RC24752 (W0902-087) February 20, 2009
Mathematics

IBM Research Report

An Efficient Decomposition Algorithm for Static, Stochastic, Linear and Mixed-Integer Linear Programs with Conditional-Value-at-Risk Constraints

Dharmashankar Subramanian, Pu Huang
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

An Efficient Decomposition Algorithm for Static, Stochastic, Linear and Mixed-Integer Linear Programs with Conditional-Value-at-Risk Constraints

Dharmashankar Subramanian, Pu Huang
Business Analytics & Mathematical Sciences, IBM Research
1101, Kitchawan Rd, Yorktown Heights, NY 10598
dharmash@us.ibm.com, puhuang@us.ibm.com

We present an efficient decomposition algorithm for single-stage, stochastic linear programs, where conditional value at risk (CVaR) appears as a risk measure in multiple constraints. It starts with a well-known nonlinear, convex reformulation of conditional value at risk constraints, and establishes the connection to a combinatorially large polyhedral representation of the convex feasible set induced by the above reformulation. The algorithm is developed as a column-generation procedure in the dual space of the resulting polyhedral representation. In the special case, where CVaR appears in the objective function alone, it offers an alternative view into *CVarMin*, a breakthrough algorithm in the literature, which specialized the L-shaped algorithm of two-stage stochastic programs with recourse for CVaR minimization. *CVarMin* concentrated on CVaR in the objective function, which led naturally to a two-stage interpretation. The optimization problem with multiple CVaR constraints does not have a natural two-stage interpretation, nevertheless we explicitly extend the polyhedral ideas of CVaR minimization and integrated chance constraints to the case of multiple CVaR constraints. We also present an algorithm that engineers the decomposition scheme to address mixed-integer linear programming problems with multiple CVaR constraints.

Key words: Conditional value at risk; CVaR constraints; CVaR Objective; Column generation; Discrete sampling approximation; Risk-constrained Optimization

---

1. Introduction.

In this article, we are interested in optimization formulations involving a decision vector $x \in \mathbb{R}^n$, along with a set of linear expressions, $a_i^T x$, $i \in M$, where each a_i is a random vector in $\mathbb{R}^n$, and is defined over its probability space, $(\Omega_i, \mathcal{F}_i, P_i)$ with a finite expected value. Such models are common in manufacturing, operations management, supply chain management, and finance in applications such as resource planning, capacity allocation, portfolio analysis, etc. The development in this article also trivially extends to the case of random affine expressions, i.e. $a_i^T x + \xi_i$, where each ξ_i, $i \in M$, is a random vector in $\mathbb{R}$. We focus on linear expressions for simplicity of notation, and note that for a given choice of x, each $a_i^T x$ is a random variable in $\mathbb{R}$. Further, let $\Phi_i(x, .)$ denote the corresponding probability distribution function. Using this notation, Value at Risk (VaR) and Conditional Value at Risk (CVaR) for each random variable, $a_i^T x$, are defined as the mappings shown below, where $1 - \beta \in (0,\, 1)$ denotes a small probability that is typically chosen as 0.05, or 0.01.

$$\mathrm{VaR}_\beta(a_i^T x) = \min\{y \in \mathbb{R} : \Phi_i(x, y) \geq \beta\} \quad (1)$$
$$\mathrm{CVaR}_\beta(a_i^T x) = \mathbb{E}[a_i^T x | a_i^T x \geq \mathrm{VaR}_\beta(a_i^T x)] \quad (2)$$

VaR is a popular asymmetric risk measure used in finance in order to quantify the downside risk of investment portfolios at a chosen quantile, say, β. However, it is undesirable from an optimization point of view, because VaR constraints are probabilistic constraints that lead in general to nonconvex mathematical programming formulations (Birge and Louveaux 2000, Kall and Wallace 1994). CVaR, on the other hand, is a convex risk measure and enjoys a convex representation from an optimization perspective (Artzner et al. (1999), Rockafellar and Uryasev (2000)).

In breakthrough papers, Rockafellar and Uryasev (2000, 2002) presented a nonlinear, convex reformulation for $\text{CVaR}_\beta(.)$ in Equation(2), in the context of a general distribution function, $\Phi_i(x,.)$. The reformulation, say, $F_\beta(.)$, is

$$F_\beta(a_i^T x) = \min_{z_i} \left[z_i + \frac{1}{1-\beta} \mathbb{E}[(a_i^T x - z_i)^+] \right] \tag{3}$$

where $t^+ = \max[0,t]$ is the positive funtion. In particular, they have shown that when each random vector, a_i has a discrete support, or is approximated by a set of N samples, say $\{(a_{i,1},\pi_1)...,(a_{i,N},\pi_N)\}$ where $\pi_k, k \in \{1,...,N\}$ denote the normalized sample probabilities, the following approximation, $\tilde{F}_\beta(.)$, has an objective function that is piece-wise linear and convex in the free variable, z_i.

$$\tilde{F}_\beta(a_i^T x) = \min_{z_i} \left[z_i + \frac{1}{(1-\beta)} \sum_{k=1}^{N} [\pi_k (a_{i,k}^T x - z_i)^+] \right] \tag{4}$$

Note that the special case of equi-probable samples corresponds to $\pi_k = 1/N, \forall k \in \{1,...,N\}$. Indeed, we use Equations (3) and (4) as the definition of CVaR for the purposes of our interest in CVaR-constrained optimization. Specifically, we are interested in CVaR-constrained, single-stage stochastic linear programming problems of the type shown below.

$$\min_x c^T x \tag{5}$$
$$\tilde{F}_\beta(a_i^T x) \le b_i, \forall i \in M \tag{6}$$
$$x \in \mathcal{P}$$

where $\mathcal{P} = \{x \,|\, Dx = d, x \ge 0\}$, represent a set of linear inequalities, $c \in \mathbb{R}^n$ is a deterministic vector of objective function coefficients. We assume for simplicity that $\mathcal{P}$ is bounded and non-empty. For each index, $i \in M$, we have a CVaR constraint that has been expressed using the reformulation shown in Equation(4). We assume in Problem(5) that we have access to a discrete sampling approximation for each random vector, a_i, along with the corresponding probabilities. Lastly, $b_i \in \mathbb{R}$ is a upper bound imposed upon the risk measure for modeling the maximum acceptable risk that one is willing to take on constraint, i.

In the sections that follow, we develop an efficient algorithm for Problem (5) by establishing the connection to a polyhedral reformulation of Constraint (6). The idea is related to the development of integrated chance constraints, originally proposed by Klein Haneveld and van der Vlerk (2006). It is also related to the development of *CVarMin*, an efficient breakthrough algorithm for CVaR minimization proposed by Künzi-Bay and Mayer (2006), where the authors concentrate on CVaR in the objective function. They imparted a two-stage stochastic programming interpretation to the objective function and specialized the L-shaped algorithm (Slyke and Wets 1969) to observe significant improvements in computational efficiency. CVaR in the constraint set does not naturally lend itself to a two-stage stochastic programming view. We explicitly extend the application of the underlying polyhedral view to address Problem (5) with CVaR constraints. As shown later in Section (3.2), Problem (5) can be reduced to CVaR minimization as a special case, and indeed, the algorithm presented in Section (2) specializes to a version of *CVarMin* in this case, and offers an alternative view into *CVarMin*.

2. An efficient algorithm for CVaR-constrained optimization

We develop an efficient algorithm for Problem (5) in this section. Firstly, we draw on a result from Krokhmal et al. (2002, Theorem 4), whereby we replace Constraint (6) with the following equivalent Constraint (8). The resulting reformulation of Problem (5) is

$$\min_{x,z}\ c^T x \tag{7}$$

$$z_i + \frac{1}{(1-\beta)} \sum_{k=1}^{N} [\pi_k (a_{i,k}^T x - z_i)^+] \le b_i, \forall i \in M \tag{8}$$

$$x \in \mathcal{P}$$

where z is a free variable in $\mathbb{R}^{|M|}$.

In terms of existing literature for Problem (5), it has been addressed by Rockafellar and Uryasev (2000), Rockafellar and Uryasev (2002), Krokhmal et al. (2002), where the authors linearize each constraint in the set (8) using N auxiliary variables and N additional constraints. This leads to a large linear program when the number of samples is large, and was solved using standard LP solvers. We refer to this formulation as the *Monolithic formulation* in Table (1). More recently, Huang and Subramanian (2008) presented a polyhedral view of CVaR constraints using the idea of an *order statistics* based CVaR estimator (Manistre and Hancock 2005) and developed an efficient algorithm, Iterative Estimation Maximization, to solve large instances of Problem (5). The development in this paper appeals to a different polyhedral view described below, which is along the lines of integrated chance constraints originally proposed by Klein Haneveld and van der Vlerk (2006), and also noted in *CVarMin* (Künzi-Bay and Mayer 2006), where the authors make a polyhedral connection for minimization of a CVaR objective via the L-shaped method. With respect to CVaR constraints, Klein Haneveld and van der Vlerk (2006) note the relationship with integrated chance constraints in their paper, but also note that the resulting reformulation may not be used directly due to the unknown Value-at-Risk (quantile) value. Instead, they suggest solving the problem as a canonical integrated chance constrained problem by using a fixed, apriori (guess) estimate of Value-at-Risk. The extension developed in this article is an improvement because it retains Value-at-Risk as an unknown variable, along with the unknown Conditional-value-at-risk, in CVaR constraints and enables an efficient decomposition algorithm.

Let us consider any representative constraint from the set in (8), say index, i, and rewrite it as,

$$(1-\beta) z_i + \sum_{k=1}^{N} \max[\pi_k (a_{i,k}^T x - z_i), 0] \le b_i (1-\beta) \tag{9}$$

We make two propositions that follow the idea of integrated chance constraints.

Proposition 1. *An equivalent representation for Constraint (9) is the following set,*

$$(1-\beta) z_i + \max_{\mathcal{A} \in \mathscr{N}} \sum_{k \in \mathcal{A}} \pi_k (a_{i,k}^T x - z_i) \le b_i (1-\beta) \tag{10}$$

$$z_i \le b_i \tag{11}$$

where $\mathscr{N}$ is the power set (excluding the empty set, $\emptyset$) of $\mathcal{N} = \{1, ..., N\}$, which is the index-set of the discrete sampling approximation.

Proof. It suffices to show that every feasible solution (z_i, x) that satisfies Constraint (9), also satisfies Constraints (10) and (11), and vice-versa.

Consider the forward direction, and say $(\hat{z}_i, \hat{x})$ satisfies Constraint (9). Firstly, Constraint (11) is directly implied by (9), because the second term on the left-hand-side of (9) is non-negative by definition. Further say $(a_{i,k}^T\hat{x}-\hat{z}_i)>0, \forall k \in \hat{K} \subseteq \mathcal{N}$, and $(a_{i,k}^T\hat{x}-\hat{z}_i) \leq 0, \forall k \in \mathcal{N} \setminus \hat{K}$. We may then rewrite Constraint (9) as,

$$(1-\beta)z_i + \sum_{k \in \hat{K}} \pi_k(a_{i,k}^T\hat{x}-\hat{z}_i) \leq b_i(1-\beta) \tag{12}$$

There are two possibilities. In the first case that $\hat{K} \neq \emptyset$, note that $\hat{K} = \hat{\mathcal{A}}^*$, i.e. the minimal (cardinality-wise) optimal subset of set $\mathcal{N}$ that maximizes $\sum_{k \in \mathcal{A}} \pi_k(a_{i,k}^T\hat{x}-\hat{z}_i)$, because it contains all the elements that make a positive contribution to the maximand. In other words, satisfaction of Constraint (9) implies (10) in this case, due to (12).

In the second case that $\hat{K} = \emptyset$, it is easy to see that,

$$\max_{\mathcal{A} \in \mathcal{N}} \sum_{k \in \mathcal{A}} \pi_k(a_{i,k}^T x - z_i) \leq 0 \tag{13}$$

Note that (13) in turn implies (10), due to (11). This concludes the forward direction.

Consider the reverse direction, and say, $(\hat{z}_i, \hat{x})$ satisfies Constraints (10) and (11). Further, say, $\hat{\mathcal{A}}^*$ is the minimal optimal subset on the left-hand-side of (10) that maximizes $\sum_{k \in \mathcal{A}} \pi_k(a_{i,k}^T\hat{x}-\hat{z}_i)$. This by definition leads to,

$$(a_{i,k}^T\hat{x}-\hat{z}_i) \leq 0, \forall k \in \mathcal{N} \setminus \hat{\mathcal{A}}^* \tag{14}$$

Now, there are two possibilities:

$$\text{Case 1: } (a_{i,k}^T\hat{x}-\hat{z}_i) > 0, \forall k \in \hat{\mathcal{A}}^* \tag{15}$$

$$\text{Case 2: } (a_{i,k}^T\hat{x}-\hat{z}_i) \leq 0, \forall k \in \hat{\mathcal{A}}^* \tag{16}$$

In Case 1, it is easy to see from (14) and (15) that,

$$\sum_{k=1}^{N} \max[\pi_k(a_{i,k}^T\hat{x}-\hat{z}_i), 0] = \sum_{k \in \hat{\mathcal{A}}^*} \pi_k(a_{i,k}^T\hat{x}-\hat{z}_i) \tag{17}$$

Equation (17) in turn implies (9) due to (10). In Case 2, it is easy to see from (14) and (16) that,

$$\sum_{k=1}^{N} \max[\pi_k(a_{i,k}^T\hat{x}-\hat{z}_i), 0] = 0 \tag{18}$$

Equation (18) in turn implies (9) due to (11). It is important to note that while z_i is a free variable in the context of Constraint (8) and Problem (7), Constraint (11) is critically essential to establish the above equivalence. $\square$

PROPOSITION 2. *An equivalent representation for Constraint (10) is the following combinatorially large constraint set given by,*

$$(1-\beta)z_i + \sum_{k \in \mathcal{A}} \pi_k(a_{i,k}^T x - z_i) \leq b_i(1-\beta) \forall \mathcal{A} \in \mathcal{N} \tag{19}$$

Proof. If each subset, $\mathcal{A} \subseteq \mathcal{N}$, in Constraint (19) satisfies the inequality, then the subset $\mathcal{A}^* \subseteq \mathcal{N}$, which is the maximum such left hand side also satisfies the inequality, thus implying Constraint (10). The reverse direction is also self-evident. If subset, $\mathcal{A}^*$, which maximizes the left hand side satisfies the inequality as per (10), then every other subset $\mathcal{A} \subseteq \mathcal{N}$ also satisfies the inequality, thus implying (19). $\square$

Using the above propositions in the context of CVaR constraints leads to the following reformulation of Problem (7).

$$\min_{x,z}\ c^T x \tag{20}$$
$$(1-\beta)z_i + \sum_{k\in\mathcal{A}} \pi_k(a_{i,k}^T x - z_i) \le b_i(1-\beta), \forall \mathcal{A} \in \mathscr{N}, \forall i \in M \tag{21}$$
$$z_i \le b_i, \forall i \in M \tag{22}$$
$$x \in \mathcal{P}$$

2.1. A column generation algorithm

We note that Problem (20) has added $|M|$ additional variables, and $2^N|M|$ additional constraints. Consider the dual corresponding to the above formulation.

$$\max_{p,q,r} \sum_{i\in M}\sum_{\mathcal{A}\in\mathscr{N}} b_i(1-\beta)p_{i,\mathcal{A}} + d^T q + b^T r \tag{23}$$
$$\sum_{i\in M}\sum_{\mathcal{A}\in\mathscr{N}}\sum_{k\in\mathcal{A}} \pi_k a_{i,k} p_{i,\mathcal{A}} + D^T q \le c \tag{24}$$
$$\sum_{\mathcal{A}\in\mathscr{N}} [(1-\beta) - \sum_{k\in\mathcal{A}} \pi_k] p_{i,\mathcal{A}} + r_i = 0, \forall i \in M \tag{25}$$
$$p_{i,\mathcal{A}}, r_i \le 0, \forall \mathcal{A} \in \mathscr{N}, i \in M$$

where $p_{i,\mathcal{A}}$ denotes the dual variable set corresponding to the constraint set (21), q is the free dual variable vector corresponding to $Dx = d$, and r is the dual variable vector corresponding to the constraint set (22). Note that Problem (23) has a combinatorially large number of columns, $\{p_{i,\mathcal{A}}\}$, which partition into $|M|$ subsets according to index, i. This lends itself naturally to a multi-column-generation scheme, which is a well-known decomposition technique (Dantzig and Wolfe 1960, Bertsimas and Tsitsiklis 1997). We describe the application of this technique to the above specific problem next.

Specifically, Problem (23) is decomposed into a master problem and a set of column-generating sub-problems, and the procedure iterates between these two levels. Such an iterative scheme starts with the master problem in iteration, $t = 0$, in the same form as Problem (23), but with a very small number of columns from the large set $\{p_{i,\mathcal{A}}\}$. In every iteration $t \ge 0$, the master problem has a small number of variables, $p_{i,\mathcal{A}}, \forall \mathcal{A} \in \mathscr{N}_t$, where $\mathscr{N}_t$ is a small subset of $\mathscr{N}$, i.e. $\mathscr{N}_t \subset \mathscr{N}, |\mathscr{N}_t| \ll |\mathscr{N}|$. Upon solution of the master problem in each iteration index, $t \ge 0$, (x_t^*, z_t^*) are the optimal dual variables corresponding to the restricted constraint sets (24) and (25). In each iteration, t, we have $|M|$ column-generating sub-problems that identify respectively the entering column from each of the $|M|$ combinatorial variable sub-sets, $\{p_{i,\mathcal{A}}\}$. These sub-problems take the form,

$$\max_{\mathcal{A}_i \subseteq \mathcal{N}} \sum_{k\in\mathcal{A}_i} \pi_k(a_{i,k}^T x_t^* - z_{it}^*) + (z_{it}^* - b_i)(1-\beta) \quad \forall i \in M \tag{26}$$
$$\mathcal{A}_i \subseteq \mathcal{N}$$

The solution to Problem (26), for each index i, is straightforward, because we need only evaluate and examine the sign of each term, $a_{i,k}^T x_t^* - z_{it}^*, k \in \mathcal{N}$, which can be accomplished in N arithmetic steps. With respect to solving the resulting sub-problems in the dual, this polyhedral view is more efficient than the order statistics based polyhedral view (Huang and Subramanian 2008) which needs a sorting procedure with a $O(N \log N)$ complexity for each dual sub-problem. Specifically, the optimal solution for each Problem (26) is,

$$\mathcal{A}_i^* = \{k \in \mathcal{N} \mid a_{i,k}^T x_t^* - z_{it}^* > 0\} \tag{27}$$

If the optimal objective in sub-problem, i, satisfies the following constraint,

$$\sum_{k\in\mathcal{A}_i{}^*}\pi_k(a_{i,k}^T x_t^* - z_{it}^*) + (z_{it}^* - b_i)(1-\beta) > 0 \tag{28}$$

then the corresponding column and variable, $p_{i,\mathcal{A}_i{}^*}$, is added to the master problem, thus leading to a maximum of $|M|$ additional columns added to the master in each iteration. The procedure terminates when none of the $|M|$ optimal sub-problem objective values satisfy Constraint (28), i.e. when there exists no column that increases the "reduced cost" for the maximization objective in the master problem. Note that it is possible to use a tolerance, say ϵ, in condition (28) to account for floating point arithmetic in a computer implementation.

In practice, it is sometimes easier to implement the above algorithm in the primal space in the form of a cutting plane method, and we summarize the resulting steps in the next section.

3. The Decomposition Algorithm steps in the primal space

We describe the multi-column generation scheme developed in Section (2.1) in the form of its correspondings steps in the primal space, where it takes the form of a delayed constraint generation algorithm. We also show the connection to *CVarMin*, when we specialize the technique to the case of a CVaR Objective. At each iteration, $t \geq 0$, the restricted master problem in the primal takes the form,

$$\min_{x,z}\ c^T x \tag{29}$$

$$\sum_{k\in\mathcal{A}}\pi_k(a_{i,k}^T x - z_i) + z_i(1-\beta) \leq b_i(1-\beta), \forall \mathcal{A}\in\mathscr{N}_{i,t}, \forall i\in M \tag{30}$$

$$z_i \leq b_i, \forall i \in M \tag{31}$$

$$x \in \mathcal{P}$$

The algorithm steps are as follows in the primal space.

Decomposition Algorithm

---

1. Initialize the iteration with $t=0$ and $\mathscr{N}_{i,0} = \mathcal{N}, \forall i\in M$.
2. For all $t \geq 0$, solve Problem (29) to optimality using a standard LP solver. If the problem is infeasible, terminate and declare infeasibility. Else, let the optimal solution be (x_t^*, z_t^*). Compute the subset $\mathcal{A}_i{}^*$ using Equation (27).
3. If Constraint (28) is violated for all $i \in M$, then (x_t^*, z_t^*) is the optimal solution to Problem (20). Terminate with the optimal solution, (x_t^*, z_t^*). Else Go to Step (4).
4. For each $i\in M$, if Constraint (28) is satisfied, update $\mathscr{N}_{i,t+1} = \mathscr{N}_{i,t}\cup\mathcal{A}_i{}^*$. Set $t = t+1$, and Go to Step (2).

Note that Equation (27) and condition (28) taken together in Steps (2) and (3) solve the *separation problem*, when viewed in this perspective. In Step (4), before updating $\mathscr{N}_{i,t+1}$, we may also choose to drop non-binding constraints from the set (30) by examining the optimal dual variable values from Step (2) in iteration t for each corresponding constraint. This is often effective in practice towards producing a smaller linear program in each iteration, thereby leading to computational speed-up for the overall algorithm. In the absence of degeneracy, dropping non-binding constraints is not a problem (Bertsimas and Tsitsiklis 1997), while in the presence of degeneracy, one may need an anti-cycling rule such as lexicographic ordering of the variable set, $p_{i,\mathcal{A}}$. It is reasonable to expect efficient performance in practice for large linear programs with multiple CVaR constraints with the above algorithm, since it corresponds identically to a column generation scheme.

3.1. The case of single-stage mixed-integer linear programs

It should be noted that the above iterative scheme will always have a finite termination if we do not drop any constraints across iterations. This is because the set $\mathscr{N}$ is finite, albeit very large. Also, termination implies optimality of the resulting solution for the full master problem, because the terminating optimal solution of the restricted master problem (with fewer constraints) will satify all the constraints in the full master problem by design. In principle, the above steps can therefore also be applied to mixed-integer linear programming (MILP) problems with CVaR constraints by solving a sequence of integer programming problems in Step (2) above. However, the number of iterations and the resulting computational burden will likely be very high if applied in such a plain iterative fashion, due to the complexity of solving an integer programming problem in each step.

We engineer the following solution approach for mixed-integer linear programs with CVaR constraints for improving the computational performance in practice. Say we restrict some (or all) of the variables, x to be binary, and add the corresponding integrality constraints to Problem (20). Let us call the corresponding full master problem, i.e. the corresponding Problem (20) augmented with binary integrality constraints, as Problem $MILP_f$. This is our single-stage mixed integer linear program with CVaR constraints. Further, let us denote the corresponding restricted master problem, i.e. the corresponding Problem (29) augmented with binary integrality constraints, as Problem $MILP_r$. The following algorithm applies to stochastic mixed-integer linear programs with CVaR constraints.

Decomposition Algorithm for the mixed-integer linear programs

---

1. Solve the LP Relaxation of $MILP_f$, using the above Decomposition Algorithm. Say the LP Relaxation terminates in iteration τ.
2. Re-Initialize the iteration counter with $t=0$ and $\mathscr{N}_{i,0}=\mathscr{N}_{i,\tau}, \forall i \in M$, where $\mathscr{N}_{i,\tau}$ is the collection of constraints accumulated from the above step of solving the LP Relaxation.
3. For all $t \geq 0$, solve the corresponding Problem $MILP_r$ (with integrality constraints), to optimality using a standard MILP solver. If the problem is infeasible, terminate and declare infeasibility. Else, let the optimal solution be (x_t^*, z_t^*). Compute the subset $\mathcal{A}_i{}^*$ using Equation (27).
4. If Constraint (28) is violated for all $i \in M$, then (x_t^*, z_t^*) is the optimal solution to Problem $MILP_f$. Terminate with the optimal solution, (x_t^*, z_t^*). Else Go to Step (5).
5. For each $i \in M$, if Constraint (28) is satisfied, update $\mathscr{N}_{i,t+1}=\mathscr{N}_{i,t} \cup \mathcal{A}_i{}^*$. Set $t=t+1$, and Go to Step (3).

Step (1) in the above algorithm is expected to contribute to the speed-up of the iterative scheme for the mixed-integer linear programming case, because it will establish a collection of cuts that are necessary to charaterize the optimality of the LP relaxation by solving a sequence of linear programs. The integrality constraints may require additional constraints to characterize the feasible set of the MILP, which is accomplished by Steps (2) - (6). The effectiveness of this solution approach on large mixed-integer problem instances will be computationally investigated in a later study.

3.2. The Special Case of CVaR Minimization

Finally, we also note that the algorithm steps in the primal space bear a connection to *CVarMin* (Künzi-Bay and Mayer 2006), and we make this explicit in this sub-section. Say we have a CVaR minimization problem given by,

$$\begin{aligned} &\min_x \; \tilde{F}_\beta(a^T x) \\ &x \in \mathcal{P} \end{aligned} \tag{32}$$

where, a is a random vector with a discrete support, or is approximated by a set of N samples, $\{(a_1,\pi_1)...,(a_N,\pi_N)\}$, and $\pi_k, k\in\mathcal{N}$ denote the normalized sample probabilities, as before. Problem (32) may be recast as a CVaR minimization problem, by adding a free auxiliary variable y, and replacing it with,

$$\min_{x,z,y}\ y \tag{33}$$
$$\tilde{F}_\beta(a^Tx)\le y \tag{34}$$
$$x\in\mathcal{P}$$

Problem (33) is equivalent to Problem (32) due to the convex properties of all expressions involved. The least value of y will coincide with the least value of the CVaR objective,$\tilde{F}_\beta(a^Tx)$. Along the same lines as the reformulation of Constraint (6) through (8) into (21), we reformulate Problem (33) as,

$$\min_{x,z,y}\ y \tag{35}$$
$$\sum_{k\in\mathcal{A}}\pi_k(a_k^Tx-z)+(z-y)(1-\beta)\le 0,\forall\mathcal{A}\in\mathscr{N} \tag{36}$$
$$z\le y \tag{37}$$
$$x\in\mathcal{P}$$

The Decomposition Algorithm in Section (3) can be specialized for Problem (35) by replacing Equation (27) and Constraint (28) respectively with,

$$\mathcal{A}_i{}^*=\{k\in\mathcal{N}\mid a_k^Tx_t^*-z_t^*>0\} \tag{38}$$

and

$$\sum_{k\in\mathcal{A}_i{}^*}\pi_k(a_k^Tx_t^*-z_t^*)+(z_t^*-y_t^*)(1-\beta)>0 \tag{39}$$

where y_t^* is the optimal solution corresponding to the auxiliary variable y in any iteration, $t\ge 0$. It can be seen that the above specialization of the algorithm to the case of a CVaR objective is equivalent to a version of the two-stage interpretation and the resulting algorithm presented in *CVarMin*.

4. Computational experience for linear programs

In order to characterize the computational performance of decomposition algorithm in practice, we conducted a comparison study on a set of problem instances. These problems were generated using the same instance generator that was reported in Huang and Subramanian (2008) for the purposes of comparison. We describe the instance generator and report the observed computational performance of our algorithm, in comparison with the monolithic linear programming reformulation (Rockafellar and Uryasev 2000) as well as Iterative Estimation Maximization (IEM). The formulation used in the instance generator was,

$$\min_x c^Tx \tag{40}$$
$$\text{CVaR}_\beta\Big(\sum_{i\in I}a_{ji}x_i\Big)\le b_j\forall j\in J \tag{41}$$
$$0\le x\le 1$$

In the above formulation, x is the decision variable vector, indexed over set, $I=\{1,...,V\}$, where V was fixed at 30. The constraint set (41) is a set of CVaR constraints, indexed over set $J=\{1,...,W\}$,

Table 1 Comparison of Computational Performance (seconds).

W	*Monolithic formulation*	*Iterative Estimation Maximization*	*Decomposition algorithm*
2	0.11	45.01	60.21
10	9.42	81.13	60.43
50	273.32	129.26	138.78
100	1148.43	420.14	211.87
150	2880.52	494.16	334.10
200	Terminated at 5400.00	730.21	672.40
250	-	812.06	513.25
300	-	592.96	612.59

where W was varied from 2 to 300. The stochastic parameters, a_{ji}, were generated using a random seed, and the following equation, which generates a set of positive, random, constraint coefficients.

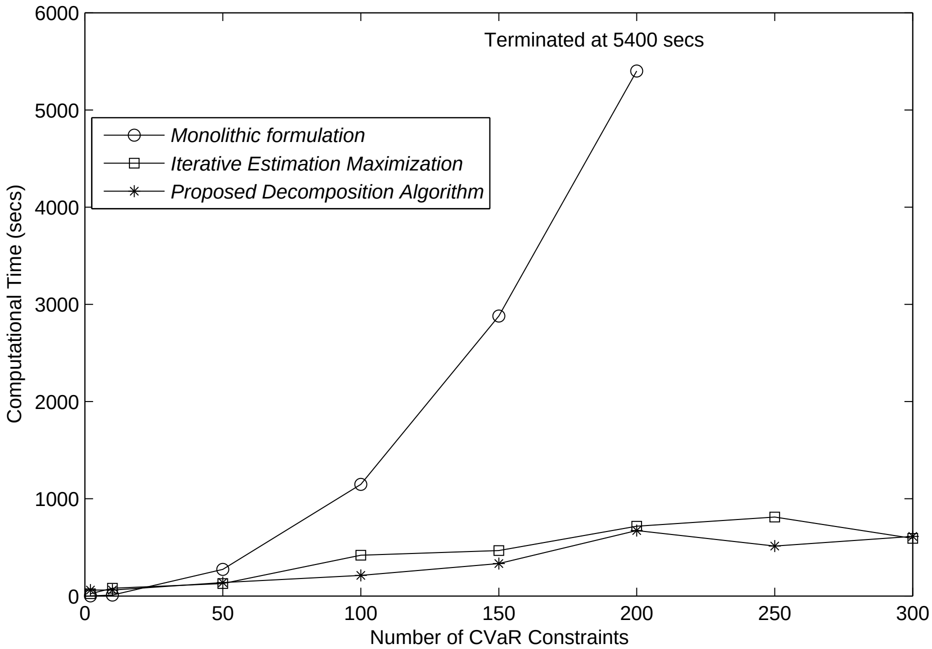
$$a_{ji} = \max(0.1, \text{Normal}(\text{Uniform}(1,10),\ \text{Uniform}(5,10))) \tag{42}$$

The objective function coefficient vector, c was fixed with random integers in the range [1,10], and the constraint right-hand-side coefficients, b_j were fixed to take a value of 1, for each constraint $j \in J$.

Note that the above instance generator will generate an problem instance, upon specification of the number of rows, W, i.e. the number of CVaR constraints. We varied the number of rows from from 2 to 300, and used a sample size of $N = 1000$ to approximate each problem instance. Further, we used a error tolerance parameter of 1E-6 in order to decide termination of the proposed algorithm as well as IEM. Lastly, we used ILOG CPLEX 10.0 as the linear programming solver for both algorithms. The computations were performed on a Lenovo Thinkpad T60p, with an Intel CPU, T2600 at 2.16 GHz, and 2 GB RAM. The table below shows the resulting computational performance. The above results in the table are also shown in Figure (1).

The above results for linear programs show that the performance of the decomposition algorithm is significantly superior to the monolithic formulation when the number of rows (constraints) is large, and is quite competitive compared to Iterative Estimation Maximization. This is expected because the monolithic linear programming reformulation (Rockafellar and Uryasev 2000) introduces a large number of additional variables and constraints. In other words, if there are W original constraints, the reformulation introduces NW auxiliary variables and NW additional constraints, where $N = 1000$ samples. As shown in Figure (1), when W increases from 2 to 200, the burden imposed by this increased complexity affects the computational performance significantly. The monolithic formulation had to be terminated at 5400 seconds for $W = 200$ without a result. On the other hand, the proposed decomposition algorithm does not suffer from such an increase in complexity. This is because in each iteration, we solve a linear programming problem that is comparable in same size and complexity to the deterministic version of problem. This is very appealing from both a computational time, as well as computer memory size point of view. It is also very appealing in terms of the approximation quality, because the algorithmic complexity is relatively less sensitive to the number of samples that are chosen in the sampling approximation. Further, it is competitive compared to Iterative Estimation Maximization (IEM) because the solving the dual sub-problems in the proposed algorithm requires only a complexity of WN steps in each iteration, as compared to $O(WN \log N)$ sorting complexity that is needed in IEM in each iteration. Nevertheless, both IEM and the proposed decomposition algorithm perform quite comparably on large instances, albeit exploiting different polyhedral reformulations.

Figure 1 Comparison of Computational Peformance



Note. Comparing the Monolithic formulation of Rockafellar and Uryasev (2000), Iterative Estimation Maximization (Huang and Subramanian 2008) and the Proposed Decomposition algorithm.

5. Conclusion

We have presented an efficient algorithm for single-stage, stochastic linear programs, where the conditional value at risk (CVaR) risk measure appears in multiple constraints. Computational tests on randomly generated problem instances show promising results in comparison with alternative approaches. It uses the well-known nonlinear, convex reformulation (Rockafellar and Uryasev 2000) of conditional value at risk constraints, and establishes the connection to a polyhedral representation that is related to integrated chance constraints originally developed by Klein Haneveld and van der Vlerk (2006). When specialized to the case of CVaR minimization, it offers an alternative view into *CVarMin* developed by Künzi-Bay and Mayer (2006). The polyhedral structure implicit in CVaR constraints lends itself to efficient algorithms for CVaR-constrained optimization, and notably has multiple polyhedral representations that can be exploited as shown in this article as well as Huang and Subramanian (2008). We have also presented an algorithm that engineers the decomposition scheme for single-stage, stochastic mixed-integer linear programs with CVaR constraints. We intend to carry out further computational tests to investigate the effectiveness on large mixed-integer problem instances in a later study.

References

Artzner, Philippe, Freddy Delbaen, Jean-Marc Eber, David Heath. 1999. Coherent measures of risk. *Mathematical Finance* **9**(3) 203–228.

Bertsimas, Dimitris, John N. Tsitsiklis. 1997. *Introduction to Linear Optimization*. Athena Scientific.

Birge, John R., Francois Louveaux. 2000. *Introduction to Stochastic Programming*. 2nd ed. Springer. URL `http://www.springer.com/business/operations+research/book/978-0-387-98217-5`.

Dantzig, George B., Philip Wolfe. 1960. Decomposition principle for linear programs. *Operations Research* **8**(1) 101–111.

Huang, Pu, Dharmashankar Subramanian. 2008. Iterative estimation maximization for stochastic linear and convex programs with conditional-value-at-risk constraints. Tech. rep., IBM T.J.Watson Research Center. Report RC24535, Also vailable at http://www.optimization-online.org/DB_HTML/2008/04/1957.html.

Kall, Peter, Stein W. Wallace. 1994. *Stochastic Programming*. John Wiley & Sons, Chichester.

Klein Haneveld, Willem K., Maarten H. van der Vlerk. 2006. Integrated chance constraints: reduced forms and an algorithm. *Computational Management Science* **3**(4) 245–269.

Krokhmal, P., J. Palmquis, S. Uryasev. 2002. Portfolio optimization with conditional value-at-risk objective and constraints. *The Journal of Risk* **4**(2) 11–27.

Künzi-Bay, A., János Mayer. 2006. Computational aspects of minimizing conditional value-at-risk. *Computational Management Science* **3** 3–27.

Manistre, B. John, Geoffrey H. Hancock. 2005. Variance of the cte estimator. *North American Actuarial Journal* **9** 129–156.

Rockafellar, R. Tyrrell, Stanislav Uryasev. 2000. Optimization of conditional value-at-risk. *The Journal of Risk* **2** 21–41.

Rockafellar, R. Tyrrell, Stanislav Uryasev. 2002. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance* **26** 1443–1471.

Slyke, R. M. Van, Roger Wets. 1969. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics* **17**(4) 638–663. doi:10.1137/0117061. URL http://link.aip.org/link/?SMM/17/638/1.