_{ij} fixed to zero. However, the constraints (141) and (140) imply that for a fixed i, the set of variables π_{ij} for $j \in \mathcal{D}$ are essentially selecting which value level y_j the variable w_i should be greater than. In particular, the set of variables $\{\pi_{ij} : j \in \mathcal{D}\}$ is a *Special Order Set*, that is, at most one of the variables in this set can be positive. As a result, it is natural to consider using a Special Order Set branching rule (see, e.g. [9]). In this branching scheme, we select a *set index* $i \in \mathcal{N}$, specifying which Special Ordered Set to branch on, and also choose a *level index* $k \in \{2, \ldots, D\}$. Then in the first branch the constraint $\sum_{j<k} \pi_{ij} = 0$ is enforced and in the second branch $\sum_{j<k} \pi_{ij} = 1$ is enforced. In an implementation, the first condition is enforced by changing the upper bound on the variables π_{ij} to zero for $j < k$, and the second condition is enforced by changing the upper bound on the variables π_{ij} to zero for $j \geq k$.

To specify a Special Ordered Set branching rule, we must state how the set and level indices are chosen. Our branching scheme is based on attempting to enforce the feasibility condition (132)

$$\mu\{W < y_k\} \leq \nu\{Y \leq y_{k-1}\} \qquad k \in \mathcal{D}.$$

At each node in which we must branch, we find $k^* = \min\big\{k \in \mathcal{D} : \{W < y_k\} > \nu\{Y \leq y_{k-1}\}\big\}$ based on the values of w in the current relaxation solution. Note that if such a k^* does not exist, then we have $W \succeq_{(1)} Y$ so the current solution is feasible. In this case, if π is not integer feasible (which may happen), we construct an integer feasible solution of the same cost as in the proof of Theorem 42, and as a result, branching is not

required at this node. We will take k^* to be the level index which we will branch on. Note that (140) and (141) imply that $w_i \geq y_1$ for all i in any relaxation solution, so that $k^* \geq 2$, making it an eligible branching level index.

We next choose a set index $i \in \mathcal{N}$ such that

$$w_i < y_{k^*} \tag{152}$$

$$\sum_{j<k^*} \pi_{ij} < 1. \tag{153}$$

We claim that such an index must exist. Indeed, let $\Omega_{k^*} = \{i \in \mathcal{N} : w_i < y_{k^*}\}$. By the definition of k^* we have $\sum_{i\in\Omega_{k^*}} p_i > \sum_{j=1}^{k-1} q_j$ and so, in particular, $\Omega_{k^*} \neq \emptyset$. If there were no $i \in \Omega_{k^*}$ which also satisfies (153), then we would have

$$\sum_{i=1}^{N} p_i \sum_{j=1}^{k^*-1} \pi_{ij} \geq \sum_{i\in\Omega_{k^*}} p_i > \sum_{j=1}^{k^*-1} q_j$$

violating (142). If there are multiple set indices which satisfy (152) and (153), we choose the index which maximizes the product $(y_{k^*} - w_i)(1 - \sum_{j<k^*} \pi_{ij})$. In the first branch, we enforce $\sum_{j<k^*} \pi_{ij} = 0$ which by (140) forces $w_i \geq y_{k^*}$. Because of (152), this will make the current relaxation solution infeasible to this branch, and will promote feasibility of (132) at the currently infeasible level k^*. In the second branch, we enforce $\sum_{j<k^*} \pi_{ij} = 1$ and because of (153) this will make the current relaxation solution infeasible for this branch. The motivation for this choice of set index i is to make progress in both of the branches. The motivation for the choice of level index k^* is that in the first branch progress towards feasibility of (132) is made, whereas by selecting k^* small, reasonable progress is also made in the second branch since this enforces $\pi_{ij} = 0$ for all $j \geq k^*$.

4.5.2 Heuristics for FSD

We now present several heuristics we have developed that can be used with formulation cFSD. An important factor in the efficiency of these heuristics, as well as more complex heuristics that may be developed, is the ability to solve the relaxation of cFSD efficiently. Consequently, the compactness of this new formulation should also be beneficial for generating high quality feasible solutions.

We first present a simple and efficient heuristic, called the *order preserving heuristic* and then present two variants of a diving heuristic which are more computationally demanding. We finally discuss how the order preserving heuristic can be integrated into the diving heuristics.

Order preserving heuristic

Given a solution x^* to a relaxation of cFSD, let $w^* \in \mathbf{R}^N$ be the vector given by $w_i^* = g(x^*, \xi^i)$ for $i \in \mathcal{N}$. The idea behind the order preserving heuristic is to use w^* as a guide to build a solution $\hat{\pi} \in \{0,1\}^{ND}$ which satisfies (141) and (142), and then solve cSSD$(\hat{\pi}, \hat{\pi})$. If this problem is feasible, it yields a feasible solution to cFSD. The heuristic is *order preserving* because it chooses $\hat{\pi}$ in such a way that if $w_i^* < w_{i'}^*$, then $\sum_{j \in \mathcal{D}} y_j \hat{\pi}_{ij} \leq \sum_{j \in \mathcal{D}} y_j \hat{\pi}_{i'j}$ so that the constraints (143) obtained with this $\hat{\pi}$ enforce lower bounds on $g(x, \xi^i)$ which are consistent with the ordering of $w_i^* = g(x^*, \xi^i)$ obtained from the current relaxation solution. The order preserving heuristic is given in Algorithm 2. The

Algorithm 2: Order preserving heuristic

Data: $w^* \in \mathbf{R}^N$

1 Sort w^* to obtain $\{i_1, \ldots, i_N\} = \mathcal{N}$ with $w_{i_1}^* \leq w_{i_2}^* \leq \cdots \leq w_{i_N}^*$;
2 Set $t := 1$ and $\hat{\pi}_{ij} := 0$ for all $i \in \mathcal{N}, j \in \mathcal{D}$;
3 **for** $k := 1$ **to** D **do**
4 **while** $t \leq N$ **and** $\sum_{j=1}^{t} p_{i_j} \leq \sum_{j=1}^{k} q_j$ **do**
5 $\hat{\pi}_{i_t k} := 1$;
6 $t := t + 1$;
7 **end**
8 **end**
9 Solve cSSD$(\hat{\pi}, \hat{\pi})$;
10 **if** cSSD$(\hat{\pi}, \hat{\pi})$ *is feasible* **then**
11 **return** $(\hat{x}, \hat{\pi})$, the optimal solution to cSSD$(\hat{\pi}, \hat{\pi})$;
12 **end**

algorithm begins by sorting the values of w^*. Then, lines 2 to 8 construct a solution $\hat{\pi}$ which is feasible to (141) and (142) by working in this order. To see that $\hat{\pi}$ satisfies (141), observe that the algorithm will terminate with $t = N + 1$, since when $k = D$, $\sum_{j=1}^{t} p_{i_j} \leq \sum_{j=1}^{D} q_j$ for all $t \leq N$, so the loop on line 4 will only terminate when $t > N$. Since $\{i_1, \ldots, i_N\} = \mathcal{N}$, this implies that for each $i \in \mathcal{N}$, there is some k such that the algorithm sets $\hat{\pi}_{ik} = 1$. The condition $\sum_{j=1}^{t} p_{i_j} \leq \sum_{j=1}^{k} q_j$ in line 4 ensures that (142) holds for $\hat{\pi}$, since it ensures that

for each $k \in \mathcal{D}$,

$$\sum_{j=1}^{k} \sum_{i=1}^{N} p_i \hat{\pi}_{ij} = \sum_{j=1}^{t(k)} p_{i_j}$$

where $t(k) = \max \left\{ t : \sum_{j=1}^{t} p_{i_j} \leq \sum_{j=1}^{k} q_j \right\}$.

The main work done in Algorithm 2 is the sorting of w^*, and solving of cSSD$(\hat{\pi}, \hat{\pi})$. Note that this problem is small relative to the original problem cFSD, since the $O(ND)$ variables π are fixed, the constraints (141) and (142) no longer need to be considered, and the constraints (143) reduce to lower bounds on the functions $g(x, \xi^i)$ for $i \in \mathcal{N}$.

Aggressive diving heuristic

The aggressive diving heuristic, described in Algorithm 3, is similar to the classic diving heuristic of integer programming in that it involves progressively fixing values of π_{ij} and re-solving the relaxation until either the relaxation solution yields value worse than a current feasible solution value (or is infeasible), or a feasible solution is found. The difference is that at each iteration, between re-solves of the relaxation, the aggressive diving heuristic fixes many values of π_{ij}. In particular, by fixing a variable π_{ik} to one, constraint (141) implies we have actually also fixed π_{ij} to zero for all $j \neq k$. Also, in each iteration, we fix multiple variables to one, in such a way that the fixing causes at least one of the constraints (143) with the *smallest* values of $g(x, \xi^i)$ to become violated, and also at least one of the constraints (143) with the *largest* values of $g(x, \xi^i)$ to become violated. Similar to the order preserving heuristic, Algorithm 3 attempts to build a solution which satisfies (141) and (142), using the values of $g(x^*, \xi^i)$ obtained from solving intermediate relaxations as a guide. The difference is that the order preserving heuristic constructs a solution $\hat{\pi}$ based on a single relaxation solution, whereas Algorithm 3 solves multiple relaxations.

In Algorithm 3, E represents the set of indices i for which the level of the SOS1 set $\{\pi_{ij} : j \in \mathcal{D}\}$ has not yet been fixed. In addition, the algorithm maintains $v_j = \sum_{i \in \mathcal{N}: l_{ij}=1} p_i$ for each $j \in \mathcal{D}$, i.e. v_j represents the sum of probabilities of sets which have been fixed at level j. As such, constraint (142) will remain satisfied as long as

$$\sum_{j=1}^{k} v_j \leq \sum_{j=1}^{k} q_j \quad \text{for all } k \in \mathcal{D}. \tag{154}$$

Algorithm 3: Aggressive diving heuristic

Data: LB, a lower bound on f^*_{FSD}

Set $E := \mathcal{N}$ and $v_j := 0.0$ for all $j \in \mathcal{D}$;
Set $l_{ij} := 0$ and $u_{ij} := 1$ for all $i \in \mathcal{N}, j \in \mathcal{D}$;
while $E \neq \emptyset$ **do**
    Solve cSSD(l, u) ;
    if cSSD(l, u) *is infeasible* **or** $f^*_{\text{cSSD}(l,u)} \leq LB$ **then** terminate ;
    Let (x^*, π^*) be an optimal solution to cSSD(l, u);
    if $\pi^* \in \{0, 1\}^{ND}$ **then** terminate with feasible solution (x^*, π^*);
    Let $w^*_i = g(x^*, \xi^i)$ for $i \in E$;
    Sort w^* to obtain $\left\{i_1, \ldots, i_{|E|}\right\} = E$ with $w^*_{i_1} \leq w^*_{i_2} \leq \cdots \leq w^*_{i_{|E|}}$;
    Set constrains := FALSE and $s := 1$;
    while constrains = FALSE **and** $s \leq |E|$ **do**
        Let $\hat{k} = 1 + \max\left\{k \in \mathcal{D} : \sum_{j=1}^{k} v_j + p_{i_s} > \sum_{j=1}^{k} q_j\right\}$;
        Set $l_{i_s \hat{k}} := 1$;
        Set $v_{\hat{k}} := v_{\hat{k}} + p_{i_s}$;
        if $w^*_{i_s} < y_{\hat{k}}$ **then** Set constrains := TRUE;
        Set $s := s + 1$ and $E = E \setminus \{i_s\}$;
    end
    Set constrains := FALSE and $t := |E|$;
    while constrains = FALSE **and** $t \geq s$ **do**
        Let $\hat{k} = \max\left\{k \in \mathcal{D} : \sum_{j=k}^{D} v_j < \sum_{j=k}^{D} q_j\right\}$;
        Set $l_{i_t \hat{k}} := 1$;
        Set $v_{\hat{k}} := v_{\hat{k}} + p_{i_t}$;
        if $w^*_{i_t} < y_{\hat{k}}$ **then** Set constrains := TRUE;
        Set $t := t - 1$ and $E = E \setminus \{i_t\}$;
    end
end
Solve cSSD(l, u) ;
if cSSD(l, u) *is feasible* **then**
    Let (x^*, π^*) be an optimal solution to cSSD(l, u);
    terminate with feasible solution (x^*, π^*);
end

The algorithm terminates either at line 5, line 7, or after executing lines 27-31 with $E = \emptyset$. If cSSD(l, u) is found to be infeasible when solved in line 4, or the relaxation solution cost exceeds LB, then the algorithm terminates at line 5 without finding an improved feasible solution. If the solution (x^*, π^*) of cSSD(l, u) has π^* integral, then we terminate in line 7 with a feasible solution. In fact, this termination criterion can be relaxed somewhat. If x^* satisfies (132), then we can construct an integer solution π' such that (x^*, π') is feasible to cFSD, and hence terminate even though π^* may not have been integral.

When $E = \emptyset$, the main loop in the algorithm terminates. In this case, in the solution of cSSD(l, u), π is actually fixed to l. We claim that at this point l satisfies (141) and (142). Indeed, for each i, $\sum_{j \in \mathcal{D}} l_{ij} = 1$ since any time an index i is removed from E in lines 16 or 24, this index had $l_{i\hat{k}}$ set to one in lines 13 or 21, and conversely any time an index i has l_{ik} set to one for some k, it is immediately removed from E, so no other $l_{ik'}$ will be set to one. In addition, we verify that (142) holds by verifying that (154) holds throughout the algorithm. First observe that (154) is trivially satisfied at the beginning of the algorithm. The only lines in which the values of v are modified are lines 14 and 22. By the definition of $\hat{k}$ in line 12,

$$\sum_{j=1}^{k} v_j + p_{i_s} \leq \sum_{j=1}^{k} q_j \quad \text{for all } k = \hat{k}, \ldots, D$$

and therefore (154) holds even after $v_{\hat{k}}$ is increased in line 14. Next, observe that the algorithm maintains

$$\sum_{j=1}^{D} v_j + \sum_{i \in E} p_i = 1. \tag{155}$$

Indeed, this is initially true since the algorithm begins with $\sum_{j=1}^{D} v_j = 0$ and $E = \mathcal{N}$. Likewise, anytime a value of v_j is increased by an amount p_i in lines 14 or 22, the index i is subsequently removed from E in lines 16 or 24. Next, by definition of $\hat{k}$ in line 20,

$$\sum_{j=k}^{D} v_j \geq \sum_{j=k}^{D} q_j \quad \text{for all } k = \hat{k} + 1, \ldots, D.$$

Then by (155) and $\sum_{i \in \mathcal{N}} p_i = 1$, this implies that at line 22

$$\sum_{j=1}^{k} v_j + p_{i_t} \leq \sum_{j=1}^{k} v_j + \sum_{i \in E} p_i \leq \sum_{j=1}^{k} p_j \quad \text{for all } k = \hat{k}, \ldots, D$$

since $i_t \in E$. Therefore (154) holds also after $v_{\hat{k}}$ is increased in line 22.

We remark that the fixing in Algorithm 3 could be done exclusively from smallest to largest, as in lines 10 to 17, or exclusively from largest to smallest, as in lines 18 to 25. However, we have observed from computational experience that going from both directions generally yields both better feasible solutions and faster convergence in terms of the number of times cSSD(l, u) has to be solved.

One-at-a-time diving heuristic

We have also considered a variant of the aggressive diving heuristic, which we call the one-at-a-time diving heuristic. The only difference with this variant is that in each iteration we fix at most two variables π_{ij} to one at each iteration: one corresponding to the smallest unfixed level, and one corresponding to the largest unfixed level. Thus, the description of this heuristic is exactly as given in Algorithm 3, with the exception that the while loops in lines 10 - 17 and 18 - 25 are executed at most once, with no check that the fixings being done are constraining. This heuristic fixes fewer variables per iteration, and hence will likely take more iterations to complete than the aggressive diving heuristic. The hope is that this extra time will lead to better feasible solutions.

Integration of order preserving and diving heuristics

In our computational experiments we found that because the order preserving heuristic is very efficient to run, it is beneficial to use it in conjunction with either of the diving heuristics. The idea is simply to call the order preserving heuristic during each iteration of the diving heuristic, using the current relaxation solution. If this yields an improved feasible solution, we keep it, but still continue the dive heuristic until it terminates. At the end, we report the best feasible solution found over all iterations in the dive.

4.6 Computational Results

We conducted computational experiments to test the new formulations for stochastic dominance we have presented. Following [42] and [54], we conducted our tests on a portfolio optimization problem with stochastic dominance constraints. In this problem, we wish to choose the fraction of our investment to invest in n different assets. The return of asset j is

a random variable given by R_j with $\mathbb{E}[R_j] = r_j$. We are also given a reference random variable Y and our objective is to maximize the expected return subject to the constraint that the random return we achieve stochastically dominates Y. Thus, the portfolio optimization problems we consider are

$$\max\Big\{\sum_{j=1}^{n} r_j x_j : \; x \in X, \; \sum_{j=1}^{n} R_j x_j \succeq_{(k)} Y\Big\} \qquad k = 1, 2 \tag{156}$$

where $X = \big\{x \in \mathbf{R}^n_+ : \sum_{j=1}^{n} x_j = 1\big\}$.

We constructed test instances using the daily returns of 435 stocks ($n = 435$) in the S&P 500, for which daily return data was available from January 2002 through March 2007. We take each daily return as an outcome that occurs with equal probability. For each desired number of outcomes N, we constructed three instances by taking the N daily returns immediately preceding March 14 of the years 2005, 2006, and 2007. For example, the instance for year 2007, with $N = 100$ is obtained by taking the daily returns in the days from November 16, 2006 through March 14, 2007.

For the reference random variable Y, we use the returns that would be obtained by investing an equal fraction in each of the available assets. That is, we take $Y = \sum_{j=1}^{n} R_j/n$. Hence, if R^i_j is the return that is achieved under outcome i for asset j, then the distribution of Y is given by $\nu\Big\{Y = \sum_{j=1}^{n} R^i_j/n\Big\} = 1/N$ for $i \in \mathcal{N}$. Note that in this case, the number of outcomes of Y is the same as the number of outcomes of R, i.e., $D = N$. This is an extreme case: in many settings we would expect D to be significantly less than N. However, this extreme case will yield challenging instances for comparing the formulations.

We used CPLEX 9.0 [40] to solve the LP and MIP formulations and all experiments were done on a computer with two 2.4 Ghz processors (although no parallelism is used) and 2.0 Gb of memory. The specialized heuristics and branching for first order stochastic dominance were implemented using callback routines provided by the CPLEX callable library.

4.6.1 Second Order Dominance

We first compared the solution times using the formulations SDLP and cSSD to solve the portfolio optimization problem (156) with second order dominance constraint ($k = 2$ in (156)). We tested eight different sizes N and three instances for each size. These linear

programs were solved using the dual simplex method, the default CPLEX setting, and a time limit of 100,000 seconds was used. Table 25 gives the solution times. From this table

Table 25: Comparison of solution times (s) for SSD

	2005		2006		2007	
N	SDLP	cSSD	SDLP	cSSD	SDLP	cSSD
100	3.3	0.6	2.4	0.3	3.5	0.9
200	102.7	4.6	82.7	3.7	121.7	7.4
300	1063.4	16.9	882.9	12.6	756.6	27.4
400	4858.7	35.9	4252.8	129.8	4291.6	110.7
500	10345.3	42.4	11364.9	89.0	12550.8	185.6
600	27734.3	180.3	43927.4	78.1	27492.3	376.4
700	69485.8	592.4	58946.8	59.9	59144.0	812.1
800	*100122.0	4834.9	*100100.0	91.9	*100095.0	491.5

* Not solved in time limit.

it is clear that when using a commercial LP solver, the new formulation cSSD allows much more efficient solution of SSD, yielding a solution in time two orders of magnitude less than using SDLP directly. Formulation cSSD has $O(N)$ rows as opposed to $O(ND) = O(N^2)$ rows in SDLP, leading to a significantly reduced basis size, so that the time per iteration using formulation cSSD is significantly less. However, we have also observed that significantly fewer iterations are required to solve cSSD, which also contributes substantially to its efficiency.

We should stress that because $N = D$ in this test, the relative improvement of cSSD over SDLP is likely the best case. For instances in which D is of much more modest size, such as $D = 10$, we would not expect such extreme difference between the performance of cSSD and SDLP.

4.6.2 First Order Dominance

We next present results of our tests on the portfolio optimization problem (156) in which a first order stochastic constraint is enforced ($k = 1$ in (156)).

We first present the results from using two variants of the heuristics presented in Section 4.5.2. The two variants correspond to the Aggressive Dive and the One-at-a-time Dive heuristics, both using the Order Preserving heuristic to attempt to find intermediate feasible

solutions at each iteration. We investigated other variants, but found these to yield the best results in terms of time and solution quality. The results for these heuristics are presented in Table 26. The instance with $N = 100$ from 2006 is excluded because the

Table 26: Results for two heuristics.

		Aggressive Dive			One-at-a-time Dive		
Year	N	Time	Its	Gap	Time	Its	Gap
2005	100	1.3	4	0.95%	1.5	28	0.90%
	150	4.5	4	1.60%	10.3	28	0.76%
	200	7.7	6	1.17%	24.2	33	1.09%
	250	19.1	7	0.53%	16.8	48	0.40%
	300	34.4	9	1.44%	206.3	53	1.37%
	350	281.4	7	3.16%	432.8	36	2.03%
	400	114.4	7	0.76%	575.1	41	1.24%
2006	150	10.5	7	1.19%	21.4	41	1.09%
	200	9.2	5	0.79%	7.4	20	0.78%
	250	21.5	4	0.63%	43.2	21	0.51%
	300	34.5	6	1.06%	509.2	24	0.68%
	350	115.0	8	0.94%	142.1	46	0.68%
	400	119.3	7	1.75%	122.6	38	1.74%
2007	100	1.5	3	0.60%	4.0	26	0.17%
	150	10.9	6	4.09%	26.7	51	3.42%
	200	39.0	7	3.14%	68.1	69	2.44%
	250	133.1	9	3.69%	179.7	56	2.94%
	300	225.3	12	2.49%	286.5	59	2.39%
	350	319.8	3	20.82%	2333.0	89	8.21%
	400	759.2	9	5.82%	4785.8	94	4.71%

optimal solution is found by CPLEX default heuristics after solving the root relaxation. For both of the heuristic variants tested we report the total time, the number of iterations, and the optimality gap of the best solution found, relative to the lower bound from the root relaxation (Gap = [Upper Bound - Lower Bound]/Upper Bound). We see that the Aggressive Diving heuristic takes less time, but usually yields worse solutions than the One-at-a-Time Diving heuristic. For the 2005 and 2006 instances, both heuristics yield reasonably good solutions, usually within one percent of the upper bound. For the 2007 instances, however, there remains significant gap between the feasible solutions generated and the upper bound, particularly when using the Aggressive Diving heuristic. Observe that the total time for running these heuristics for yielding solutions to FSD compares very

favorably with the time to solve the formulation SDLP for SSD to simply yield an initial lower bound for FSD.

We tested four solution methods for solving FSD:

- FDMIP: Solve FDMIP with default CPLEX settings,
- cFSD: Solve cFSD with default CPLEX settings and CPLEX SOS1 branching,
- cFSD+H: Solve cFSD with CPLEX SOS1 branching and specialized heuristic, and
- cFSD+H+B: Solve cFSD with CPLEX, specialized heuristic, and specialized branching.

When solving cFSD with and without the heuristic (but not with the specialized branching), we declare the sets of variables $\{\pi_{ij} : j \in \mathcal{D}\}$ for $i \in \mathcal{N}$ as special ordered sets, allowing CPLEX to perform its general purpose special ordered set (SOS1) branching, as discussed in Section 4.5.1. We found that this yields better results than having CPLEX perform its default single variable branching. Note that our specialized branching scheme also uses SOS1 branching, but crucially differs from the CPLEX implementation in the selection of the SOS1 set and level to branch on.

The heuristic used in the last two methods is the Aggressive Diving heuristic. We chose this heuristic because it runs faster, and the disadvantage that it tends to yield lower quality solutions can be offset by calling it more frequently within the branch-and-cut solution method. In our implementation, we call the heuristic at every node of depth less than five, at every fifth node for the first 100 nodes, at every 20th node between 100 and 1000 nodes, and at every 100th node thereafter. When the heuristic is used we turn off the CPLEX heuristics and preprocessing. The preprocessing was turned off for implementation convenience, but we found it had little effect for formulation cFSD anyway.

The specialized branching used in the last method is the branching strategy given in Section 4.5.1. For this case, we set the CPLEX branching variable selection to select the most fractional variable since this takes the least time and we do not use CPLEX's choice of branching variable anyway.

We first compare the time to solve the root linear program relaxations and the resulting lower bound from formulations FDMIP and cFSD. These results are given in Table 27. For formulation FDMIP we report the results before and after addition of CPLEX cuts. The results obtained after the addition of CPLEX cuts are under the column FDMIP.C. For cFSD, we report only the results after the initial relaxation solution, because CPLEX cuts had little effect in this formulation. For formulation FDMIP we report the percent by

Table 27: Comparison of root LP relaxations for FSD formulations.

		Time (s)			Percent above cFSD UB	
Year	N	cFSD	FDMIP	FDMIP.C	FDMIP	FDMIP.C
2005	100	1.0	6.6	41.4	5.36%	3.46%
	150	1.8	19.7	89.5	7.64%	6.18%
	200	4.7	36.3	196.2	8.42%	5.87%
	250	15.1	49.9	365.0	9.34%	6.78%
	300	31.0	232.6	681.5	9.78%	7.50%
	350	88.0	509.7	1201.0	4.36%	3.05%
	400	97.6	427.7	1566.2	5.14%	3.19%
2006	100	0.4	3.9	4.3	0.21%	0.00%
	150	3.8	16.2	82.0	1.54%	1.03%
	200	4.8	26.3	140.9	1.38%	1.08%
	250	17.5	91.1	325.8	3.99%	2.45%
	300	16.4	191.3	575.6	4.60%	3.53%
	350	52.3	227.7	1157.8	8.49%	6.52%
	400	69.1	1254.7	2188.6	6.92%	5.77%
2007	100	2.0	4.5	33.5	7.55%	3.70%
	150	8.1	17.0	148.4	7.69%	6.06%
	200	17.8	33.3	300.8	9.75%	8.26%
	250	36.1	121.4	413.1	14.13%	10.71%
	300	43.5	298.6	732.6	11.12%	8.26%
	350	114.0	320.9	1060.7	10.80%	10.60%
	400	245.7	2010.8	3664.2	11.53%	11.02%

which the upper bound (UB) obtained from the relaxation with and without cuts exceeds the upper bound obtained from the relaxation of cFSD. It is clear from Table 27 that the relaxation of formulation cFSD provides significantly better upper bounds in significantly less time.

We next tested how the different methods performed when run for a time limit of 10000 seconds. Table 28 reports the optimality gap remaining after this time limit. All

formulations were able to solve the 2006 instance with $N = 100$ in less than a minute, so this instance is excluded. Using formulation cFSD with the heuristic and specialized branching, 8 of the remaining 20 instances were solved to optimality within the time limit, and for these instances the solution time is reported. From Table 28 we observe that

Table 28: Comparison of optimality gaps for FSD after time limit.

		Optimality Gap			cFSD+H+B	
Year	N	FDMIP	cFSD	cFSD+H	Gap	Time (s)
2005	100	1.69%	0.68%	0.68%	-	864.0
	150	2.84%	0.99%	0.73%	-	223.1
	200	4.46%	1.09%	0.87%	-	1987.3
	250	8.82%	0.31%	0.24%	-	2106.6
	300	**	3.41%	1.21%	1.15%	
	350	**	**	2.15%	1.39%	
	400	**	10.67%	0.73%	0.31%	
2006	150	1.71%	0.77%	0.55%	0.18%	
	200	1.25%	0.57%	0.55%	-	1752.1
	250	4.82%	0.97%	0.44%	-	274.9
	300	4.56%	4.24%	0.85%	-	9386.8
	350	**	1.96%	0.65%	0.53%	
	400	**	4.77%	1.21%	0.87%	
2007	100	0.13%	0.14%	0.15%	-	41.6
	150	13.90%	4.11%	2.37%	1.85%	
	200	**	3.80%	1.64%	0.67%	
	250	**	9.13%	2.12%	0.67%	
	300	**	**	2.43%	2.01%	
	350	**	**	6.74%	6.37%	
	400	**	**	5.82%	5.79%	

** No feasible solution found.

even without the use of specialized heuristic or branching formulation cFSD outperforms formulation FDMIP. However, in several instances cFSD fails to find a feasible solution, and in several others the optimality gaps for the feasible solutions found are quite bad. This is remedied to a significant extent by using the specialized heuristic, in which case a feasible solution is found for every instance, and in most cases it is within 2% of the upper bound. If, in addition, we use the specialized branching scheme, the final optimality gaps are reduced even further, with many of the instances being solved to optimality.

Table 29 gives more detailed results for the methods based on formulation cFSD. First,

for each of these methods, the table indicates the percent by which the final upper bound (UB) was below the initial upper bound (Root UB) obtained simply from solving the linear programming relaxation. These results indicate that by using CPLEX branching, with and without the specialized heuristic, very little progress is made in improving the upper bound through branching. In contrast, the specialized branching scheme improves the upper bound considerably, with the exception of the two largest instances for 2007. However, for these instances the majority of the time was spent solving the initial relaxation and running the heuristic, and so very few branches were performed. For the instances which were solved to optimality using the specialized heuristics and branching, the percent by which the final upper bound is below the root upper bound represents the gap between the SSD relaxation bound and optimal value of FSD. These results confirm the results of [54] which indicate that the problem SSD is a tight relaxation of FSD. Table 29 also reports the

Table 29: Lower and upper bounds results using cFSD.

		% UB below Root UB			% LB below Best UB		
Year	N	cFSD	+H	+H+B	cFSD	+H	+H+B
2005	100	0.02%	0.02%	0.69%	0.01%	0.01%	0.01%
	150	0.00%	0.00%	0.66%	0.33%	0.07%	0.01%
	200	0.00%	0.00%	0.74%	0.36%	0.14%	0.01%
	250	0.00%	0.00%	0.20%	0.11%	0.04%	0.01%
	300	0.00%	0.00%	0.06%	3.47%	1.17%	1.16%
	350	0.00%	0.00%	0.26%	**	1.93%	1.41%
	400	0.00%	0.00%	0.23%	11.68%	0.50%	0.31%
2006	150	0.04%	0.04%	0.42%	0.38%	0.17%	0.18%
	200	0.00%	0.00%	0.54%	0.04%	0.01%	0.01%
	250	0.00%	0.00%	0.38%	0.60%	0.06%	0.01%
	300	0.00%	0.00%	0.57%	3.84%	0.29%	0.01%
	350	0.00%	0.00%	0.14%	1.86%	0.52%	0.54%
	400	0.00%	0.00%	0.64%	4.34%	0.57%	0.88%
2007	100	0.00%	0.00%	0.14%	0.01%	0.02%	0.01%
	150	0.23%	0.25%	0.79%	3.70%	1.88%	1.88%
	200	0.01%	0.00%	0.70%	3.23%	0.95%	0.67%
	250	0.00%	0.02%	1.20%	8.72%	0.96%	0.68%
	300	0.00%	0.00%	0.45%	**	2.03%	2.05%
	350	0.00%	0.00%	0.00%	**	7.22%	6.80%
	400	0.00%	0.00%	0.03%	**	6.15%	6.15%

** No feasible solution found.

percent by which the value of the best feasible solution found (LB) is below the best upper bound found over all methods (Best UB). Not surprisingly, these results indicate that using the specialized heuristic significantly improves the value of the feasible solutions found. However, these results also indicate that combining the specialized branching with the heuristic usually yields further improved solutions, indicating that the specialized branching scheme is beneficial both for improving the upper bounds and for finding better feasible solutions.

4.7 Concluding Remarks

We have introduced new formulations for optimization under first and second order stochastic dominance constraints, in the case when there are finitely many possible outcomes of the random variables. For the second order dominance constraint, the new formulation is a linear program and has the advantage that it introduces many fewer rows than the previously introduced linear programming formulation. Computational results indicate that for the case in which the number of possible outcomes for the reference random variable is large, the new formulation can be solved orders of magnitude faster than the previous formulation.

For the first order dominance constraint, the new formulation is a mixed-integer program and also has the advantage that it introduces many fewer rows than the previously introduced MIP formulation. In addition, the linear programming relaxation of this formulation is also a formulation for second order stochastic dominance, leading to a tight relaxation bound. Despite these advantages, directly solving this new formulation with a commercial MIP solver was not effective at generating good feasible solutions and improving the upper bound beyond the initial linear programming relaxation. Thus, we developed a specialized branching scheme and specialized heuristics for use with this formulation. Computational results indicate that using these techniques with the new formulation allow solution to optimality of some large instances, and yields low remaining optimality gaps for the majority of the remaining instances. Although the specialized branching strategy and heuristics yielded improved performance, this was critically enabled by the compactness of

the new formulation and tightness of its relaxation bound.

More computational experiments need to be performed to test the effectiveness of the new formulations in different settings. For example, we tested the case in which the number of possible realizations of the reference random variable, D, is large. The case in which D is small should also be tested. This case is interesting because it may represent a model in which the decision-maker wishes to enforce a small collection of risk constraints. In particular, from Lemma 39, we recognize the first-order stochastic dominance constraint as simply a collection of probabilistic constraints, or *Value-at-Risk* constraints, as they are known in the finance literature. In addition, when D is significantly smaller, it may be possible to significantly increase the number of possible realizations, N, of the random vector appearing in the constraints. Another setting in which to test the new formulations is in two-stage stochastic programming with stochastic dominance constraints, as has been recently studied in [29, 30], where they use the previous, less compact, formulations for the stochastic dominance constraints.

Finally, it will be interesting to study a Monte Carlo sampling based approximation scheme for problems with stochastic dominance constraints having more general distributions. If the reference random variable has a finite distribution, then the first order stochastic dominance constraint is a collection of finitely many probabilistic or chance constraints. Thus, results on sample approximations for probabilistic constraints may be applied to yield approximations for stochastic dominance constraints in which the random vector ξ appearing in the constraint may have general distribution. See Chapter 3 and references therein for results on sample approximations of probabilistic constraints. It will be interesting to explore whether the specific structure of the first order stochastic dominance constraint can yield results beyond direct application of the results for probabilistic constraints. For second order stochastic dominance, similar sample approximation results may be possible, but there is less previous work to build from.

REFERENCES

[1] Adams, W. P. and Sherali, H. D., "Mixed-integer bilinear programming problems," *Mathematical Programming*, vol. 59, pp. 279–305, 1993.

[2] Ahmed, S. and Shapiro, A., "The sample average approximation method for stochastic programs with integer recourse." Preprint available at www.optimization-online.org, February 2002.

[3] Atamtürk, A., Nemhauser, G. L., and Savelsbergh, M. W. P., "The mixed vertex packing problem," *Mathematical Programming*, vol. 89, pp. 35–53, 2000.

[4] Atlason, J., Epelman, M. A., and Henderson, S. G., "Call center staffing with simulation and cutting plane methods," *Annals of Operations Research*, vol. 127, pp. 333–358, 2004.

[5] Balas, E., "Disjunctive programming: Properties of the convex hull of feasible points.," *Discrete Applied Mathematics*, vol. 89, pp. 3–44, 1998.

[6] Bárány, I., Edmonds, J., and Wolsey, L. A., "Packing and covering a tree by subtrees," *Combinatorica*, vol. 6, pp. 221–233, 1986.

[7] Bárány, I., Van Roy, T. J., and Wolsey, L. A., "Strong formulations for multi-item capacitated lot sizing," *Management Science*, vol. 30, pp. 1255–1261, 1984.

[8] Bastin, F., Cirillo, C., and Toint, P. L., "Convergence theory for nonconvex stochastic programming with application to mixed logit," *Mathematical Programming*, vol. 108, pp. 207–234, 2006.

[9] Beale, E. and Tomlin, J., "Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables," in *Proc. Fifth Internat. Conf. on OR* (Lawrence, J., ed.), (London, UK), pp. 447–454, Tavistock Publications, 1970.

[10] Beasley, J., "Or-library: Distributing test problems by electronic mail," *Journal of the Operational Research Society*, vol. 41, pp. 1069–1072, 1990.

[11] Ben-Tal, A. and Nemirovski, A., "Robust convex optimization," *Mathematics of Operations Research*, vol. 23, pp. 769–805, 1998.

[12] Beraldi, P. and Ruszczyński, A., "The probabilistic set covering problem," Tech. Rep. RRR 32-2001, RUTCOR, 2001.

[13] Beraldi, P. and Ruszczyński, A., "A branch and bound method for stochastic integer programs under probabilistic constraints," *Optimization Methods and Software*, vol. 17, pp. 359–382, 2002.

[14] Beraldi, P. and Ruszczyński, A., "The probabilistic set-covering problem," *Operations Research*, vol. 50, pp. 956–967, 2002.

[15] Bertsimas, D. and Sim, M., "The price of robustness," *Operations Research*, vol. 52, pp. 35–53, 2004.

[16] Calafiore, G. C. and Campi, M. C., "Uncertain convex programs: randomized solutions and confidence levels," *Mathematical Programming*, vol. 102, pp. 25–46, 2005.

[17] Calafiore, G. C. and Campi, M. C., "The scenario approach to robust control design," *IEEE Transactions on Automatic Control*, vol. 51, pp. 742–753, 2006.

[18] Cheon, M.-S., Ahmed, S., and Al-Khayyal, F., "A branch-reduce-cut algorithm for the global optimization of probabilistically constrained linear programs," *Mathematical Programming*, vol. 108, pp. 617–634, 2006.

[19] Crowder, H., Johnson, E. L., and Padberg, M. W., "Solving large scale zero-one linear programming problems," *Operations Research*, vol. 31, pp. 803–834, 1983.

[20] Dai, L., Chen, H., and Birge, J. R., "Convergence properties of two-stage stochastic programming," *Journal of Optimization Theory and Applications*, vol. 106, pp. 489–509, 2000.

[21] Dentcheva, D., Prékopa, A., and Ruszczyński, A., "Concavity and efficient points of discrete distributions in probabilistic programming," *Mathematical Programming*, vol. 89, pp. 55–77, 2000.

[22] Dentcheva, D. and Ruszczyński, A., "Optimization with stochastic dominance constraints," *SIAM Journal on Optimization*, vol. 14, no. 2, pp. 548–566, 2003.

[23] Dentcheva, D. and Ruszczyński, A., "Optimality and duality theory for stochastic optimization problems with nonlinear dominance constraints," *Mathematical Programming*, vol. 99, pp. 329–350, March 2004.

[24] Dentcheva, D. and Ruszczyński, A., "Semi-infinite probabilistic optimization: first-order stochastic dominance constraints," *Optimization*, vol. 53, no. 5–6, pp. 583–601, 2004.

[25] Eppen, G. D. and Martin, R. K., "Solving multi-item capacitated lot-sizing problems using variable redefinition," *Operations Research*, vol. 35, pp. 832–848, 1987.

[26] Forrest, J., de la Nuez, D., and Lougee-Heimer, R., *CLP User Guide*, 2004.

[27] Ghaoui, L. E. and Lebret, H., "Robust solutions to least-squares problems with uncertain data," *SIAM Journal on Matrix Analysis and Applications*, vol. 18, pp. 1035–1064, 1997.

[28] Gollmer, R., Gotzes, U., Neise, F., and Schultz, R., "Risk modeling via stochastic dominance in power systems with dispersed generation," Tech. Rep. Preprint 651/2007, Department of Mathematics, University of Duisburg-Essen, 2007. www.uni-duisburg.de/FB11/disma/preprints.shtml (accessed July 2007).

[29] Gollmer, R., Gotzes, U., and Schultz, R., "Second-order stochastic dominance constraints induced by mixed-integer linear recourse," Tech. Rep. Preprint 644/2007, Department of Mathematics, University of Duisburg-Essen, 2007. www.uni-duisburg.de/FB11/disma/preprints.shtml (accessed July 2007).

[30] Gollmer, R., Neise, F., and Schultz, R., "Stochastic programs with first-order dominance constraints induced by mixed-integer linear recourse," Tech. Rep. Preprint 641/2006, Department of Mathematics, University of Duisburg-Essen, 2006. www.uni-duisburg.de/FB11/disma/preprints.shtml (accessed July 2007).

[31] Guan, Y., Ahmed, S., and Nemhauser, G. L., "Sequential pairing of mixed integer inequalities," *Discrete Optimization*, vol. 4, pp. 21–39, 2007.

[32] Günlük, O. and Pochet, Y., "Mixing mixed-integer inequalities," *Mathematical Programming*, vol. 90, pp. 429–457, 2001.

[33] Hardy, G., Littlewood, J. E., and Pólya, G., *Inequalities*. Cambridge University Press, 1934.

[34] Heilmann, W.-R., "A note on chance-constrained programming," *Journal of the Operational Research Society*, vol. 34, pp. 533–537, 1983.

[35] Henrion, R., Li, P., Möller, A., Steinbach, M. C., Wendt, M., and Wozny, G., "Stochastic optimization for operating chemical processes under uncertainty," in *Online Optimization of Large Scale Systems* (Grötschel, M., Krunke, S., and Rambau, J., eds.), pp. 457–478, Springer, 2001.

[36] Henrion, R. and Möller, A., "Optimization of a continuous distillation process under random inflow rate," *Computers and Mathematics with Applications*, vol. 45, pp. 247–262, 2003.

[37] Hoeffding, W., "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, pp. 13–30, 1963.

[38] Horst, R. and Tuy, H., *Global Optimization*. Berlin: Springer-Verlag, 1990.

[39] Hsu, V. N., "Dynamic capacity expansion problem with deferred expansion and age-dependent shortage cost," *Manufacturing & Service Operations Management*, vol. 4, pp. 44–54, 2002.

[40] Ilog, *Ilog CPLEX 9.0 User's Manual*, 2003.

[41] Kleywegt, A. J., Shapiro, A., and de Mello, T. H., "The sample average approximation method for stochastic discrete optimization," *SIAM Journal on Optimization*, vol. 12, pp. 479–502, 2001.

[42] Kuosmanen, T., "Efficient diversification according to stocastic dominance criteria," *Management Science*, vol. 50, pp. 1390–1406, 2004.

[43] Lejeune, M. A. and Ruszczyński, A., "An efficient trajectory method for probabilistic inventory-production-distribution problems," *Operations Research*, Forthcoming, 2007.

[44] Li, S. and Tirupati, D., "Dynamic capacity expansion problem with multiple products: Technology selection and timing of capacity additions," *Operations Research*, vol. 42, pp. 958–976, 1994.

[45] Linderoth, J., Shapiro, A., and Wright, S., "The empirical behavior of sampling methods for stochastic programming," *Annals of Operations Research*, vol. 142, pp. 215–241, 2006.

[46] Luss, H., "Operations research and capacity expansion problems: a survey," *Operations Research*, vol. 30, pp. 907–947, 1982.

[47] Marshall, A. W. and Olkin, I., *Inequalities: Theory of Majorization*. New York: Academic Press, 1979.

[48] Miller, A. J. and Wolsey, L. A., "Tight formulations for some simple mixed integer programs and convex objective integer programs," *Mathematical Programming*, vol. 98, pp. 73–88, 2003.

[49] Murr, M. R. and Prékopa, A., "Solution of a product substitution problem using stochastic programming," in *Probabilistic Constrained Optimization: Methodology and Applications* (Uryasev, S. P., ed.), pp. 252–271, Kluwer Academic Publishers, 2000.

[50] Nemhauser, G. L. and Wolsey, L. A., *Integer and Combinatorial Optimization*. Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley, 1988.

[51] Nemirovski, A. and Shapiro, A., "Scenario approximation of chance constraints," in *Probabilistic and Randomized Methods for Design Under Uncertainty* (Calafiore, G. and Dabbene, F., eds.), pp. 3–48, London: Springer, 2005.

[52] Nemirovski, A. and Shapiro, A., "Convex approximations of chance constrained programs," *SIAM Journal on Optimization*, vol. 17, pp. 969–996, 2006.

[53] Noyan, N., Rudolf, G., and Ruszczyński, A., "Relaxations of linear programming problems with first order stochastic dominance constraints," *Operations Research Letters*, vol. 34, pp. 653–659, November 2006.

[54] Noyan, N. and Ruszczyński, A., "Valid inequalities and restrictions for stochastic programming problems with first order stochastic dominance constraints," *Mathematical Programming*, 2007. To appear.

[55] Pochet, Y. and Wolsey, L. A., "Lot-sizing models with backlogging: Strong reformulations and cutting planes," *Mathematical Programming*, vol. 40, pp. 317–335, 1988.

[56] Prékopa, A., "On probabilistic constrained programmming," in *Proceedings of the Princeton Symposium on Mathematical Programming* (Kuhn, H. W., ed.), (Princeton, N.J.), pp. 113–138, Princeton University Press, 1970.

[57] Prékopa, A., "Probabilistic programming," in *Stochastic Programming* (Ruszczyński, A. and Shapiro, A., eds.), vol. 10 of *Handbooks in Operations Research and Management Science*, pp. 267–351, Elsevier, 2003.

[58] Rajagopalan, S., "Capacity expansion and equipment replacement: A unified approach," *Operations Research*, vol. 46, pp. 846–847, 1998.

[59] Rajagopalan, S., Singh, M. R., and Morton, T. E., "Capacity expansion and replacement in growing markets with uncertain technological breakthroughs," *Management Science*, vol. 44, pp. 12–30, 1998.

[60] Rajagopalan, S. and Soteriou, A. C., "Capacity acquisition and disposal with discrete facility sizes," *Management Science*, vol. 40, pp. 903–917, 1994.

[61] Ruszczyński, A., "Probabilistic programming with discrete distributions and precedence constrained knapsack polyhedra," *Mathematical Programming*, vol. 93, pp. 195–215, 2002.

[62] Saxena, A., "A short note on the probabilistic set covering problem." Available online at www.optimizationonline.org, March 2007.

[63] Saxena, A., Goyal, V., and Lejeune, M., "MIP reformulations of the probabilistic set covering problem." Available online at www.optimizationonline.org, February 2007.

[64] Shapiro, A., "Asymptotic behavior of optimal solutions in stochastic programming," *Mathematics of Operations Research*, vol. 18, pp. 829–845, 1993.

[65] Shapiro, A. and Homem-de-Mello, T., "On the rate of convergence of optimal solutions of Monte Carlo approximations of stochastic programs," *SIAM Journal on Optimization*, vol. 11, pp. 70–86, 2000.

[66] Shulman, A., "An algorithm for solving dynamic capacitated plant location problems with discrete expansion sizes," *Operations Research*, vol. 39, pp. 423–436, 1991.

[67] Takyi, A. K. and Lence, B. J., "Surface water quality management using a multiple-realization chance constraint method," *Water Resources Research*, vol. 35, pp. 1657–1670, 1999.

[68] Van Roy, T. J. and Erlenkotter, D., "A dual-based procedure for dynamic facility location," *Management Science*, vol. 28, pp. 1091–1105, 1982.

[69] Van Vyve, M., "The continuous mixing polyhedron," *Mathematics of Operations Research*, vol. 30, pp. 441–452, 2005.

[70] Vogel, S., "Stability results for stochastic programming problems," *Optimization*, vol. 19, pp. 269–288, 1988.

[71] Whitmore, G. and Findlay, M., eds., *Stochastic Dominance*. Lexington, MA: D.C. Heath and Company, 1978.