

New Formulations for Optimization Under Stochastic Dominance Constraints

James Luedtke
IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
jluedtk@us.ibm.com

April 26, 2008

Abstract

Stochastic dominance constraints allow a decision-maker to manage risk in an optimization setting by requiring their decision to yield a random outcome which stochastically dominates a reference random outcome. We present new integer and linear programming formulations for optimization under first and second-order stochastic dominance constraints, respectively. These formulations are more compact than existing formulations, and relaxing integrality in the first-order formulation yields a second-order formulation, demonstrating the tightness of this formulation. We also present a specialized branching strategy and heuristics which can be used with the new first-order formulation. Computational tests illustrate the potential benefits of the new formulations.

Keywords: Stochastic Programming, Stochastic Dominance Constraints, Risk, Probabilistic Constraints, Integer Programming

1 Introduction

Optimization 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 preferable in a strong sense than a given reference random outcome. Here, “preferable” is taken to mean that the random outcome we achieve stochastically dominates the reference outcome. A simple example application is to choose investments to 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, see e.g. [7]. Stochastic dominance constraints have also been used in risk modeling in power systems with dispersed generation [10]. In addition, dose-volume restrictions appearing

in radiation treatment planning problems [18] can be formulated as a first-order stochastic dominance constraint. Stochastic programming under stochastic dominance constraints has recently been studied in [4, 5, 6, 8, 11, 12, 24, 25, 26].

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

$$F(\eta) \leq G(\eta) \quad \forall \eta \in \mathbb{R}. \quad (1)$$

The random variable W dominates Y in the second order, written $W \succeq_{(2)} Y$, if

$$\mathbb{E}[\max\{\eta - W, 0\}] \leq \mathbb{E}[\max\{\eta - Y, 0\}] \quad \forall \eta \in \mathbb{R}. \quad (2)$$

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. [29]) $W \succeq_{(1)} Y$ if and only if

$$\mathbb{E}[h(W)] \geq \mathbb{E}[h(Y)]$$

for all nondecreasing functions $h : \mathbb{R} \rightarrow \mathbb{R}$ 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 $W \succeq_{(2)} Y$ if and only if

$$\mathbb{E}[h(W)] \geq \mathbb{E}[h(Y)]$$

for all nondecreasing and concave functions $h : \mathbb{R} \rightarrow \mathbb{R}$ 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 paper, we present new, computationally attractive formulations for *optimization under stochastic dominance constraints*. Let $X \subseteq \mathbb{R}^n$ and let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ 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 \mathbb{R}^m . Finally, let $g : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ 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

$$\max_x \{f(x) : x \in X, g(x, \xi) \succeq_{(1)} Y\} \quad (\text{FSD})$$

and

$$\max_x \{f(x) : x \in X, g(x, \xi) \succeq_{(2)} Y\}. \quad (\text{SSD})$$

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 two new linear formulations for SSD which have $O(N + D)$ constraints, as opposed to $O(ND)$ constraints in an existing linear formulation. Computational results indicate that this yields significant improvement in solution time for instances in which $N = D$.

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 [24], and shown to be a tight relaxation of FSD in [25].
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 relatively large instances using this approach.

We do not make any assumptions on the set X or the mapping g in the development of the 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 [6] 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 the new formulation for FSD yields a formulation for SSD.

In Section 2 we review some basic results about stochastic dominance and present existing formulations for FSD and SSD. In Section 3 we present the new formulations for SSD and in Section 4 we present the new formulation for FSD. In Section 5 we present a specialized branching scheme and some heuristics for solving the new formulation of FSD. In Section 6 we present some illustrative computational results, and we close with some concluding remarks in Section 7.

2 Review of Existing Results

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

$$\mu\{W = w_i\} = p_i \quad i \in \mathcal{N} := \{1, \dots, N\} \quad (3)$$

$$\nu\{Y = y_k\} = q_k \quad k \in \mathcal{D} := \{1, \dots, D\} \quad (4)$$

where μ and ν are the probability distributions induced by W and Y respectively. Furthermore, we assume without loss of generality that $y_1 < y_2 < \dots < y_D$.

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

$$w_i \leq g(x, \xi^i) \quad i \in \mathcal{N} \quad (5)$$

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 (5), the reader should think of the values w_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 [4, 5]. We let $y_0 \in \mathbb{R}$ be such that $y_0 < y_1$ and introduce the notation $(\cdot)^+ = \max\{0, \cdot\}$.

Lemma 1. *Let W, Y be random variables with distributions given by (3) and (4). Then, $W \succeq_{(2)} Y$ if and only if*

$$\mathbb{E}[(y_k - W)^+] \leq \mathbb{E}[(y_k - Y)^+] \quad k \in \mathcal{D} \quad (6)$$

and $W \succeq_{(1)} Y$ if and only if

$$\mu\{W < y_k\} \leq \nu\{Y \leq y_{k-1}\} \quad k \in \mathcal{D}. \quad (7)$$

The key simplification is that the infinite sets of inequalities in the definitions (1) and (2) can be reduced to a finite set when Y has a finite distribution.

Second-order stochastic dominance (SSD) constraints are known to define a convex feasible region [4]. In fact, condition (6) 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 $(y_k - w_i)^+$, see e.g. [4]. Thus, $W \succeq_{(2)} Y$ if and only if there exists $s \in \mathbb{R}_+^{ND}$ such that

$$\begin{aligned} \sum_{i=1}^N p_i s_{ik} &\leq \sum_{j=1}^D q_j (y_k - y_j)^+ & k \in \mathcal{D} \\ s_{ik} + w_i &\geq y_k & i \in \mathcal{N}, k \in \mathcal{D}. \end{aligned}$$

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 (6) 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 [5]. 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 [9] or the commercial solver Ilog CPLEX [14]. In addition, if the base problem contains integer restrictions on some of the variables x , a linear formulation is advantageous because it can be solved as mixed-integer *linear* program, as opposed to a mixed-integer nonlinear program.

The condition for second-order dominance given in (6) can also be interpreted as a collection of D *integrated chance constraints*, as introduced by Klein Haneveld [15]. In [16], Klein Haneveld and van der Vlerk propose a cutting plane algorithm for solving problems with integrated chance constraints and demonstrate its computational efficiency. Due to (6), this approach can also be used

for problems with second-order stochastic dominance constraints, as has been observed in [8]. Independently, Gabor and Ruszczyński [26] proposed a primal cutting plane method and a dual column generation method for optimization problems with second-order stochastic dominance constraints, and the primal method is shown to be computationally efficient. In the case of finite distributions, the primal cutting plane method is equivalent to the cutting plane method used for integrated chance constraints in [16].

Condition (7) can be used to derive a MIP formulation for a first-order stochastic dominance (FSD) constraint [24, 25]. $W \succeq_{(1)} Y$ if and only if there exists β such that

$$\sum_{i=1}^N p_i \beta_{ik} \leq \sum_{j=1}^{k-1} q_j \quad k \in \mathcal{D} \quad (8)$$

$$\begin{aligned} w_i + M_{ik} \beta_{ik} &\geq y_k & i \in \mathcal{N}, k \in \mathcal{D} \\ \beta_{ik} &\in \{0, 1\} & i \in \mathcal{N}, k \in \mathcal{D}. \end{aligned} \quad (9)$$

We refer to this formulation as FDMIP. Here, M_{ik} is sufficiently large to guarantee that if $\beta_{ik} = 1$, then the corresponding constraint (9) will be redundant. For example, if other constraints in the model imply $w_i \geq l_i$, then we can take $M_{ik} = y_k - l_i$. Although this formulation was presented in [24, 25], 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 use the cutting plane algorithm proposed in [26] for solving problem SSD, which yields bounds for FSD, and then they improve these bounds using disjunctive cuts [1]. In addition, problem SSD is used as a basis for heuristics to find feasible solutions for FSD. It is demonstrated in [25] that the bounds from using SSD as a relaxation of FSD are usually good, and that the heuristics are able to obtain good feasible solutions. However, these results do not yield a convergent algorithm for finding an optimal solution to FSD. As these results are based on solving problem SSD, an easily implementable and computationally efficient formulation for solving SSD will also enhance this approach.

3 New Formulations for Second-Order Stochastic Dominance

When all outcomes are equally likely and $N = D$, a formulation for second-order stochastic dominance based on majorization theory [13, 21] can be derived which introduces $O(N^2)$ variables but only $O(N)$ rows. This has been done implicitly in [6] when proving that in this case the SSD constraint yields the convexification of the FSD constraint, and explicitly in [17] to derive a test

for second-order stochastic dominance. In this section we present two formulations for second-order dominance between finitely distributed random variables which do not require all outcomes to be equally likely and allow $N \neq D$. The formulations will not be based on the majorization theory, and instead will follow from the following theorem due to Strassen, which we state here in a form that is convenient for our use.

Theorem 2 (e.g. Corollary 1.5.21 in [22]). *Let W and Y be random variables with finite means. Then $W \succeq_{(2)} Y$ if and only if there exists random variables W' and Y' , with the same distributions as W and Y , such that almost surely*

$$\mathbb{E}[Y'|W'] \leq W'. \quad (10)$$

Theorem 3. *Let W, Y be random variables with distributions given by (3) and (4). Then $W \succeq_{(2)} Y$ if and only if there exists $\pi \in \mathbb{R}_+^{ND}$ which satisfies*

$$\sum_{j=1}^D y_j \pi_{ij} \leq w_i \quad i \in \mathcal{N} \quad (11)$$

$$\sum_{j=1}^D \pi_{ij} = 1 \quad i \in \mathcal{N} \quad (12)$$

$$\sum_{i=1}^N p_i \pi_{ik} = q_k \quad k \in \mathcal{D}. \quad (13)$$

Proof. First suppose $W \succeq_{(2)} Y$. By Theorem 2, there exists random variables W' and Y' (defined, say, on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$) such that $\mathbb{E}[Y'|W'] \leq W'$ and $\mathbb{P}\{W' = w_i\} = p_i$ for $i \in \mathcal{N}$ and $\mathbb{P}\{Y' = y_k\} = q_k$ for $k \in \mathcal{D}$. Define a vector $\pi \in \mathbb{R}^{ND}$ by $\pi_{ik} = \mathbb{P}\{Y' = y_k | W' = w_i\}$ for $i \in \mathcal{N}$, $k \in \mathcal{D}$. By definition, $\pi \geq 0$ and $\sum_{k \in \mathcal{D}} \pi_{ik} = 1$ for each $i \in \mathcal{N}$. Also, for each $k \in \mathcal{D}$

$$q_k = \mathbb{P}\{Y' = y_k\} = \sum_{i=1}^N \mathbb{P}\{Y' = y_k | W' = w_i\} \mathbb{P}\{W' = w_i\} = \sum_{i=1}^N p_i \pi_{ik}.$$

Finally, for each $i \in \mathcal{N}$

$$w_i \geq \mathbb{E}[Y'|W' = w_i] = \sum_{k=1}^D y_k \pi_{ik}$$

and hence π satisfies (11) - (13).

Now suppose there exists $\pi \in \mathbb{R}_+^{ND}$ which satisfies (11) - (13). Let $\Omega = \{(i, k) : i \in \mathcal{N}, k \in \mathcal{D}\}$ and define the probability measure \mathbb{P} on Ω by $\mathbb{P}\{(i, k)\} = p_i \pi_{ik}$. Note that \mathbb{P} is well-defined since by (12) $\sum_{k \in \mathcal{D}} \sum_{i \in \mathcal{N}} p_i \pi_{ik} = 1$. Now define W' by $W'((i, k)) = w_i$ for $i \in \mathcal{N}$, $k \in \mathcal{D}$ and Y' by $Y'((i, k)) = y_k$ for $i \in \mathcal{N}$, $k \in \mathcal{D}$. Then, $\mathbb{P}\{W' = w_i\} = p_i \sum_{k \in \mathcal{D}} \pi_{ik} = p_i$ by (12), and so W' has

the same distribution as W . Also, $\mathbb{P}\{Y' = y_k\} = \sum_{i \in \mathcal{N}} p_i \pi_{ik} = q_k$ by (13), and so Y' has the same distribution as Y . Finally, for each $i \in \mathcal{N}$,

$$\mathbb{E}[Y'|W' = w_i] = \sum_{k=1}^D y_k \pi_{ik} \leq w_i$$

by (11). It follows from Theorem 2 that $W \succeq_{(2)} Y$. □

To use Theorem 3 to obtain a formulation for SSD, we replace w_i with $g(x, \xi^i)$ so that (11) becomes

$$g(x, \xi^i) \geq \sum_{j=1}^D y_j \pi_{ij} \quad i \in \mathcal{N} \quad (14)$$

and thus obtain our first new formulation for SSD given by

$$f_{\text{SSD}}^* = \max_{x, \pi} \{f(x) : (12), (13), (14), x \in X, \pi \in \mathbb{R}_+^{ND}\}. \quad (\text{cSSD1})$$

This formulation, which we refer to as cSSD1, introduces ND variables and $O(N + D)$ linear constraints.

Theorem 4. *Let W, Y be random variables with distributions given by (3) and (4). Then $W \succeq_{(2)} Y$ if and only if there exists $\pi \in \mathbb{R}_+^{ND}$ which satisfies (11), (12) and*

$$\sum_{i=1}^N p_i \sum_{j=1}^{k-1} (y_k - y_j) \pi_{ij} \leq \sum_{j=1}^{k-1} (y_k - y_j) q_j \quad k = 2, \dots, D. \quad (15)$$

Proof. First suppose $W \succeq_{(2)} Y$. Then by Theorem 3 there exists $\pi \in \mathbb{R}_+^{ND}$ which satisfies (11) - (13). Then,

$$\sum_{i=1}^N p_i \sum_{j=1}^{k-1} (y_k - y_j) \pi_{ij} = \sum_{j=1}^{k-1} (y_k - y_j) \sum_{i=1}^N p_i \pi_{ij} = \sum_{j=1}^{k-1} (y_k - y_j) q_j$$

for $k = 2, \dots, D$ by (13) and hence π satisfies (15).

Now suppose there exists $\pi \in \mathbb{R}_+^{ND}$ which satisfies (11), (12) and (15). For any $i \in \mathcal{N}$, $k \in \mathcal{D}$ we have

$$\begin{aligned} (y_k - w_i)^+ &\leq (y_k - \sum_{j=1}^D y_j \pi_{ij})^+ && \text{by (11)} \\ &= \left(\sum_{j=1}^D (y_k - y_j) \pi_{ij} \right)^+ && \text{by (12)} \\ &\leq \sum_{j=1}^D (y_k - y_j)^+ \pi_{ij} = \sum_{j=1}^{k-1} (y_k - y_j) \pi_{ij} && \text{since } \pi \geq 0. \end{aligned}$$

Thus, for each $k \in \mathcal{D}$,

$$\begin{aligned} \mathbb{E}[(y_k - W)^+] &= \sum_{i=1}^N p_i (y_k - w_i)^+ \leq \sum_{i=1}^N p_i \sum_{j=1}^{k-1} (y_k - y_j) \pi_{ij} \\ &\leq \sum_{j=1}^{k-1} (y_k - y_j) q_j = \mathbb{E}[(y_k - Y)^+]. \end{aligned}$$

where the second inequality follows from (15). Thus, condition (6) in Lemma 1 implies that $W \succeq_{(2)} Y$. \square

When using the formulation arising from Theorem 4, it is beneficial for computational purposes to use an equivalent formulation in which we introduce variables $v \in \mathbb{R}^D$ and replace the constraints in (15) with the $2D$ constraints

$$v_j - \sum_{i=1}^N p_i \pi_{ij} = 0 \quad j \in \mathcal{D} \quad (16)$$

$$\sum_{j=1}^{k-1} (y_k - y_j) v_j \leq \sum_{j=1}^{k-1} (y_k - y_j) q_j \quad k \in \mathcal{D}. \quad (17)$$

Thus, our second formulation for SSD is given by

$$f_{\text{SSD}}^* = \max_{x, \pi, v} \{f(x) : (12), (14), (16), (17), x \in X, \pi \in \mathbb{R}_+^{ND}, v \in \mathbb{R}^D\}. \quad (\text{cSSD2})$$

The advantage of using (16) and (17) instead of (15) is that this yields a formulation with $O(ND)$ nonzeros, as compared to $O(ND^2)$ nonzeros if we used (15). This formulation, which we refer to as cSSD2, introduces $(N+1)D$ new variables and $O(N+D)$ linear constraints.

One motivation for introducing formulation cSSD2 is that we have empirical evidence (Section 6) that it performs better than cSSD1, at least when solved with the dual simplex algorithm (as implemented in Ilog CPLEX [14]). cSSD2 is also interesting because a slight generalization of this formulation can be used to compactly model a collection of expected shortfall constraints of the form:

$$\mathbb{E}[(y_k - g(x, \xi))^+] \leq L_k \quad k \in \mathcal{D} \quad (18)$$

where $y_1 < y_2 < \dots < y_D$ are given targets and $0 \leq L_1 \leq L_2 \leq \dots \leq L_D$ are given limits on the expected shortfalls of these targets. Note that if

$$L_k = \mathbb{E}[(y_k - Y)^+] \quad k \in \mathcal{D} \quad (19)$$

where Y is a random variable with distribution given by (4), then the inequalities (18) are equivalent to (6), and hence (18) are satisfied exactly when $W \succeq_{(2)} Y$. If L_k are not defined by a random variable Y as in (19), formulation cSSD2 can still be extended to directly model (18), provided that $L_1 = 0$ which implies that we require $g(x, \xi) \geq y_1$ with probability 1. All that is required is to replace the term $\sum_{j=1}^{k-1} (y_k - y_j) q_j$ in the right-hand side of (15) with L_k .

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 [17]. Once again, we are able to generalize this to the case in which the probabilities are not necessarily equal and $N \neq D$.

Theorem 5. *Let W, Y be random variables with distributions given by (3) and (4). Then $W \succeq_{(1)} Y$ if and only if there exists $\pi \in \{0, 1\}^{ND}$ such that (w, π) satisfy (11),(12) and*

$$\sum_{i=1}^N p_i \sum_{j=1}^{k-1} \pi_{ij} \leq \sum_{j=1}^{k-1} q_j \quad k = 2, \dots, D. \quad (20)$$

Proof. First suppose $W \succeq_{(1)} Y$. Then, by condition (7) in Lemma 1 we have

$$\mu\{W < y_k\} \leq \nu\{Y \leq y_{k-1}\} = \sum_{i=1}^{k-1} q_i \quad (21)$$

for each $k \in \mathcal{D}$. In particular, (21) for $k = 1$ implies $\mu\{W \geq y_1\} = 1$, and hence $w_i \geq y_1$ for all $i \in \mathcal{N}$. Now, for each $i \in \mathcal{N}$, $k \in \mathcal{D}$, let $\pi_{ik} = 1$ if $y_k \leq w_i < y_{k+1}$ and $\pi_{ik} = 0$ otherwise, where we take $y_{D+1} \equiv +\infty$. Then, $\sum_{k=1}^D \pi_{ik} = 1$ because $w_i \geq y_1$ for all i , and so π satisfies (12). It is also immediate by the definition of π_{ik} that $w_i \geq \sum_{k=1}^D y_k \pi_{ik}$ and so π satisfies (11). Finally, note that $w_i < y_k$ if and only if $\sum_{j=1}^{k-1} \pi_{ij} = 1$. Thus,

$$\mu\{W < y_k\} = \sum_{i \in \mathcal{N}: w_i < y_k} p_i = \sum_{i=1}^N p_i \sum_{j=1}^{k-1} \pi_{ij}.$$

This combined with (21) proves that π satisfies (20).

Now suppose $\pi \in \{0, 1\}^{ND}$ satisfies (11),(12) and (20). Note that by (11) and (12) if $w_i < y_k$, then $\sum_{j=1}^{k-1} \pi_{ij} = 1$. Thus,

$$\mu\{W < y_k\} = \sum_{i \in \mathcal{N}: w_i < y_k} p_i \leq \sum_{i=1}^N p_i \sum_{j=1}^{k-1} \pi_{ij} \leq \sum_{j=1}^{k-1} q_j = \nu\{Y \leq y_{k-1}\}$$

where the second inequality follows from (20). It follows that $W \succeq_{(1)} Y$ by condition (7) in Lemma 1. \square

As in the new formulation for second-order stochastic dominance cSSD2, for computational purposes it is beneficial to use the equivalent formulation

obtained by introducing variables $v \in \mathbb{R}^D$ and replacing the constraints (20) with the constraints

$$v_j - \sum_{i=1}^N p_i \pi_{ij} = 0 \quad j \in \mathcal{D} \quad (22)$$

$$\sum_{j=1}^{k-1} v_j \leq \sum_{j=1}^{k-1} q_j \quad k \in \mathcal{D}. \quad (23)$$

Thus, taking $w_i = g(x, \xi^i)$, and using (22) and (23) in place of (20) Theorem 5 yields the formulation for FSD given by

$$f_{\text{FSD}}^* = \max_{x, \pi} \left\{ f(x) : (12), (14), (22), (23), x \in X, \pi \in \{0, 1\}^{ND} \right\}. \quad (\text{cFSD})$$

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*.

Theorem 6. *Let W, Y be random variables with distributions given by (3) and (4). 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 $\pi \in \mathbb{R}_+^{ND}$ such that (w, π) satisfy (11), (12) and (20).*

Proof. Let $\pi \in \mathbb{R}_+^{ND}$ and (w, π) satisfy (11), (12) and (20). Then,

$$\begin{aligned} \sum_{i=1}^N p_i \sum_{j=1}^{k-1} \pi_{ij} (y_k - y_j) &= \sum_{i=1}^N p_i \sum_{j=1}^{k-1} \pi_{ij} \sum_{l=j+1}^k (y_l - y_{l-1}) \\ &= \sum_{l=2}^k (y_l - y_{l-1}) \sum_{i=1}^N p_i \sum_{j=1}^{l-1} \pi_{ij} \\ &\leq \sum_{l=1}^k (y_l - y_{l-1}) \sum_{j=1}^{l-1} q_j \quad \text{by (20)} \\ &= \sum_{j=1}^{k-1} q_j (y_k - y_j). \end{aligned}$$

and hence π also satisfies (15) which implies $W \succeq_{(2)} Y$ by Theorem 4.

Now suppose $W \succeq_{(2)} Y$. Then by Theorem 3 there exists $\pi \in \mathbb{R}_+^{ND}$ which satisfies (11) - (13). Then, (13) implies

$$\sum_{i=1}^N p_i \sum_{j=1}^{k-1} \pi_{ik} = \sum_{j=1}^{k-1} \sum_{i=1}^N p_i \pi_{ik} = \sum_{j=1}^{k-1} q_j$$

for $k = 2, \dots, D$ and hence (20) holds. \square

As a result, we obtain another formulation for SSD, but more importantly, 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.

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 $y_k > y_{k-1} > \dots > y_1$, then we also know $w_i \geq y_{k-1} > \dots > y_1$. Thus, we lose nothing by setting $\beta_{i,k-1} = \dots = \beta_{i1} = 0$. Hence, we can add the inequalities

$$\beta_{ik} \leq \beta_{i,k+1} \quad i \in \mathcal{N}, k \in \mathcal{D} \quad (24)$$

and maintain a valid formulation. The inequalities (9) can then be replaced by

$$w_i - \sum_{k=1}^D (\beta_{i,k+1} - \beta_{ik}) y_k \geq 0 \quad i \in \mathcal{N}$$

which together with inequalities (24) 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 $\beta_{i,D+1} = 1$.

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. As has been shown in [25], 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 the MIP solver we use (Ilog CPLEX 9.0 [14]) 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 present a specialized branching approach and two heuristics which exploit the structure of this formulation. The computational benefits of these techniques will be demonstrated in Section 6.

5.1 Branching for FSD

Standard variable branching for mixed-integer programming would select a variable π_{ij} which is fractional in the current node relaxation solution, and then branch to create two new nodes, one with π_{ij} fixed to one and one with π_{ij} fixed to zero. However, the constraints (11) and (12) 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 of Type 1* (SOS1), that is, at most one of

the variables in this set can be positive. As a result, it is natural to consider using an SOS1 branching rule (see, e.g. [2]). 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* $k \in \{2, \dots, D\}$. 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 an SOS1 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 (7)

$$\mu\{W < y_k\} \leq \nu\{Y \leq y_{k-1}\} \quad k \in \mathcal{D}.$$

At each node in which we must branch, we find $k^* = \min\{k \in \mathcal{D} : \mu\{W < y_k\} > \nu\{Y \leq y_{k-1}\}\}$ 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 5, 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 (11) and (12) 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

$$w_i < y_{k^*} \tag{25}$$

$$\sum_{j < k^*} \pi_{ij} < 1. \tag{26}$$

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 (26), 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 (20). If there are multiple set indices which satisfy (25) and (26), we choose an 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 (11) forces $w_i \geq y_{k^*}$. Because of (25), this will make the current relaxation solution infeasible to this branch, and will promote feasibility of (7) at the currently infeasible level k^* . In the second branch, we enforce $\sum_{j < k^*} \pi_{ij} = 1$ and because of (26) 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 (7) is made, whereas by selecting k^* as small as possible, reasonable progress is also made in the second branch since this enforces $\pi_{ij} = 0$ for all $j \geq k^*$.

5.2 Heuristics for FSD

We now present some heuristics we have developed that can be used with formulation cFSD. We first present a simple and efficient heuristic, called the *order-preserving heuristic*, and then present a variant of a diving heuristic which can be integrated with the order-preserving heuristic.

Order-preserving heuristic

Given a solution x^* to a relaxation of cFSD, let $w^* \in \mathbb{R}^N$ 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 (12) and (20), and then solve the problem with π fixed to $\hat{\pi}$. 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 (14) 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 1.

Algorithm 1: Order-preserving heuristic

Data: $w^* \in \mathbb{R}^N$

- 1 Sort w^* to obtain $\{i_1, \dots, i_N\} = \mathcal{N}$ with $w_{i_1}^* \leq w_{i_2}^* \leq \dots \leq w_{i_N}^*$;
- 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 $\text{cSSD}(\hat{\pi}) = \max_x \{f(x) : x \in X, g(x, \xi_i) \geq \sum_{j=1}^D \hat{\pi}_{ij} y_j \ i \in \mathcal{N}\}$;
- 10 **if** $\text{cSSD}(\hat{\pi})$ *is feasible* **then**
- 11 Let \hat{x} be the optimal solution to $\text{cSSD}(\hat{\pi})$;
- 12 **return** $(\hat{x}, \hat{\pi})$;
- 13 **end**

The algorithm begins by sorting the values of w^* . Then, in lines 2 to 8 a solution $\hat{\pi}$ is constructed which is feasible to (12) and (20) by working in this order. To see that $\hat{\pi}$ satisfies (12), 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 $\{i_1, \dots, i_N\} = \mathcal{N}$, 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 (20) 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 1 is the sorting of w^* , and solving of $\text{cSSD}(\hat{\pi})$. Note that this problem is small relative to the original problem cFSD , since the $O(ND)$ variables π are fixed, the constraints (12) and (20) no longer need to be considered, and the constraints (14) reduce to lower bounds on the functions $g(x, \xi^i)$ for $i \in \mathcal{N}$.

Integrated order-preserving and diving heuristic

Diving is a classic heuristic strategy for integer programs which alternates between fixing one or more integer variables based on the current linear programming (LP) relaxation solution and re-solving the relaxation. We have developed a variant of the diving heuristic for solving cSSD , which we call the *Aggressive Diving Heuristic*. For brevity, we only outline the idea of the heuristic here; for details, we refer the reader to [19]. Within each iteration of the aggressive diving heuristic, the heuristic repeatedly selects the index $i \in \mathcal{N}$ which has minimum value of $w_i^* = g(x^*, \xi^i)$ and has not yet had π_{ij} fixed to one for any $j \in \mathcal{D}$. A variable π_{ik} is then fixed to one, where k is the minimum index such that π_{ik} could feasibly be fixed to one and still satisfy (20). This is done until one of the fixings causes inequality (14) to be violated by the current solution, that is, until a π_{ik} is fixed to one with $w_i^* < y_k$. Also within an iteration, a similar sequence of fixings is done for indices $i \in \mathcal{N}$ which have *maximum* value of w_i^* until one of the fixings implies (14) is violated by the current solution. After these fixings have been done, the LP relaxation is re-solved, and the next iteration begins. The heuristic terminates when the current LP relaxation yields a feasible integer solution or is infeasible (where infeasibility would be caused by the lower bounds implied by (14) due to the fixed variables). The key advantages of the aggressive diving heuristic are that it fixes multiple variables in each iteration, leading to faster convergence, and the variables are fixed in such a way that constraints (12) and (20) will not become violated.

Integration of the order-preserving heuristic with the aggressive diving heuristic is accomplished by calling the order-preserving heuristic during each iteration of the diving heuristic, using the current relaxation solution. If this yields an improved feasible solution, it is saved, but the heuristic still continues the dive until it terminates. At the end, the best feasible solution found over all iterations in the dive is reported.

6 Computational Results

We conducted computational experiments to test the new formulations for stochastic dominance. Following [17] and [25], we conducted 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 the 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

$$\max \left\{ \sum_{j=1}^n r_j x_j : x \in X, \sum_{j=1}^n R_j x_j \succeq_{(k)} Y \right\} \quad k = 1, 2 \quad (27)$$

where $X = \{x \in \mathbb{R}_+^n : \sum_{j=1}^n x_j = 1\}$.

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_j^i is the return that is achieved under outcome i for asset j , then the distribution of Y is given by $\nu\{Y = \sum_{j=1}^n R_j^i/n\} = 1/N$ 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 [14] 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.

6.1 Second-Order Dominance

We first compared the solution times using the formulations SDLP, cSSD1 and cSSD2 to solve the portfolio optimization problem (27) with second-order dominance constraint ($k = 2$ in (27)). We tested seven 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 1 gives the solution time and number of simplex iterations for each formulation on each instance. From this table it is clear that when using a commercial LP solver, the new formulations cSSD1 and cSSD2 allow much more efficient solution of SSD. Formulation cSSD1 yields a solution an order of magnitude faster than SDLP, whereas cSSD2 yields a solution roughly two

Table 1: Computational results for SSD formulations.

Year	N	Solution Time (s)			Iterations		
		SDLP	cSSD1	cSSD2	SDLP	cSSD1	cSSD2
2005	200	103	18	3	30851	10336	1921
	300	1063	61	19	76438	16879	6019
	400	4859	127	23	118328	17692	5698
	500	10345	509	17	121067	34380	4770
	600	27734	528	39	202490	40430	5854
	700	69486	3366	434	318030	112848	20788
	800	*100122	8272	1476	*361600	222967	42773
2006	200	83	13	3	20009	7449	2134
	300	883	44	8	52457	12004	3244
	400	4253	190	25	109493	24398	5549
	500	11365	332	63	117559	37086	7904
	600	43927	670	198	307680	41360	16443
	700	58947	6067	94	346077	173483	13026
	800	*100100	10406	50	*433400	245401	6307
2007	200	122	25	9	19359	13771	4597
	300	757	61	30	64795	15585	8253
	400	4292	214	59	89731	28024	8265
	500	12551	609	178	154287	46973	14914
	600	27492	1213	271	172905	66164	18611
	700	59144	1888	338	308064	92365	19009
	800	*100095	23171	74	*385700	544342	8174

* Not solved in time limit.

orders of magnitude faster. Both formulation cSSD1 and cSSD2 have $O(N)$ 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 these formulations is significantly less. The additional reduction in computation time obtained from formulation cSSD2 can be explained by the large reduction in the number of simplex iterations.

We should stress that because $N = D$ in this test, the relative improvement of cSSD1 and cSSD2 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.

6.2 First-Order Dominance

We next present results of the tests on the portfolio optimization problem (27) in which a first-order stochastic constraint is enforced ($k = 1$ in (27)).

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 of Type 1, allowing CPLEX to perform its general purpose SOS1 branching, as discussed in Section 5.1. We found that this yields better results than having CPLEX perform its default single variable branching. Note that the 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 integrated with the order-preserving heuristic. 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 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 2. 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 which the

Table 2: Comparison of root LP relaxations for FSD formulations.

Year	N	Time (s)			Percent above cFSD UB	
		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%

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 2 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 10,000 seconds. Table 3 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 (these are the instances with ‘-’ in the ‘‘Gap’’ column). From Table 3 we observe that even without the use of specialized heuristic

Table 3: Comparison of optimality gaps for FSD after time limit.

Year	N	Optimality Gap			cFSD+H+B	
		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.

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 4 gives more detailed results for the methods based on formulation cFSD for the 2005 instances (results for the other instances yield similar insights and are excluded for brevity). 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. Table 4 also reports the percent by

Table 4: Lower and upper bounds results using cFSD.

Year	N	% UB below Root UB			% LB below Best UB		
		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%

** No feasible solution found.

which the value of the best feasible solution found (LB) is below the best upper bound found over all methods (Best UB). These results indicate that the specialized heuristic significantly improves the value of the feasible solutions found, and that integrating the specialized branching with the heuristic often yields even further improvement in solution quality.

7 Concluding Remarks

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 since this is likely the case when a stochastic dominance constraint is used to model a collection of risk constraints. It would be particularly interesting to test these formulations for radiation treatment planning models with dose-volume constraints. We expect that when D is small it will 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 [11, 12], 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. Results on sample approximations for probabilistic constraints (e.g. [3, 20, 23]) may be applied to yield approximations for first-order stochastic dominance constraints in which the random vector ξ appearing in the constraint may have general distribution. 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. Similarly, results on sample approximations for optimization problems with expected value constraints (e.g. [27, 28]) may be applied to yield approximations for second-order dominance constraints.

Acknowledgements

The author is grateful to Shabbir Ahmed for his helpful comments on a draft of this paper. The author also thanks Darinka Dentcheva, Andrzej Ruszczyński, and an anonymous referee for pointing out the connection between the SSD formulations presented and Strassen's theorem.

References

- [1] E. BALAS, *Disjunctive programming: Properties of the convex hull of feasible points.*, Discrete Appl. Math., 89 (1998), pp. 3–44.
- [2] E. BEALE AND J. TOMLIN, *Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables*, in Proc. Fifth Internat. Conf. on OR, J. Lawrence, ed., London, UK, 1970, Tavistock Publications, pp. 447–454.
- [3] G. CALAFIORE AND M. CAMPI, *The scenario approach to robust control design*, IEEE Trans. Automat. Control, 51 (2006), pp. 742–753.
- [4] D. DENTCHEVA AND A. RUSZCZYŃSKI, *Optimization with stochastic dominance constraints*, SIAM J. Optim., 14 (2003), pp. 548–566.
- [5] ———, *Optimality and duality theory for stochastic optimization problems with nonlinear dominance constraints*, Math. Program., 99 (2004), pp. 329–350.
- [6] ———, *Semi-infinite probabilistic optimization: first-order stochastic dominance constraints*, Optimization, 53 (2004), pp. 583–601.
- [7] ———, *Portfolio optimization with stochastic dominance constraints*, Journal of Banking & Finance, 30 (2006), pp. 433–451.
- [8] C. FÁBIÁN, G. MITRA, AND D. ROMAN, *Processing second-order stochastic dominance models using cutting-plane representations*. CARISMA Technical Report CTR/75/08, 2008.
- [9] J. FORREST, D. DE LA NUEZ, AND R. LOUGEE-HEIMER, *CLP User Guide*, 2004.
- [10] R. GOLLMER, U. GOTZES, F. NEISE, AND R. SCHULTZ, *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.

- [11] R. GOLLMER, U. GOTZES, AND R. SCHULTZ, *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.
- [12] R. GOLLMER, F. NEISE, AND R. SCHULTZ, *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.
- [13] G. HARDY, J. LITTLEWOOD, AND G. PÓLYA, *Inequalities*, Cambridge University Press, 1934.
- [14] ILOG, *Ilog CPLEX 9.0 User's Manual*, 2003.
- [15] W. K. KLEIN HANEVELD, *Duality in Stochastic Linear and Dynamic Programming*, vol. 274 of Lecture Notes in Economics and Math. Systems, Springer-Verlag, New York, NY, 1986.
- [16] W. K. KLEIN HANEVELD AND M. H. VAN DER VLERK, *Integrated chance constraints: reduced forms and an algorithm*, *Comput. Manage. Sci.*, 3 (2006), pp. 245–269.
- [17] T. KUOSMANEN, *Efficient diversification according to stochastic dominance criteria*, *Manage. Sci.*, 50 (2004), pp. 1390–1406.
- [18] E. LEE, T. FOX, AND I. CROCKER, *Integer programming applied to intensity-modulated radiation therapy treatment planning*, *Ann. Oper. Res.*, 119 (2003), pp. 165–181.
- [19] J. LUEDTKE, *Integer Programming Approaches to Some Non-convex and Stochastic Optimization Problems*, PhD thesis, Georgia Institute of Technology, Atlanta, GA, 2007. Available online at etd.gatech.edu.
- [20] J. LUEDTKE AND S. AHMED, *A sample approximation approach for optimization with probabilistic constraints*, *SIAM J. Optim.*, (2008). To appear.
- [21] A. MARSHALL AND I. OLKIN, *Inequalities: Theory of Majorization*, Academic Press, New York, 1979.
- [22] A. MÜLLER AND D. STOYAN, *Comparison Methods for Stochastic Models and Risks*, John Wiley & Sons, LTD, Chichester, England, 2002.
- [23] A. NEMIROVSKI AND A. SHAPIRO, *Scenario approximation of chance constraints*, in *Probabilistic and Randomized Methods for Design Under Uncertainty*, G. Calafiore and F. Dabbene, eds., Springer, London, 2005, pp. 3–48.
- [24] N. NOYAN, G. RUDOLF, AND A. RUSZCZYŃSKI, *Relaxations of linear programming problems with first order stochastic dominance constraints*, *Oper. Res. Lett.*, 34 (2006), pp. 653–659.

- [25] N. NOYAN AND A. RUSZCZYŃSKI, *Valid inequalities and restrictions for stochastic programming problems with first order stochastic dominance constraints*, Math. Program., 114 (2008), pp. 249–275.
- [26] G. RUDOLF AND A. RUSZCZYŃSKI, *Optimization problems with second order stochastic dominance constraints: duality, compact formulations, and cut generation methods*. RUTCOR Research Report, Rutgers University, Piscataway, NJ, 2007.
- [27] A. SHAPIRO, *Asymptotic behavior of optimal solutions in stochastic programming*, Math. Oper. Res., 18 (1993), pp. 829–845.
- [28] W. WANG AND S. AHMED, *Sample average approximation of expected value constrained stochastic programs*. Preprint available at www.optimization-online.org, 2007.
- [29] G. WHITMORE AND M. FINDLAY, eds., *Stochastic Dominance*, D.C. Heath and Company, Lexington, MA, 1978.