An Alternating Direction Method for Chance-Constrained Optimization Problems with Discrete Distributions

Xiaodi Bai
Department of Management Science, School of Management, Fudan University, Shanghai 200433, P. R. China, xdbai@fudan.edu.cn

Jie Sun
Department of Decision Sciences and Risk Management Institute, National University of Singapore, Singapore 119245, jsun@nus.edu.sg

Xiaoling Sun
Department of Management Science, School of Management, Fudan University, Shanghai 200433, P. R. China, xls@fudan.edu.cn

Xiaojin Zheng
School of Economics and Management, Tongji University, Shanghai 200092, P. R. China, xjzheng@tongji.edu.cn

We consider a chance-constrained optimization problem (CCOP), where the random variables follow finite discrete distributions. The problem is in general nonconvex and can be reformulated as a mixed-integer program. By exploiting the special structure of the probabilistic constraint, we propose an alternating direction method for finding suboptimal solutions of CCOP. At each iteration, this method solves a convex programming subproblem and a 0-1 knapsack subproblem, which can be computed in quasi-linear time in the case of equal probabilities. We establish the convergence of the method to a first-order stationary point under certain mild conditions. Preliminary computational results are reported for VaR-constrained portfolio selection problems and chance-constrained transportation problems. Numerical results show that the proposed method is promising for finding solutions of good quality and compares favorably with CPLEX and other existing approximation methods, especially for large-size problems.

Key words: Chance-constrained optimization problem; finite discrete distribution; alternating direction method; augmented Lagrangian decomposition; first-order stationary point

1. Introduction

Consider the following chance-constrained optimization problem:

$$\begin{aligned} \text{(P)} \qquad & \min \ f(x) \\ & \text{s.t. } \mathbb{P}\{g(x,\xi) \geq 0\} \geq 1-\alpha, \\ & \quad\ x \in X, \end{aligned}$$

where $f: \Re^n \to \Re$ is a continuously differentiable convex function, $g = (g_1, \ldots, g_m): \Re^n \times \Re^s \to \Re^m$, $g_j(x,\xi)$ $(j = 1, \ldots, m)$ are continuously differentiable concave functions of x, ξ is a random vector taking values in $\Re^s$, $\alpha \in (0, 0.5)$ is a given risk level, $\mathbb{P}$ denotes the probability, and X is a nonempty compact convex set in $\Re^n$. Throughout the paper, we assume that problem (P) is feasible. Problem (P) can be viewed as a stochastic version of the conventional constrained convex optimization problem. The probabilistic constraint in (P) is also called *joint chance constraints*, which allows violation of the constraints $g(x,\xi) \geq 0$ with a small probability α. It is a reasonable relaxation of requiring $g(x,\xi) \geq 0$ to be held for all possible realizations of ξ, which could be very expensive or impossible in many situations.

Chance-constrained or probabilistically constrained optimization problems were first introduced in Charnes and Cooper (1959) and have been studied extensively in the stochastic optimization literature. Readers are referred to Prékopa (2003) and Shapiro et al. (2009) for comprehensive reviews on chance-constrained optimization problems. One of the major difficulties in solving problem (P) is the nonconvexity of the probabilistic constraint (see Prékopa (2003); Henrion and Strugarek (2008); Lagoa et al. (2005)). To circumvent this difficulty, several approximation approaches have been proposed to build tractable convex approximations to (P). Nemirovski and Shapiro (2006a) proposed a class of general convex safe (conservative) approximations which include the CVaR approximation of Rockafellar and Uryasev (2000) and the Bernstein approximation as special cases. Hong et al. (2011) developed a novel sequential convex approximation to (P) using a DC (difference of two convex functions) representation of the nonconvex probability function. Scenario approximations were discussed in Calafiore and Campi (2005, 2006) and Nemirovski and Shapiro (2006b). Ruszczyński (2002) showed that problem (P) can be reformulated as a mixed-integer convex programming under finite discrete distribution and developed a general iterative method using valid inequalities. Recently, Luedtke (2013) proposed a branch-and-cut decomposition algorithm for chance-constrained mathematical program with finite support.

In this paper, we focus on a special case of problem (P) that satisfies the following two assumptions:

(A1) There are only finitely many realizations (scenarios) of ξ, i.e., $\mathbb{P}(\xi = \xi^i) = p_i$, $i = 1, \ldots, N$, $\sum_{i=1}^N p_i = 1$.

(A2) The constraint functions $g_j(x, \xi)$ $(j = 1, \ldots, m)$ are affine in x.

Assumption A1 arises frequently in applications, either directly, or as empirical approximations of the underlying distribution. In many cases, ξ^i $(i = 1, \ldots, N)$ are independent observations of ξ. Two notable examples that satisfy Assumption A2 are Value-at-Risk (VaR) constrained portfolio selection problem and chance-constrained linear programming problem.

The VaR constraint in portfolio selection and risk management has the following form:

$$\mathbb{P}(\xi^T x - R \geq 0) \geq 1 - \alpha,$$

where ξ is the random vector of returns of n risky assets, $x \in \Re^n$ is the vector of portfolio weights and R is a prescribed minimal level of return. VaR-constrained portfolio selection problems have attracted much attention in recent years due to the importance of VaR as a popular risk measure in risk management. Alexander and Baptista (2004) and Alexander et al. (2007) investigated the VaR constrained mean-variance models and analyzed the effects of introducing a VaR constraint when the returns follow normal distributions. Bonami and Lejeune (2009) considered the mean-variance model with both integer constraints and probabilistic constraint, which can be reduced to a second-order cone constraint under certain continuous probability distributions or approximated by a second-order cone constraint using Chebychev's inequality. Gaivoronski and Pflug (2005) proposed a smoothing function to approximate the VaR function when ξ has a finite discrete distribution. Benati and Rizzi (2007) considered mean-VaR portfolio selection models under finite discrete distribution and presented a mixed integer programming reformulation. Recently, Zheng et al. (2012) derived a new mixed-integer quadratic programming reformulation for this model by using Lagrangian decomposition technique and semidefinite programming (SDP) relaxations.

Chance-constrained linear programming (CCLP) with the following constraint has received considerable attention in the literature:

$$\mathbb{P}(Tx \geq \eta) \geq 1 - \alpha,$$

where T is an $m \times n$ random matrix and η is a random vector in $\Re^m$. Cheng and Lisser (2012) and van Ackooij et al. (2011) proposed efficient algorithms for CCLP when η is deterministic and T is a random matrix with continuous distributions; for instance, each row of T is normally distributed. Chen et al. (2010) studied various tractable convex relaxations of this problem. Extensive research has been carried out in the literature for CCLP when the matrix T is deterministic and the right-hand side vector η is random. Several authors considered mixed-integer linear programming (MILP) reformulations for CCLP when η has finite discrete distribution. By identifying p-efficient points, Prékopa (1990) constructed improved MILP reformulations for CCLP. Lejeune and Noyan (2010) discussed how to generate p-efficient points using mathematical programming approaches. Dentcheva et al. (2000) introduced r-concave distribution functions of discrete random variables and derived lower and upper bounds of CCLP. Branch-and-bound methods based on efficient convex relaxations were developed in Beraldi and Ruszczyński (2002a,b) for CCLP. Luedtke et al. (2010) proposed strengthened MILP reformulations of CCLP using strong valid inequalities. Sen (1992) presented methods of disjunctive programming to approximate the convex hull of the feasible region. Vielma et al. (2012) proposed two new formulations for probabilistic constraints based on extended disjunctive formulations for the case when T is an identity matrix. Cheon et al. (2006) proposed a branch-reduce-cut algorithm based on domain reduction and linear programming relaxation for CCLP with a general distribution.

As the problem (P) is in general nonconvex, exact solution methods are only suitable for small-size problems. For large-size problems, it is natural to consider local or approximate methods which do not guarantee the global optimality but are capable of finding a suboptimal solution of (P) in reasonable computing time. The purpose of this paper is to present an efficient local method for problem (P) under assumptions A1 and A2. The method was motivated by the special structure of the probabilistic constraint under finite discrete distribution, namely, the augmented Lagrangian decomposition formulation of the mixed-integer program reformulation of (P) can be reduced to a convex subproblem or a 0-1 linear knapsack subproblem if certain variables are fixed. Furthermore, we can show that the 0-1 linear knapsack subproblem can be solved in quasi-linear time $O(N\log(N))$ in the case of equal probabilities. This suggests us to use alternating direction method to the augmented Lagrangian decomposition formulation in order to solve problem (P). Under certain mild conditions, we establish the convergence of the method to a first-order stationary point of (P). To evaluate the performance of our method, we conduct computational experiments on test problems of VaR-constrained portfolio selection problem and constrained-constrained transportation problem with random supply and demand. Numerical results show that our method is capable of finding suboptimal solutions of good quality. We also compare the proposed method with CPLEX and two approximate methods for (P): CVaR approximation and the DC method in Hong et al. (2011). Comparison results indicate that the proposed alternating direction method is advantageous in terms of the trade-off between the quality of the solutions and the computing time.

The remainder of the paper is organized as follows. In Section 2, we derive the augmented Lagrangian decomposition formulation of (P) and discuss the two subproblems obtained from fixing some variables. In Section 3, we describe the alternating direction method and prove its convergence to a first-order stationary point of (P). We conduct computational experiments in Section 4 on test problems from the VaR-constrained portfolio selection problem and the chance-constrained transportation problem with random supply and demand. Finally, we give a few concluding remarks in Section 5.

2. Augmented Lagrangian Decomposition Formulation and Subproblems

In this section, we first derive an augmented Lagrangian decomposition formulation for the mixed-integer programming reformulation of (P). We then discuss the subproblems obtained by fixing certain variables in the formulation.

Let $d_i \in \Re^m$ be a lower bound vector of $g(x, \xi^i)$ over X for $i = 1, \ldots, N$. Since X is compact and $g(x, \xi^i)$ is affine in x, d_i is finite for $i = 1, \ldots, N$. Without loss of generality, we assume that $d_i \leq 0$. By introducing a binary variable $z_i \in \{0, 1\}$ for each scenario ξ^i, $i = 1, \ldots, N$, the probabilistic constraint $\mathbb{P}\{g(x, \xi) \geq 0\} \geq 1 - \alpha$ can be expressed as

$$g(x, \xi^i) \geq z_i d_i,\ i = 1, \ldots, N,\ \sum_{i=1}^{N} p_i z_i \leq \alpha,\ z \in \{0, 1\}^N.$$

Hence, problem (P) can be reformulated as the following equivalent mixed integer programming (see Ruszczyński (2002)):

$$(\mathrm{MIP}_0) \qquad \begin{aligned} &\min f(x) \\ &\text{s.t. } g(x, \xi^i) \geq z_i d_i,\ i = 1, \ldots, N, \end{aligned}$$

$$\sum_{i=1}^{N} p_i z_i \le \alpha,\ z \in \{0,1\}^N,$$
$$x \in X.$$

Introducing $y_i = g(x, \xi^i)$ $(i = 1, \ldots, N)$, we can rewrite (MIP$_0$) as

$$
\begin{aligned}
(\text{MIP}_1) \qquad & \min f(x) \\
& \text{s.t. } y_i = g(x, \xi^i),\ i = 1, \ldots, N, \\
& \quad y_i \ge z_i d_i,\ i = 1, \ldots, N, \\
& \quad \sum_{i=1}^{N} p_i z_i \le \alpha,\ z \in \{0,1\}^N, \\
& \quad x \in X.
\end{aligned}
$$

The constraints $y_i = g(x, \xi^i)$ $(i = 1, \ldots, N)$ can be viewed as *link* constraints, which can be enforced approximately by adding a Lagrangian term and a quadratic penalty term in the objective function. This leads to the following augmented Lagrangian function for (MIP$_1$):

$$\mathcal{L}(x, y, \lambda) := f(x) + \sum_{i=1}^{N} \lambda_i^T [y_i - g(x, \xi^i)] + \frac{\rho}{2} \sum_{i=1}^{N} \|y_i - g(x, \xi^i)\|^2, \tag{1}$$

where each $\lambda_i \in \Re^m$ is a multiplier vector and $\rho > 0$ is a penalty parameter. The resulting augmented Lagrangian decomposition formulation is

$$
\begin{aligned}
& \min\ \mathcal{L}(x, y, \lambda) \\
& \text{s.t. } (y, z) \in \Omega, \\
& \qquad x \in X,
\end{aligned}
\tag{2}
$$

where

$$\Omega = \{(y, z) \in \Re^{m \times N} \times \{0,1\}^N \mid y_i \ge z_i d_i,\ \sum_{i=1}^{N} p_i z_i \le \alpha,\ i = 1, \ldots, N\}. \tag{3}$$

We first notice that for given $\lambda = \bar{\lambda}$, when $(y, z) = (\bar{y}, \bar{z}) \in \Omega$ is fixed, (2) becomes

$$
\begin{aligned}
& \min\ \mathcal{L}(x, \bar{y}, \bar{\lambda}) \\
& \text{s.t. } x \in X.
\end{aligned}
\tag{4}
$$

Since $f(x)$ is a convex function and $g_j(x, \xi^i)$ $(j = 1, \ldots, m)$ are affine functions of x for each i, the objective function of (4) is a convex function of x. Thus, (4) is a convex programming problem of variables x over the compact convex set X.

Next, we consider the subproblem of (2) when $x = \bar{x} \in X$ is fixed. For given $\lambda = \bar{\lambda}$, problem (2) becomes

$$\begin{aligned} &\min\ \mathcal{L}(\bar{x}, y, \bar{\lambda}) \\ &\text{s.t. } (y, z) \in \Omega, \end{aligned} \tag{5}$$

where Ω is defined in (3). Let $\bar{\delta}$ denote the optimal value of problem (5). Then

$$\bar{\delta} \;=\; f(\bar{x}) - \tau + \frac{\rho}{2}\min\{\sum_{i=1}^{N} \|y_i - w_i\|^2 \mid (y, z) \in \Omega\}, \tag{6}$$

where $w_i = g(\bar{x}, \xi^i) - \bar{\lambda}_i/\rho \in \Re^m$ and $\tau = \frac{1}{2\rho}\sum_{i=1}^{N} \|\bar{\lambda}_i\|^2$ is a constant. Let

$$q_i = \min_{y_i \geq 0} \|y_i - w_i\|^2,\ r_i = \min_{y_i \geq d_i} \|y_i - w_i\|^2, \quad i = 1, \ldots, N. \tag{7}$$

It follows from (3) and (6) that

$$\begin{aligned} \bar{\delta} \;&=\; f(\bar{x}) - \tau + \frac{\rho}{2}\min\left\{\sum_{i=1}^{N} r_i z_i + q_i(1 - z_i) \mid \sum_{i=1}^{N} p_i z_i \leq \alpha,\ z \in \{0, 1\}^N\right\} \\ &=\; \bar{f} + \frac{\rho}{2}\min\left\{\sum_{i=1}^{N} (r_i - q_i) z_i \mid \sum_{i=1}^{N} p_i z_i \leq \alpha,\ z \in \{0, 1\}^N\right\}, \end{aligned} \tag{8}$$

where $\bar{f} = f(\bar{x}) - \tau + \frac{\rho}{2}\sum_{i=1}^{N} q_i$. Since $d_i \leq 0$, we have $r_i \leq q_i$ for each i. Thus, the minimization problem in (8) is equivalent to the following 0-1 linear knapsack problem.

$$\max\left\{\sum_{i=1}^{N} (q_i - r_i) z_i \mid \sum_{i=1}^{N} p_i z_i \leq \alpha,\ z \in \{0, 1\}^N\right\}. \tag{9}$$

Although (9) is still NP-hard for general finite discrete distribution, it is much more tractable than the original problem as it can be solved by dynamic programming when p_i $(i = 1, \ldots, N)$ are rational numbers.

We now consider an important case of the 0-1 linear knapsack problem (9) when $p_i = 1/N$ for $i = 1, \ldots, N$. Let $K = \lfloor N\alpha \rfloor$, where $\lfloor N\alpha \rfloor$ is the maximum integer number less than or equal to $N\alpha$. The constraint in the 0-1 linear knapsack problem (9) reduces to $\sum_{i=1}^{N} z_i \leq K$. Thus, (9) is equivalent to the sum of the K largest entries of $\{q_i - r_i\}_{i=1}^{N}$, which can be computed in $O(N \log(N))$ by ranking the sequence $\{q_i - r_i\}_{i=1}^{N}$.

Finally, we see that q_i and r_i defined in (7) can be calculated in the following way:

$$q_i = \sum_{j=1}^{m} q_{ij},\ r_i = \sum_{j=1}^{m} r_{ij}, \quad i = 1, \ldots, N,$$

where

$$q_{ij} = \min_{y_{ij} \geq 0} (y_{ij} - w_{ij})^2 = \begin{cases} w_{ij}^2, & w_{ij} < 0, \\ 0, & w_{ij} \geq 0, \end{cases} \tag{10}$$

$$r_{ij} = \min_{y_{ij} \geq d_{ij}} (y_{ij} - w_{ij})^2 = \begin{cases} (d_{ij} - w_{ij})^2, & w_{ij} < d_{ij}, \\ 0, & w_{ij} \geq d_{ij}. \end{cases} \tag{11}$$

Therefore, the optimal y_i in the subproblem (5) can be easily determined by using (7), (10) and (11) via the optimal solution of (9).

In summary, we have shown that the two subproblems obtained by fixing x or (y, z) in the augmented Lagrangian decomposition formulation (2) can be solved efficiently.

3. Alternating Direction Method

In this section, we describe an *alternating direction method* for (P) and prove its convergence to a first-order stationary point of (P).

Alternating direction methods, also known as block coordinate descent methods, have been successfully applied to structured convex programming problems and some nonconvex optimization problems arising from image processing and matrix optimization (see, e.g., Goldstein (2009); Yin et al. (2008); He et al. (2012); Shen et al. (2013); Sun and Zhang (2010); Zhang et al. (2013)). Our method was stimulated by the penalty decomposition methods in Lu and Zhang (2011, 2013) for rank minimization and sparse optimization, where the difficulties arise from the rank constraint of matrices or the sparsity constraint of vectors.

The idea of our method is to solve the two subproblems (4) and (5) alternatively while updating the Lagrangian multipliers so as to enforce the least square penalty term to diminish iteratively. By doing so, we are able to "isolate" the probabilistic constraint from other deterministic constraints, thus alleviating the combinatorial difficulty of the probabilistic constraint inherent in the original problem. A detailed description of the method is as follows.

Algorithm 1 (Alternating Direction Method for (P))

Step 0. Choose tolerance parameter $\epsilon \geq 0$, multiplier vector λ^0, penalty parameter $\rho > 0$ and step-size $\kappa > 0$. Choose $x^0 \in X$. Set the iteration counter $k = 0$.

Step 1. Solve the subproblem problem (5) with $\bar{x} = x^k$ to obtain an optimal solution (y^{k+1}, z^{k+1}).

Step 2. Solve the convex programming subproblem (4) with $(\bar{y}, \bar{z}) = (y^{k+1}, z^{k+1})$ to obtain an optimal solution x^{k+1}.

Step 3. If $\sum_{i=1}^{N} \|y_i^{k+1} - g(x^{k+1}, \xi^i)\|^2 \leq \epsilon$, stop, x^{k+1} as an approximate optimal solution. Otherwise, go to Step 4.

Step 4. Update the multipliers by

$$\lambda_i^{k+1} = \lambda_i^k + \kappa\rho[y_i^{k+1} - g(x^{k+1}, \xi^i)],\ i = 1, \ldots, N. \tag{12}$$

Set $k := k + 1$, and go to Step 1.

Before establishing the convergence properties of Algorithm 1, we need to give the local optimality conditions for (P). In the sequel, we make the following assumption:

(A3) The probabilities of each scenario $\xi = \xi^i$ $(i = 1, \ldots, N)$ are equal, i.e., $p_i = \mathbb{P}(\xi = \xi^i) = 1/N$ for $i = 1, \ldots, N$.

For $y = (y_1, \ldots, y_N) \in \Re^{m \times N}$, where $y_i = (y_{i1}, \ldots, y_{im})^T$, define $s(y) = (s(y)_1, \ldots, s(y)_N)^T$, where

$$s(y)_i = \min_{j=1,\ldots,m} y_{ij}, \qquad i = 1, \ldots, N.$$

Let $G(x) = [g(x, \xi^1), \ldots, g(x, \xi^N)] \in \Re^{m \times N}$. Then, $g(x, \xi^i) \geq 0$ if and only if $s(G(x))_i \geq 0$. Since each g_j is a linear function of x, $s(G(x))_i$ $(i = 1, \ldots, N)$ are concave functions of x. Let $a_{[k]}$ denote the k-th smallest entry of a vector $a \in \Re^N$. Then, by Assumption A3, we have

$$\mathbb{P}\{g(x, \xi) \geq 0\} \geq 1 - \alpha \Leftrightarrow s(G(x))_{[K]} \geq 0, \tag{13}$$

where $K = \lfloor N\alpha \rfloor$. For convenience, we denote in the sequel that $\chi(x) := s(G(x))_{[K]}$. Therefore, problem (P) can be rewritten as

$$\begin{aligned} &\min\ f(x) \\ &\text{s.t. } \chi(x) \geq 0, \\ &\qquad x \in X. \end{aligned} \tag{14}$$

Lemma 1 *The function $\chi(x)$ is locally Lipschitz on $\Re^n$.*

Proof. Note that each $\chi(x) = s(G(x))_{[K]}$ is a composite function of $u = s(G(x))$ and $u_{[K]}$. Since $s(G(x))_i = \min_{j=1,\ldots,m} g_j(x, \xi^i)$ is concave and hence locally Lipschitz on $\Re^n$, it suffices to show that $u_{[K]}$ is a locally Lipschitz function of $u \in \Re^N$ (see Clarke (1983)). We see that $u_{[K]}$ can be expressed as

$$u_{[K]} = \sum_{j=K}^{N} u_{[j]} - \sum_{j=K+1}^{N} u_{[j]}. \tag{15}$$

Let $\psi_t(u) = \sum_{j=t}^{N} u_{[j]}$. Then, $u_{[K]} = \psi_K(u) - \psi_{K+1}(u)$. Since $\psi_t(u)$ is the sum of the $N-t+1$ largest components of u, it is a convex function of u (see, e.g., Example 3.6 in Boyd and Vandenberghe (2004)). Thus, $u_{[K]}$ is the different of two convex functions and hence is locally Lipschitz function since convex function and concave functions are locally Lipschitz and the sum of two locally Lipschitz functions is a locally Lipschitz function. □

By Lemma 1, the Clarke generalized gradient of $\chi(x)$ exists for any $x \in \Re^n$. Moreover, $\chi(x)$ can be viewed as a composite of $y = G(x)$ and $s(y)_{[K]}$. By the chain rule (see Clarke (1983)), we have

$$\partial\chi(x) = \text{conv}\left\{ \sum_{i=1}^{N} \nabla g(x, \xi^i)\theta_i : \; \theta_i \in \partial_{y_i}(s(y)_{[K]}) \mid_{y=G(x)} \right\}, \tag{16}$$

where $\nabla g(x, \xi^i) \in \Re^{n\times m}$ is the Jacobian matrix of $g(x, \xi^i)$ and $\partial_{y_i}(s(y)_{[K]}) \mid_{y=G(x)}$ denotes the partial generalized Clarke gradient of $s(y)_{[K]}$ at $y = G(x)$ with respect to $y_i \in \Re^m$, $i = 1, \ldots, N$.

We have the following first-order stationary condition for (14) (see Clarke (1983)): If x^* is a local minimizer of problem (14), then there exists $\pi^* \geq 0$ such that

$$0 \in \nabla f(x^*) - \pi^*\partial\chi(x^*) + \mathcal{N}_X(x^*), \tag{17}$$

$$\pi^*\chi(x^*) = 0, \tag{18}$$

where $\mathcal{N}_X(x^*)$ is the normal cone of X at x^*.

We are now ready to state the convergence theorem of Algorithm 1.

Theorem 1 *Let (x^*, y^*, λ^*) be any accumulation point of $\{(x^k, y^k, \lambda^k)\}$ generated by Algorithm 1. Assume that $\{\lambda^k\}$ is bounded and*

$$\sum_{k=1}^{\infty} \|\lambda^{k+1} - \lambda^k\|^2 < \infty. \tag{19}$$

Then, x^* satisfies the first-order stationary conditions (17)-(18).

Before proceeding to the proof of Algorithm 1, a few remarks are in order. The assumption on the boundedness of the multiplier vectors in Theorem 1 is a standard condition in the convergence analysis of augmented Lagrangian methods for nonconvex optimization problems (see, e.g., Bertsekas (1982); Luo et al. (2007)). Roughly speaking, the condition (19) is to ensure that the difference of multipliers between two iterations shrinks faster than $1/\sqrt{k}$ as the iteration proceeds. Similar conditions have been used in the convergence analysis of alternating direction methods for nonconvex optimization problems (see, e.g., Shen et al. (2013)).

We need the following lemma.

Lemma 2 *Under the assumptions of Theorem 1, it holds that*

$$\|x^k - x^{k+1}\| \to 0, \ \|y^k - y^{k+1}\| \to 0, \ k \to \infty. \tag{20}$$

Proof. Since x^{k+1} solves (4) at the k-th iteration and $x^k - x^{k+1}$ is a feasible direction with respect to X, we have

$$\nabla_x \mathcal{L}(x^{k+1}, y^{k+1}, \lambda^k)^T (x^k - x^{k+1}) \geq 0. \tag{21}$$

From (1), we see that $\mathcal{L}(x, y, \lambda)$ is a strongly convex function (with certain parameter $\bar{\rho}/2$) of variable x , we have

$$\begin{aligned} \mathcal{L}(x^k, y^{k+1}, \lambda^k) - \mathcal{L}(x^{k+1}, y^{k+1}, \lambda^k) \ &\geq \ \nabla_x \mathcal{L}(x^{k+1}, y^{k+1}, \lambda^k)^T (x^k - x^{k+1}) \\ &\quad + \frac{\bar{\rho}}{2}\|x^k - x^{k+1}\|^2. \end{aligned}$$

This together with (21) gives rise to

$$\mathcal{L}(x^k, y^{k+1}, \lambda^k) - \mathcal{L}(x^{k+1}, y^{k+1}, \lambda^k) \geq \frac{\bar{\rho}}{2}\|x^k - x^{k+1}\|^2. \tag{22}$$

On the other hand, since y^{k+1} solves (5) at the k-iteration and y^k is feasible to (5), we have

$$\mathcal{L}(x^k, y^k, \lambda^k) - \mathcal{L}(x^k, y^{k+1}, \lambda^k) \geq 0. \tag{23}$$

Moreover, by (1) and (12), we have

$$\begin{aligned} & \mathcal{L}(x^{k+1}, y^{k+1}, \lambda^k) - \mathcal{L}(x^{k+1}, y^{k+1}, \lambda^{k+1}) \\ = \ & \sum_{i=1}^{N} (\lambda_i^k - \lambda_i^{k+1})(y_i^{k+1} - g(x^{k+1}, \xi^i)) \\ = \ & -\frac{1}{\kappa\rho}\|\lambda^k - \lambda^{k+1}\|^2. \end{aligned} \tag{24}$$

Combining with (22)-(24), we obtain

$$\begin{aligned}
& \mathcal{L}(x^k, y^k, \lambda^k) - \mathcal{L}(x^{k+1}, y^{k+1}, \lambda^{k+1}) \\
= \ & \mathcal{L}(x^k, y^k, \lambda^k) - \mathcal{L}(x^k, y^{k+1}, \lambda^k) \\
& + \mathcal{L}(x^k, y^{k+1}, \lambda^k) - \mathcal{L}(x^{k+1}, y^{k+1}, \lambda^k) \\
& + \mathcal{L}(x^{k+1}, y^{k+1}, \lambda^k) - \mathcal{L}(x^{k+1}, y^{k+1}, \lambda^{k+1}) \\
\geq \ & \frac{\bar{\rho}}{2}\|x^k - x^{k+1}\|^2 - \frac{1}{\kappa\rho}\|\lambda^k - \lambda^{k+1}\|^2. \qquad (25)
\end{aligned}$$

Since $\{x^k\} \subset X$ and by assumption X and $\{\lambda^k\}$ are bounded, we deduce from (12) that $\{y^k\}$ is also bounded. Thus, $\{\mathcal{L}(x^k, y^k, \lambda^k)\}$ is bounded. Summing up both sides of (25) over all k yields

$$\frac{\bar{\rho}}{2}\sum_{k=1}^{\infty}\|x^k - x^{k+1}\|^2 - \frac{1}{\kappa\rho}\sum_{k=1}^{\infty}\|\lambda^k - \lambda^{k+1}\|^2 < \infty. \qquad (26)$$

By assumption, $\sum_{k=1}^{\infty}\|\lambda^k - \lambda^{k+1}\|^2 < \infty$, thus (26) implies that

$$\|x^k - x^{k+1}\| \to 0, \qquad k \to \infty. \qquad (27)$$

Since $\lambda^k - \lambda^{k+1} \to 0$ $(k \to \infty)$, it follows from (12) that

$$\|y^{k+1} - g(x^{k+1}, \xi^i)\| = \frac{1}{\kappa\rho}\|\lambda^k - \lambda^{k+1}\| \to 0, \qquad k \to \infty. \qquad (28)$$

This together with (27) and the continuity of $g(x, \xi^i)$ with respect to x implies that

$$\begin{aligned}
y^k - y^{k+1} &= [y^k - g(x^k, \xi^i)] + [g(x^k, \xi^i) - g(x^{k+1}, \xi^i)] \\
&\quad + [g(x^{k+1}, \xi^i) - y^{k+1}] \to 0, \qquad k \to \infty.
\end{aligned}$$

Therefore, (20) holds. □

Proof of Theorem 1. Let (x^*, y^*, λ^*) be an accumulation point of $\{(x^k, y^k, \lambda^k)\}$. Then, there exists a subsequence $\{(x^k, y^k, \lambda^k)\}_{k\in\mathcal{I}}$ that converges to (x^*, y^*, λ^*). We first note that under Assumption A3, $(y, z) \in \Omega$ is equivalent to $s(y)_{[K]} \geq 0$. Hence, the subproblem (5) with $\bar{x} = x^k \in X$ can be rewritten as

$$\begin{aligned}
& \min \ \mathcal{L}(x^k, y, \lambda^k) \qquad (29) \\
& \text{s.t. } s(y)_{[K]} \geq 0.
\end{aligned}$$

Since y^{k+1} solves (29), there exists $\pi^k \geq 0$ such that

$$\nabla_{y_i}\mathcal{L}(x^k, y^{k+1}, \lambda^k) - \pi^k \zeta_i^k = 0, \qquad i = 1, \ldots, N, \tag{30}$$

$$\pi^k s(y^{k+1})_{[K]} = 0, \tag{31}$$

where $\zeta_i^k \in \partial_{y_i}(s(y^{k+1})_{[K]})$ and

$$\nabla_{y_i}\mathcal{L}(x^k, y^{k+1}, \lambda^k) = \lambda_i^k + \bar{\rho}(y_i^{k+1} - g(x^k, \xi^i)). \tag{32}$$

It is easy to verify that $\partial_{y_i} s(y) = \text{conv}\{e_j \mid y_j = s(y),\ j = 1, \ldots, m\}$, where e_j is the j-th unit vector in $\Re^m$, and $\partial_i(u_{[K]}) = \text{conv}\{e_i :\ u_i = u_{[K]}\}$, where e_i is the i-th unit vector in $\Re^N$. Hence, the set $\partial_{y_i}(s(y^{k+1})_{[K]})$ is bounded away from zero. Consequently, there exists some constant $\delta_0 > 0$ such that $\|\zeta_i^k\| \geq \delta_0$ for all k and $i = 1, \ldots, N$. It then follows from (30) that $\{\pi^k\}_{k\in\mathcal{I}}$ is bounded. Also, since the Clarke generalized gradient is a lower semi-continuous set-valued mapping (see Proposition 2.4.4 in Clarke (1983)), $\{\zeta^k\}_{k\in\mathcal{I}}$ is bounded. Without loss of generality, we assume that $\pi^k \to \pi^*$ and $\zeta^k \to \zeta^*$ $(k \to \infty,\ k \in \mathcal{I})$. Talking limits in (30)-(32) for $k \in \mathcal{I}$ and using Lemma 2, we get

$$\lambda_i^* - \pi^* \zeta_i^* = 0, \qquad i = 1, \ldots, N, \tag{33}$$

$$\pi^* s(y^*)_{[K]} = 0, \tag{34}$$

where $\zeta_i^* \in \partial_{y_i}(s(y^*)_{[K]})$, $\pi^* \geq 0$, $y_i^* = g(x^*, \xi^i)$, $i = 1, \ldots, N$.

On the other hand, since x^{k+1} solves the subproblem (4) when $(\bar{y}, \bar{z}) = (y^{k+1}, z^{k+1})$, the first-order optimality condition of (4) is

$$0 \in \nabla_x \mathcal{L}(x^{k+1}, y^{k+1}, \lambda^k) + \mathcal{N}_X(x^{k+1}), \tag{35}$$

where

$$\nabla_x \mathcal{L}(x^{k+1}, y^{k+1}, \lambda^k) = \nabla f(x^{k+1}) - \sum_{i=1}^{N} \nabla g(x^{k+1}, \xi^i)[\lambda_i^k + \bar{\rho}(y_i^{k+1} - g(x^{k+1}, \xi^i)].$$

Since the normal cone of a compact set is a lower semi-continuous set-valued mapping (see Proposition 2.4.4 in Clarke (1983)), taking limits for $k \in \mathcal{I}$ on (35) and using Lemma 2 and (28), we obtain

$$0 \in \nabla f(x^*) - \sum_{i=1}^{N} \nabla g(x^*, \xi^i)\lambda_i^* + \mathcal{N}_X(x^*). \tag{36}$$

Note that $\chi(x^*) = s(G(x^*))_{[K]} = s(y^*)_{[K]}$. Combining (33), (34) and (36), and noting the chain rule (16), we conclude that x^* is a first-order stationary point satisfying conditions (17) and (18). □

4. Numerical Results

In this section, we conduct computational experiments to evaluate the performance of the alternating direction method (Algorithm 1). Two types of chance-constrained problems are considered in our test: the VaR-constrained portfolio selection problem and the chance-constrained transportation problem. The main purpose of our computational experiments is to test the capability of the proposed method for finding good quality solution. In particular, we compare the method against two existing approximate methods and CPLEX when applied to the mixed-integer program reformulation (MIP_0).

The numerical tests were implemented in Matlab and run on a PC (3.2G Hz, 8GB RAM). All linear and quadratic programming subproblems in Algorithm 1 and the mixed-integer linear and quadratic reformulations are solved by the QP solver and mixed-integer MIP or QCP solver in CPLEX 12.5 with Matlab interface. In our test, the parameters in Algorithm 1 are set as follows: the tolerance parameter is set as $\epsilon = 10^{-6}$; the step length $\kappa = 0.1$; the initial Lagrangian multiplier is set as $\lambda^0 = 0$, and the initial penalty ρ is set as 1 for portfolio selection problem and 0.01 for chance-constrained transportation problem, respectively. The maximum CPU time limit for CPLEX is set to 1800 seconds. For all the test problems, we set $p_i = 1/N$ for $i = 1, \ldots, N$. The data files of the test problems and the Matlab codes in our numerical test are available at: `http://my.gl.fudan.edu.cn/teacherhome/xlsun/ccop`.

4.1 CVaR approximation and DC method

We will compare Algorithm 1 with two approximate methods for chance-constrained optimization problems: the CVaR approximation and the DC method proposed by Hong et al. (2011).

The idea of using CVaR as a convex (conservative) approximation is due to Rockafellar and Uryasev (2000). Under assumption A1, the CVaR approximation of problem (P) can be expressed as the following convex programming problem:

$$
\begin{aligned}
(\mathrm{CVaR}) \qquad & \min f(x) \\
& \text{s.t. } t + \frac{1}{\alpha N}\sum_{i=1}^{N} y_i \le 0, \\
& \quad y_i \ge 0,\ y_i \ge -g_j(x, \xi^i) - t,\ i = 1, \ldots, N,\ j = 1, \ldots, m, \\
& \quad x \in X,\ t \in \mathbb{R}.
\end{aligned}
$$

The DC method for joint chance constraints is proposed by Hong et al. (2011). Note that $\mathbb{P}(g(x,\xi) \geq 0) \geq 1-\alpha$ is equivalent to

$$\mathbb{E}[1_{(0,+\infty)}(-s(x,\xi))] \leq \alpha,$$

where $s(x,\xi) = \min\{g_1(x,\xi),\dots,g_m(x,\xi)\}$ and 1_A denotes the indicator function of set A, i.e., $1_A(z) = 1$ if $z \in A$ and 0 if $z \notin A$. The idea of the DC method is to use a DC function to approximate the indicator function $1_{(0,+\infty)}(z)$ and consequently construct the following DC approximation to the probabilistic constraint in (P):

$$\inf_{t>0}\{\frac{1}{t}[h_1(x,t) - h_2(x)]\} \leq \alpha,$$

where $h_1(x,t) = \mathbb{E}[t - s(x,\xi)]^+$ and $h_2(x) = \mathbb{E}[-s(x,\xi)]^+$. Successive linearization algorithm are then used to generate a sequence of convex subproblems which converges to a KKT point of the DC approximation problem. See Hong et al. (2011) for a detailed description of the DC method.

4.2 Portfolio selection problem

In this subsection, we consider VaR-constrained mean-variance portfolio selection problem of the following form:

$$\begin{aligned} \text{(VaR-MV)} \qquad & \min\ x^T\Sigma x \\ & \text{s.t. } \mathbb{P}(\xi^T x \geq R) \geq 1-\alpha, \\ & \qquad e^T x = 1,\ 0 \leq x \leq u, \end{aligned}$$

where Σ is the covariance matrix, e is the all-one column vector and $0 < u \leq e$ is an upper bound vector for x. In our test, we set $u = 0.5e$ and $\alpha = 0.05$ or 0.10, respectively. For this problem, (MIP$_0$) is a mixed-integer quadratic programming problem.

To build the test problems for (VaR-MV), we use both real market data and simulated data to construct the asset return samples ξ and the covariance matrix Σ.

- **Real data set**. We use 2646 daily return data of 449 stocks included in Standard & Poor's 500 index between December 2001 and February 2012. The covariance matrix Σ is calculated by the sample covariance using the 2646 historical data. We generate the test problems with n ranged from 100 to 400. For each n, we generate 10 instances by randomly selecting n stocks from the 449 stocks and $N = 3n$ samples ξ^i $(i = 1,\dots,N)$

Table 1: Comparison results for (VaR-MV) with different number of stocks using real data set

		CVaR		DC		ADM			CPLEX	
n	α	fval	time	fval	time	fval	time	iter	fval	time
100	0.05	1.9310	0.03	1.2502	0.38	1.1968	12.9	1150	1.1847	90.4
150	0.05	1.9070	0.07	1.2141	1.0	1.1666	14.4	1122	1.1552	1491.4
200	0.05	1.7349	0.14	1.1728	2.4	1.1339	17.4	1129	1.1205	1800
250	0.05	1.6931	0.24	1.1306	4.6	1.0817	22.0	1143	1.0744	1800
300	0.05	1.6376	0.39	1.1057	7.5	1.0613	24.8	1135	1.0590	1800
350	0.05	1.5832	0.51	1.0887	12.1	1.0329	30.6	1138	1.0327	1800
400	0.05	1.5876	0.74	1.0879	18.2	1.0270	37.0	1146	1.0435	1800
100	0.1	2.0867	0.04	1.1623	0.70	1.1321	11.3	1033	1.1229	34.2
150	0.1	1.8783	0.08	1.1461	1.9	1.1165	13.0	1034	1.1083	1329.6
200	0.1	1.5822	0.14	1.0299	4.0	1.0067	15.0	1009	0.9991	1663.0
250	0.1	1.5043	0.23	0.9863	7.0	0.9640	17.8	950	0.9603	1800
300	0.1	1.4250	0.36	0.9865	12.1	0.9655	21.7	995	0.9598	1800
350	0.1	1.4166	0.55	0.9726	19.1	0.9486	26.4	992	0.9445	1800
400	0.1	1.2794	0.64	0.9375	28.7	0.9166	31.6	983	0.9156	1800

from the 2646 daily return data. For each instance, the prescribed annualized return level is set equal to 5%, so the daily return level is $R = 5\%/250 = 0.02\%$.

- **Simulation data set**. To evaluate the performance of the method for large-size problems, we also construct test problems of (VaR-MV) with n up to 1000 and $N = 3000$, using simulation data generated in a similar fashion as in Nemirovski and Shapiro (2006a). The first asset has deterministic return and the returns of the remaining $n-1$ assets are random variables with expectations $\mathbb{E}[r_i] = \bar{r}_i$, where the nominal profits $\bar{r}_i$ vary in [0, 0.1]. The random returns of assets are generated by the following factor model:

$$r_i = \eta_i + \sum_{l=1}^{8} \gamma_{il}\zeta_l, \qquad i = 2, \dots, n, \tag{37}$$

where $\log \eta_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, $\log \zeta_l \sim \mathcal{N}(0, 0.01)$, $\mu_i = \sigma_i$, and $\gamma_{il} \geq 0$ are deterministic coefficients. The coefficients γ_{il} and the parameters μ_i are chosen in such a way that $\mathbb{E}[\sum_{l=1}^{8} \gamma_{il}\zeta_l] = \bar{r}_i/2$ and $E[\eta_i] = \bar{r}_i/2$ for all i. For each n, we construct 10 instances by randomly generating N samples of $\xi^i = (r_1, \dots, r_n)^T$ by the factor model (37). The covariance matrix Σ is calculated using (37). The return level is set as $R = 6\%$.

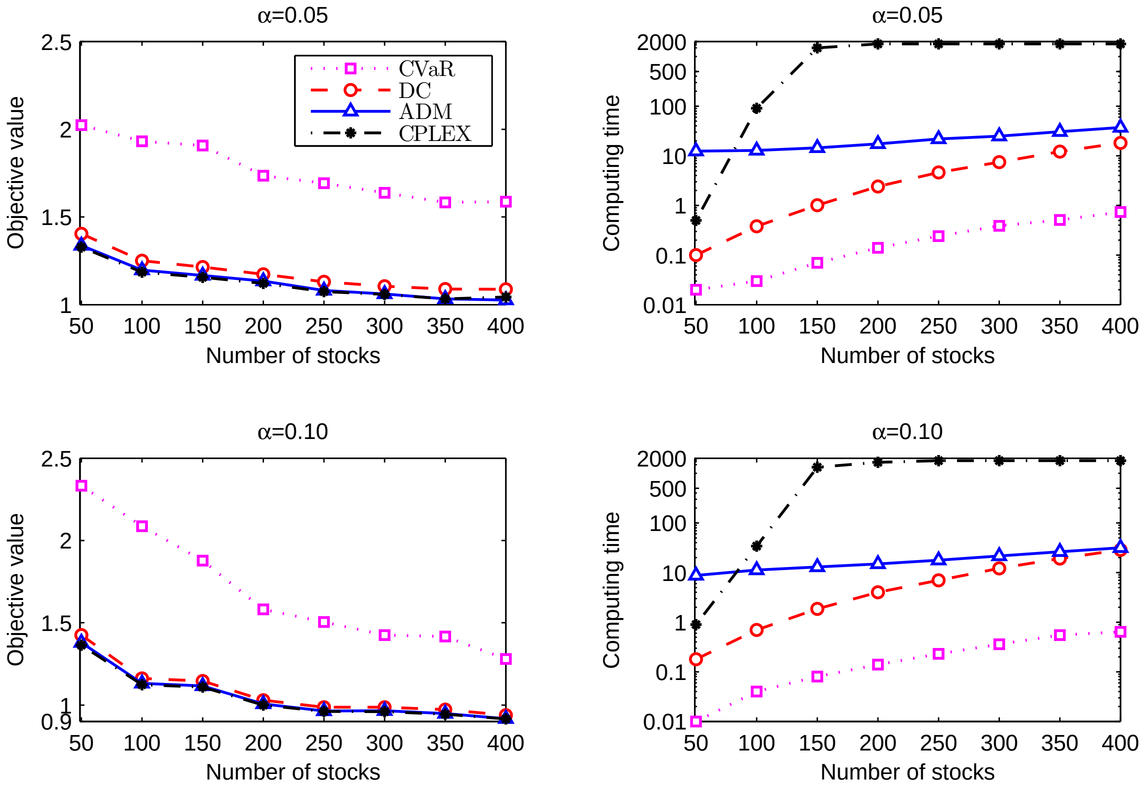

Figure 1: Objective value versus computing time for (VaR-MV) with different number of stocks using real data set.

Table 1 summarizes the comparison results for test problem (VaR-MV) with n up to 400 and $N = 3n$ using real data set. We use the following notation in Table 1:

- “CVaR”: the CVaR approximation method;
- “DC”: the DC method;
- “ADM”: the alternating direction method (Algorithm 1);
- “CPLEX”: the CPLEX mixed-integer program solver applied to (MIP$_0$);
- “fval”: the average objective value for the 10 test problems;
- “time”: the average computing time in seconds for the 10 test problems;
- “iter”: the average number of iterations of ADM for the 10 test problems.

From Tables 1, we see that ADM, the DC method and CPLEX can find solutions with significantly better objective values than those found by the CVaR approximation. While

Table 2: Comparison results for (VaR-MV) with different number of stocks using simulation data set

		CVaR		DC		ADM			CPLEX	
n	α	fval	time	fval	time	fval	time	iter	fval	time
400	0.05	1.8020	4.1	1.7467	187.6	1.7466	70.5	362	1.8018	1800
500	0.05	1.5381	5.9	1.4941	251.5	1.4941	115.9	360	1.5404	1800
600	0.05	1.3623	7.8	1.3265	327.4	1.3267	179.9	359	1.3608	1800
700	0.05	1.2437	10.1	1.2117	403.2	1.2118	275.7	359	1.2464	1800
800	0.05	1.1505	12.8	1.1213	513.7	1.1214	391.4	355	1.1491	1800
900	0.05	1.0825	15.8	1.0551	626.0	1.0550	510.0	332	1.0782	1800
1000	0.05	1.0289	19.5	1.0037	730.7	1.0036	658.9	322	1.0282	1800
400	0.1	1.7541	5.9	1.6982	347.2	1.6982	70.1	361	1.7652	1800
500	0.1	1.5102	8.0	1.4643	454.4	1.4643	119.1	359	1.5238	1800
600	0.1	1.3425	10.4	1.3036	580.7	1.3038	181.3	359	1.3616	1800
700	0.1	1.2270	13.1	1.1931	707.4	1.1932	278.0	359	1.2461	1800
800	0.1	1.1387	16.7	1.1076	874.4	1.1076	398.9	355	1.1568	1800
900	0.1	1.0703	20.4	1.0417	1040.7	1.0415	498.2	324	1.0868	1800
1000	0.1	1.0149	26.6	0.9881	1270.3	0.9880	645.5	313	1.0350	1800

ADM can achieve better objective values than the DC method, it spends slightly more computing time than the DC method. We also observe that CPLEX can find slightly better solutions than ADM at the price of using much more time than ADM. As is expected, the computing time of the CVaR approximation is much less than the other three methods because it only solves a single convex quadratic programming problem. We further plots the objective value versus computing time in Figure 1, from which we can see that the computing time for ADM and the DC method tends to be close as n increases.

To see the performance of different methods for large-size problems, we report in Table 2 the comparison results for (VaR-MV) with n up to 1000 and $N = 3000$ using simulation data set. For these large-size test problems, ADM appears to outperform all other methods in terms of the objective values and the computing time. The objective values achieved by ADM and the DC method are similar while ADM gives slightly better solutions for almost cases in less computing time. The number of iterations of ADM does not vary dramatically as n increases. Figure 2 further illustrates the trend of objective value versus computing time. We see that the performance of ADM is quite robust for large-size problems. We observe also that for all large-size instances, CPLEX was terminated when reaching the CPU time limit (1800 seconds).

In summary, the above comparison results suggest that the proposed alternating direction

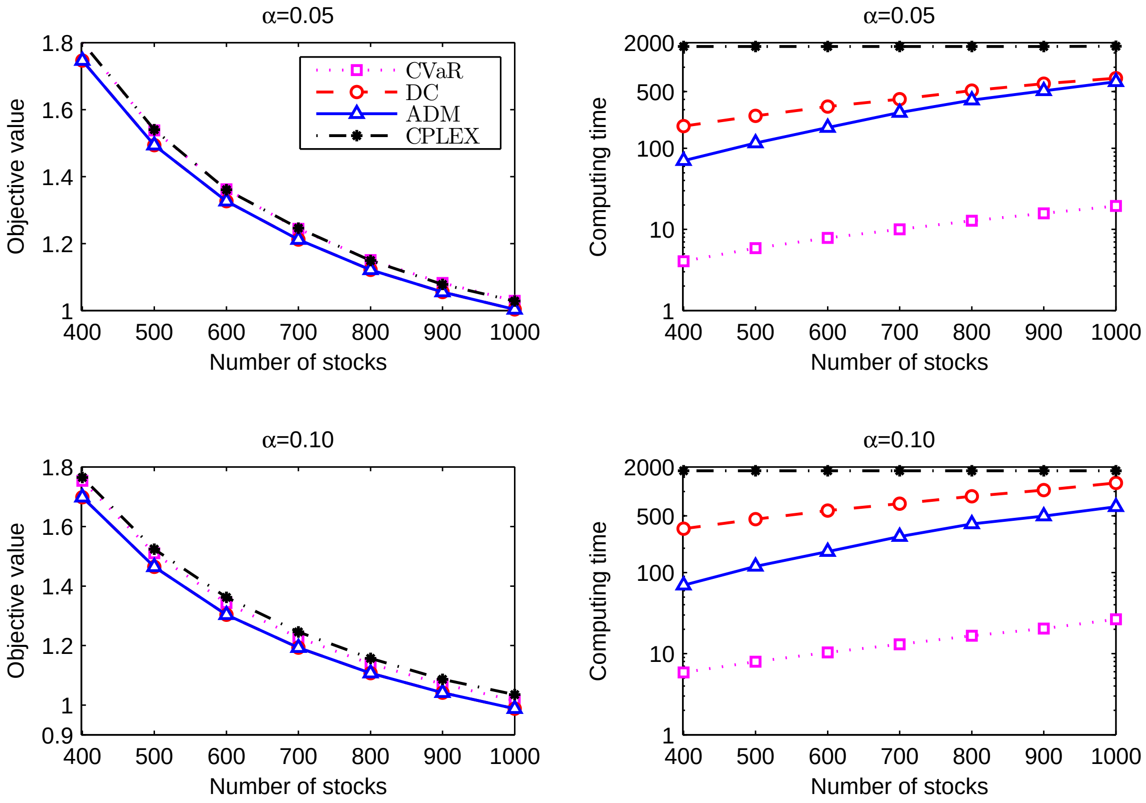

Figure 2: Comparison results for (VaR-MV) with different number of stocks using simulation data set.

method is efficient in finding solutions of good quality for large-size problems of (VaR-MV) and is advantageous over the two approximate methods and CPLEX in terms of the trade-off between the solution quality and the computing time.

4.3 Chance-constrained transportation problem

In this subsection, we consider a probabilistic version of the conventional transportation problem. Suppose there are n suppliers and m customers. The transportation cost for shipping one unit product from supplier i to customer j is c_{ij}. Suppose that the customers' demands d and the suppliers' capacity ξ are *both random*. The problem is how to make decision on the shipment strategy such that the expected production transportation cost is minimized without the exact knowledge about the supplier capacity and the customer demands.

It is worth pointing out most of the solutions methods in the literature for chance-constrained linear programming are only applicable to the cases when either the demand d

or the capacity ξ is random, but not both. For instance, the solution methods based on p-efficient points (see, e.g., Prékopa (1990); Lejeune and Noyan (2010); Dentcheva et al. (2000)) and the strengthened MILP reformulations (see, e.g., Luedtke et al. (2010); Vielma et al. (2012)) are only applicable to stochastic transportation problem with random right-hand side (demand).

The chance-constrained transportation problem in our test has the following form:

$$\begin{aligned}
\text{(STP)} \qquad & \min \quad \sum_{i=1}^{n} \mathbb{E}[\xi_i] \sum_{j=1}^{m} c_{ij} x_{ij} \\
& \text{s.t.} \quad \mathbb{P}\left\{ \sum_{i=1}^{n} \xi_i x_{ij} \geq d_j,\ j = 1, \ldots, m \right\} \geq 1 - \alpha, \\
& \qquad\quad \sum_{j=1}^{m} x_{ij} \leq 1,\ x_{ij} \geq 0,\ i = 1, \ldots, n,\ j = 1, \ldots, m,
\end{aligned}$$

where $x_{ij} \geq 0$ is the percentage of production shipped from supplier i to customer j. When d and ξ have finite discrete distributions, problem (STP) is a special case of (P).

We randomly generate instances of (STP) with $(n, m) = (40, 100)$ and $N = 500$,600, …, 1000. The cost coefficients c_{ij} are from the uniform distribution $\mathcal{U}[0, 1]$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m$. To generate the samples or scenarios of ξ and d, we first generate μ_i from $\mathcal{U}[100, 120]$ and σ_i from $\mathcal{U}[0, 10]$. The samples ξ^i $(i = 1, \ldots, N)$ are then generated from $\mathcal{N}(\mu, \Sigma)$, where $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_n^2)$. Similarly, we generate d^i $(i = 1, \ldots, N)$ from $\mathcal{N}(\mu, \Sigma)$ with $\mu_i \sim \mathcal{U}[30, 40]$ and $\sigma_i \sim \mathcal{U}[0, 5]$. For each sample size N, we generate 10 random instances and the average computational results are recorded. The risk level is set as $\alpha = 0.05$ and 0.10, respectively.

The comparison results for (STP) are summarized in Table 3. Comparing the results in Table 3, we see that the average objective values achieved by ADM and CPLEX are much better than those achieved by the CVaR approximation and the DC method, while ADM is able to get slightly better solutions than CPLEX in much less computing time. For all instances, CPLEX was terminated when reaching the CPU time limit (1800 seconds). We also observe from Figure 3 that the computing time of the DC method is slightly less than that of ADM for many instances, but it grows faster than ADM as the sample size increases. Overall, it appears that ADM is efficient for finding good optimal solutions of problem (STP) and compare favorably with the CVaR approximation, the DC method and CPLEX in terms of the objective value and the computing time.

Table 3: Comparison results for (STP) with different number of samples

		CVaR		DC		ADM			CPLEX	
N	α	fval	time	fval	time	fval	time	iter	fval	time
500	0.05	1355.1	2.5	1302.3	39.7	1259.4	151.4	1473	1258.1	1800
600	0.05	1313.8	2.4	1278.4	49.3	1237.9	164.4	1487	1237.2	1800
700	0.05	1363.1	3.7	1291.7	79.8	1241.9	172.2	1518	1243.8	1800
800	0.05	1359.3	4.6	1278.4	113.4	1221.0	184.8	1580	1227.1	1800
900	0.05	1459.4	5.1	1387.6	140.7	1320.0	191.1	1549	1323.4	1800
1000	0.05	1408.2	6.6	1309.7	194.4	1232.3	192.2	1562	1240.6	1800
500	0.1	1361.3	2.3	1301.3	70.3	1261.6	157.2	1443	1260.9	1800
600	0.1	1355.7	3.8	1299.5	102.7	1261.5	165.9	1485	1261.9	1800
700	0.1	1463.0	4.9	1337.2	170.7	1278.1	169.7	1522	1289.3	1800
800	0.1	1457.4	5.8	1365.0	206.4	1309.0	186.1	1552	1315.6	1800
900	0.1	1427.7	7.3	1328.8	289.0	1261.2	188.7	1554	1268.8	1800
1000	0.1	1537.1	8.4	1354.8	374.9	1259.0	208.4	1639	1278.6	1800

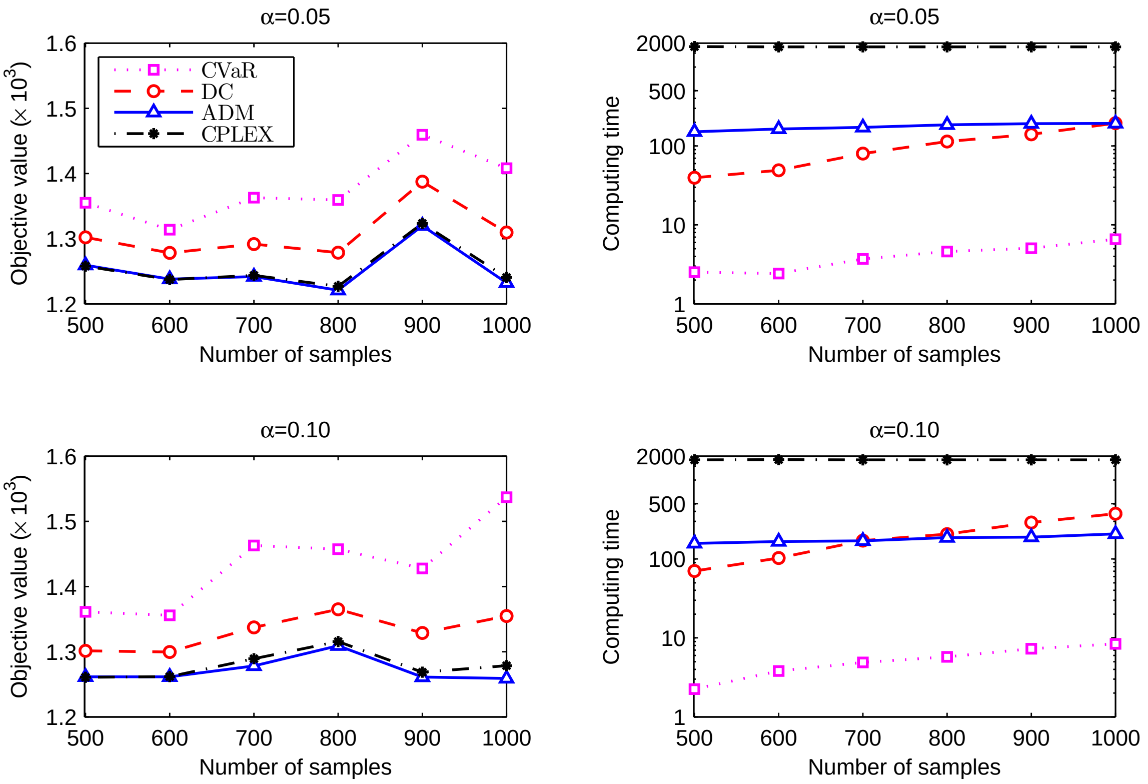

Figure 3: Objective values versus computing time for (STP) with different number of samples.

5. Conclusions

We have presented an alternating direction method (ADM) for solving the chance-constrained optimization problem with finite discrete distribution. Different from the convex conservative approximation methods, the proposed method is an iterative algorithm applied to the augmented Lagrangian decomposition formulation of the mixed-integer program reformulation of (P). The special structure of the probabilistic constraint motivated us to derive tractable subproblems by fixing certain variables and alternatively solve them at each iteration of the algorithm. We have also established the convergence of the method to the first-order stationary point of the original (nonconvex) problem. Our preliminary computational results show that the proposed method is promising for finding good quality approximate solutions and compares favorably with other existing approximate methods and CPLEX in terms of the trade-off between solution quality and computing time, especially for large-size problems.

Acknowledgments

This research was supported by the NSFC grants 11101092 and 11371103, Shanghai Pujiang Program 13PJC108, the Graduate Innovation grant of Fudan University, and the Provost's Chair Grant at School of Business, National University of Singapore.

References

Alexander, G.J., A.M. Baptista. 2004. A comparison of VaR and CVaR constraints on portfolio selection with the mean-variance model. *Management Science* **50** 1261–1273.

Alexander, G.J., A.M. Baptista, S. Yan. 2007. Mean-variance portfolio selection with 'at-risk' constraints and discrete distributions. *Journal of Banking and Finance* **31** 3761–3781.

Benati, S., R. Rizzi. 2007. A mixed integer linear programming formulation of the optimal mean/Value-at-Risk portfolio problem. *European Journal of Operational Research* **176** 423–434.

Beraldi, P., A. Ruszczyński. 2002a. A branch and bound method for stochastic integer problems under probabilistic constraints. *Optimization Methods and Software* **17** 359–382.

Beraldi, P., A. Ruszczyński. 2002b. The probabilistic set-covering problem. *Operations Research* **50** 956–967.

Bertsekas, D. P. 1982. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, Boston.

Bonami, P., M.A. Lejeune. 2009. An exact solution approach for portfolio optimization problems under stochastic and integer constraints. *Operations Research* **57** 650–670.

Boyd, S., L. Vandenberghe. 2004. *Convex Optimization.* Cambridge University Press.

Calafiore, G., M.C. Campi. 2005. Uncertain convex programs: Randomized solutions and confidence levels. *Mathematical Programming* **102** 25–46.

Calafiore, G.C., M.C. Campi. 2006. The scenario approach to robust control design. *IEEE Transactions on Automatic Control* **51** 742–753.

Charnes, A., W.W. Cooper. 1959. Chance-constrained programming. *Management Science* **6** 73–79.

Chen, W., M. Sim, J. Sun, C.P. Teo. 2010. From CVaR to uncertainty set: Implications in joint chance-constrained optimization. *Operations Research* **58** 470–485.

Cheng, J. Q., A. Lisser. 2012. A second-order cone programming approach for linear programs with joint probabilistic constraints. *Operations Research Letters* **40** 325–328.

Cheon, M.S., S. Ahmed, F. Al-Khayyal. 2006. A branch-reduce-cut algorithm for the global optimization of probabilistically constrained linear programs. *Mathematical Programming* **108** 617–634.

Clarke, F. H. 1983. *Optimization and Nonsmooth Analysis*. Wiley, New York.

Dentcheva, D., A. Prékopa, A. Ruszczynski. 2000. Concavity and efficient points of discrete distributions in probabilistic programming. *Mathematical Programming* **89** 55–77.

Gaivoronski, A.A., G. Pflug. 2005. Value at risk in portfolio optimization: Properties and computational approach. *Journal of Risk* **7** 1–31.

Goldstein, S., T.and Osher. 2009. The split Bregman method for ℓ_1-regularized problems. *SIAM Journal on Imaging Sciences* **2** 323–343.

He, B. S., M. Tao, X. M. Yuan. 2012. Alternating direction method with Gaussian back substitution for separable convex programming. *SIAM Journal on Optimization* **22** 313–340.

Henrion, R., C. Strugarek. 2008. Convexity of chance constraints with independent random variables. *Computational Optimization and Applications* **41** 263–276.

Hong, L.J., Y. Yang, L.W. Zhang. 2011. Sequential convex approximations to joint chance constrained programs: A Monte Carlo approach. *Operations Research* **59** 617–630.

Lagoa, C.M., X. Li, M. Sznaier. 2005. Probabilistically constrained linear programs and risk-adjusted controller design. *SIAM Journal on Optimization* **15** 938–951.

Lejeune, M., N. Noyan. 2010. Mathematical programming approaches for generating p-efficient points. *European Journal of Operational Research* **207** 590–600.

Lu, Z., Y. Zhang. 2011. Penalty decomposition methods for rank minimization. *Advances in Neural Information Processing Systems*. 46–54.

Lu, Z., Y. Zhang. 2013. Sparse approximation via penalty decomposition methods. *SIAM Journal on Optimization* To appear.

Luedtke, J. 2013. A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. *Mathematical Programming* doi: 10.1007/s10107-013-0684-6.

Luedtke, J., S. Ahmed, G.L. Nemhauser. 2010. An integer programming approach for linear programs with probabilistic constraints. *Mathematical Programming* **122** 247–272.

Luo, H. Z., X. L. Sun, D. Li. 2007. On the convergence of augmented Lagrangian methods for constrained global optimization. *SIAM Journal on Optimization* **18** 1209–1230.

Nemirovski, A., A. Shapiro. 2006a. Convex approximations of chance constrained programs. *SIAM Journal on Optimization* **17** 969–996.

Nemirovski, A., A. Shapiro. 2006b. Scenario approximations of chance constraints. G. Calafiore, F. Dabbene, eds., *Probabilistic and Randomized Methods For Design Under Uncertainty*. Springer-Verlag, London, 3–47.

Prékopa, A. 1990. Sharp bounds on probabilities using linear programming. *Operations Research* **38** 227–239.

Prékopa, A. 2003. Probabilistic programming. A. Ruszczyński, A. Shapiro, eds., *Handbooks in Operations Research and Management Science*, vol. 10. Elsevier, Amsterdam, 267–351.

Rockafellar, R.T., S. Uryasev. 2000. Optimization of conditional value-at-risk. *Journal of Risk* **2** 21–42.

Ruszczyński, A. 2002. Probabilistic programming with discrete distributions and precedence constrained knapsack polyhedra. *Mathematical Programming* **93** 195–215.

Sen, S. 1992. Relaxations for probabilistically constrained programs with discrete random variables. *Operations Research Letters* **11** 81–86.

Shapiro, A., D. Dentcheva, A. Ruszczyński. 2009. *Lectures on Stochastic Programming: Modeling and Theory*. MPS-SIAM Series on Optimization, SIAM, Philadelphia.

Shen, Y., Z. Z. Wen, Y. Zhang. 2013. Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization. *Optimization Methods and Software* doi:10.1080/10556788.2012.700713.

Sun, J., S. Zhang. 2010. A modified alternating direction method for convex quadratically constrained quadratic semidefinite programs. *European Journal of Operational Research* **207** 1210–1220.

van Ackooij, W., R. Henrion, A. Möller, R. Zorgati. 2011. On joint probabilistic constraints with Gaussian coefficient matrix. *Operations Research Letters* **39** 99–102.

Vielma, J. P., S. Ahmed, G. L. Nemhauser. 2012. Mixed integer linear programming formulations for probabilistic constraints. *Operations Research Letters* **40** 153–158.

Yin, W.T., S. Osher, D. Goldfarb, J. Darbon. 2008. Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing. *SIAM Journal on Imaging Sciences* **1** 143–168.

Zhang, S., J. Ang, J. Sun. 2013. An alternating direction method for solving convex nonlinear semidefinite programming problems. *Optimization* **62** 527–543.

Zheng, X. J., X. L. Sun, D. Li, X. T. Cui. 2012. Lagrangian decomposition and mixed-integer quadratic programming reformulations for probabilistically constrained quadratic programs. *European Journal of Operational Research* **221** 38–48.