

RC24535 (W0804-055) April 16, 2008

Mathematics

# IBM Research Report

## Iterative Estimation Maximization for Stochastic Linear and Convex Programs with Conditional-Value-at-Risk Constraints

**Pu Huang, Dharmashankar Subramanian**

IBM Research Division

Thomas J. Watson Research Center

P.O. Box 218

Yorktown Heights, NY 10598



Research Division

Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

# Iterative Estimation Maximization for Stochastic Linear and Convex Programs with Conditional-Value-at-Risk Constraints

Pu Huang  
IBM Research  
Yorktown Heights, NY, 10598

Dharmashankar Subramanian  
IBM Research  
Yorktown Heights, NY, 10598

## Abstract

We present a new algorithm, Iterative Estimation Maximization (IEM), for stochastic linear and convex programs with Conditional-Value-at-Risk (CVaR) constraints. IEM iteratively constructs a sequence of compact-sized linear, or convex, optimization problems, and solves them sequentially to find the optimal solution. The problem size IEM solves in each iteration is unaffected by the size of random samples, which makes it extreme efficient for real-world, large-scale problems. We prove that IEM converges to the true optimal solution, and give a lower bound on the number of samples required to probabilistically satisfy a CVaR constraint. Experiments show that IEM is an order of magnitude faster than the best known algorithm on large problem instances.

## 1. Introduction

Consider the following risk-constrained stochastic linear program,

$$\begin{aligned} & \max c^T x \\ \text{(CSLP)} \quad & \text{CVaR}_\beta(\zeta^T x) \leq b \\ & x \in D \end{aligned}$$

where  $D \subset \mathbf{R}^n$  is a convex polytope generated by a set of linear inequalities,  $c \in \mathbf{R}^n$  is a vector of objective function coefficients,  $\zeta$  is a random vector in space  $\mathbf{R}^n$ ,  $\text{CVaR}_\beta(\cdot)$  is a risk measure called Conditional-Value-at-Risk (CVaR), which maps a random variable to a real number in  $\mathbf{R}$ ,  $b \in \mathbf{R}$  is a upper bound imposed upon risk measure  $\text{CVaR}_\beta(\cdot)$ , modeling the maximum acceptable risk one is willing to take.

Problem CSLP extends the linear programming model. It allows input parameters to be random variables, and thus can capture uncertainty associated with the problem of interest. CSLP can be applied to model a large set of practically important problems. One application is in portfolio optimization. In a portfolio optimization problem, random vector  $\zeta$  represents the future security prices, decision variable  $x$  represents the portions of total wealth allocated to individual securities,  $\text{CVaR}_\beta(\zeta^T x)$  measures the risk of a given allocation  $x$ ,  $b$  bounds the maximum level of risk,  $c$  typically contains the expected return of individual securities, and budget and other constraints can be modeled as  $x \in D$ .

The risk measure that we consider in this paper is CVaR, which is coherent in the sense of Artzer et al. (1999). It was proposed as an alternative to Value-at-Risk (VaR), which is a popular risk measure widely used in financial industry. VaR is the lowest potential loss that may occur with a given small probability. Let  $Y$  denote a random variable with continuous probability density function<sup>1</sup>, and  $1 - \beta \in (0, 1)$  denote a small probability that a large loss may occur, and then VaR is defined as

$$\text{VaR}_\beta(Y) = \inf\{y \in \mathbf{R} : \Pr[Y > y] \leq 1 - \beta\},$$

and CVaR is defined as

$$\text{CVaR}_\beta(Y) = E[Y | Y \geq \text{VaR}_\beta(Y)].$$

Basically, CVaR measures the average extreme loss given that the loss is greater than VaR.

If we replace the CVaR risk measure in problem CSLP by VaR, we get a chance-constrained stochastic linear program, which has been studied extensively (see, e.g., Kall and Wallace (1994), and for most recent development, see Nemirovski and Shapiro (2006) and references therein). One major difficulty of applying a chance constraint to bound

---

<sup>1</sup> The definition of CVaR for general discontinuous probability density functions is more complicated, involving a procedure of “splitting probability mass” at discontinuous points. For simplicity, we assume random variables have continuous density functions throughout the paper. Our method is not affected by this assumption.

risk is that in general it leads to a non-convex problem, which is difficult to solve except for a few special cases.

The feasible region of problem CSLP is convex because of the nice properties that CVaR enjoys. Follow Artzer et al. (1999), CVaR is a coherent risk measure and thus satisfies the following four properties any coherent risk  $\rho(\cdot)$  does:

- Translation invariance. For any random variable  $Y$  and any real number  $k$ ,  
$$\rho(Y + k) = \rho(Y) + k .$$
- Subadditivity. For any random variables  $X$  and  $Y$ ,  
$$\rho(X + Y) \leq \rho(X) + \rho(Y) .$$
- Positive homogeneity. For any random variable  $Y$  and any real number  $\lambda \geq 0$ ,  
$$\rho(\lambda Y) = \lambda \rho(Y) .$$
- Monotonicity. For any random variables  $X$  and  $Y$ , if  $X \leq Y$  almost surely,  
$$\rho(X) \leq \rho(Y) .$$

The feasible region defined by the constraint,  $\text{CVaR}_\beta(\zeta^T x) \leq b$ , is clearly convex, as implied by subadditivity and positive homogeneity. This implies that, potentially, large instances of problem CSLP can be solved efficiently.

Though it is a convex problem, finding the exact solution to problem CSLP is impossible in the general case. This is because even for a given  $x$ , computing the distribution of  $\zeta^T x$  requires performing a  $n$ -dimensional integration ( $n$  is the dimension of  $x$  and  $\zeta$ ). If  $n$  is large, we will be overwhelmed by the “curse of dimensionality”.

One way to overcome this difficulty is to take a sample approximation approach, in which the original problem CSLP is approximated by another problem that is constructed based on a set of samples of random parameter  $\xi$ . We take this approach in this paper. For brevity, we do not differentiate between problem CSLP and its sample approximation, and simply call the optimal solution to a sample approximation as “the optimal solution to problem CSLP”.

In this paper, we propose an algorithm, Iterative Estimation Maximization (IEM), to solve problem CSLP. IEM iteratively constructs a sequence of compact-sized linear programs and solves them sequentially to find the optimal solution. The solution obtained in the preceding iteration is used to construct the linear program for the next one, which guarantees that we always get a better approximation to the optimal solution as the algorithm progresses. The size of the linear program that IEM solves in each iteration is comparable to the original problem, and is unaffected by the size of random samples. This makes it extremely efficient for large problem instances. Indeed, we can show that IEM can be interpreted as embedding a set of column generation sub-problems to the well-known simplex method. In this interpretation, the sub-problems are constructed in such a way that each sub-problem can be solved in  $O(N \log N)$  time, where  $N$  is the size of samples. This interpretation explains the good practical performance of IEM. Considering we are solving a stochastic linear program with  $N$  samples, the extra overhead of  $O(N \log N)$  seems to be unavoidable.

We also extend IEM to solve the general CVaR-constrained stochastic convex programs, which has the following canonical representation

$$\begin{aligned}
 & \max h(x) \\
 \text{(CSCP)} \quad & \text{CVaR}_\beta[g(\xi, x)] \leq b \\
 & x \in D
 \end{aligned}$$

where  $D$  is a convex subset of  $\mathbf{R}^n$ ,  $h(x)$  is a strictly concave function, and  $g(\xi, x)$  is a convex function on  $x$  for any  $\xi$ . Problem CSCP is a natural extension to convex programs, along the same lines as problem CSLP is an extension to linear programs. IEM takes a similar procedure to solve problem CSCP: a new convex program is constructed and then solved iteratively. Like in the linear case, the convex problem solved in each IEM iteration is unaffected by the size of samples. As far as we know, IEM is the first algorithm that solves problem CSCP.

Rockafellar and Uryasev (2000), and Krokmal et al. (2002) proposed a sample approximation based method for problem CSLP. Their key observation is that the l.h.s.

for the CVaR constraint can be approximated by its sample average estimation. Let  $\xi_1, \xi_2, \dots, \xi_N$  denote  $N$  samples of random vector  $\xi$ , then constraint  $\text{CVaR}_\beta(\xi^T x) \leq b$  can be approximated by

$$\alpha + \frac{1}{N(1-\beta)} \sum_{i=1}^N (\xi_i^T x - \alpha)^+ \leq b$$

where  $\alpha$  is a free auxiliary variable. Introducing variable  $z_i$ ,  $i = 1, 2, \dots, N$ , the above constraint can be linearized and replaced by the following set of constraints

$$\begin{aligned} a + \frac{1}{N(1-\beta)} \sum_{i=1}^N z_i &\leq b \\ z_i &\geq \xi_i^T x - \alpha, \quad i = 1, 2, \dots, N \\ z_i &\geq 0, \quad i = 1, 2, \dots, N \end{aligned}$$

This procedure transforms problem CSLP into a large linear program and solves it accordingly. One major shortcoming of this approach is that both the number of auxiliary variables  $z_i$  and the number of newly introduced constraints are proportional to the number of samples, which makes this approach impractical even for modest size original problems, when the sample size is large. Our IEM algorithm does not suffer from this shortcoming.

The rest of the paper is organized as follows. Section 2 presents the IEM algorithm and proves it correctively solves CVaR-constrained stochastic linear programs. In Section 3, we extend the IEM algorithm to CVaR-constrained stochastic convex programs and prove that it converges. Section 4 gives a lower bound on the size of samples required so that a CVaR constraint can be satisfied with high probability. Section 5 shows the performance of IEM on randomly generated problem instances. Section 6 concludes the paper.

## 2. Iterative Estimation Maximization

We first describe the IEM algorithm.

---

Iterative Estimation Maximization:

1. Generate  $N$  samples  $\xi_1, \xi_2, \dots, \xi_N$  for the random vector  $\xi$ .
2. Set iteration index  $t = 0$ . let  $LP^0$  denote the initial linear program at  $t = 0$ ,

$$(LP^0) \quad \max_{x \in D} c^T x$$

Solve  $LP^0$  to get the initial value  $x^0 \in D$ .

3. Let  $LP^t$  and  $x^t$  denote the linear program and the solution in the current iteration  $t$ . Perform the following steps for the constraint,  $\text{CVaR}_\beta(\xi^T x) \leq b$ , in the original problem CSLP:

- a. Compute  $L_i = \xi_i^T x^t$  for each sample  $\xi_i, i = 1, 2, \dots, N$ .
- b. Sort  $L_i, i = 1, 2, \dots, N$  in ascending order. Let  $L_{(i)}$  denote the  $i$ -th smallest value in the list of sorted  $L_i$  values, and  $\xi_{(i)}$  denote the corresponding sample used to compute  $L_{(i)}$ , i.e.,

$$\xi_{(1)}^T x^t \leq \xi_{(2)}^T x^t \leq \dots \leq \xi_{(N)}^T x^t \quad (1)$$

- c. Let  $K$  denote the smallest integer that is greater than  $N\beta$ , i.e.,

$$K = \min\{i \in \{1, 2, \dots, N\} : i > N\beta\}, \quad (2)$$

add the following constraint to the current linear program  $LP^t$ ,

$$\frac{1}{N(1-\beta)} \sum_{i=K}^N \xi_{(i)}^T x \leq b. \quad (3)$$

4. After finishing step 3, we obtain a new linear program  $LP^{t+1}$  by adding a new constraint (3) to  $LP^t$ . Now solve  $LP^{t+1}$  to get a new solution  $x^{t+1}$ .
5. Remove all non-binding constraints like (3) that have been added to  $LP^{t+1}$  in previous iterations.

6. If  $\|x^{t+1} - x^t\|$  is small enough, terminate, and output  $x^* = x^{t+1}$  as the optimal solution to the original problem CSLP. Otherwise, set  $t = t + 1$ ,  $x^t = x^{t+1}$ , and go back to step 3 to start the next iteration.
- 

For a given solution,  $x^t$ , step 3a and 3b estimate  $\text{CVaR}_\beta(\zeta^T x^t)$  based on samples  $\xi_1^T x^t, \xi_2^T x^t, \dots, \xi_N^T x^t$ . In step 3c, those  $\xi_i$ 's that have contributed to the estimation, i.e., top  $N - K + 1$  samples, are used to create constraint (3). Intuitively, IEM seeks to iteratively replace the CVaR constraint in the original problem with estimates in the form of linear constraints as shown in step (3). It does so, by using the best solution obtained so far to speculate the coefficients of the linear constraint (3). When IEM terminates,  $\text{CVaR}_\beta(\zeta^T x^*)$ , the CVaR value of  $\zeta^T x^*$  corresponding to the optimal solution  $x^*$ , is estimated by the l.h.s. of constraint (3), and conversely, solving a linear program with such a constraint leads to the optimal solution to the original problem CSLP.

To prove that IEM indeed converges, we first construct a sample approximation to problem CSLP, and then demonstrate that IEM solves such an approximation.

Let  $\Phi$  denote the collection of all subsets of the set,  $\{\xi_1, \xi_2, \dots, \xi_N\}$ , with size,  $N - K + 1$ , and  $I_A(\cdot)$  denote the indicator function of a subset  $A \in \Phi$ , i.e.,

$$I_A(a) = \begin{cases} 1 & a \in A \\ 0 & a \notin A \end{cases}.$$

Corresponding to all subsets  $A \in \Phi$ , we can construct a set of linear constraints (the number of constraints is  $C_{N-K-1}^N$ ), which take the following form.

$$\frac{1}{N(1-\beta)} \sum_{i=1}^N \xi_i^T x I_A(\xi_i) \leq b, \quad \forall A \in \Phi. \quad (4)$$

Note that constraint (3) in step 3c of the IEM algorithm belongs to constraint set (4) for some subset  $A \in \Phi$ .



**Proposition 1:** Constraint set (4) is a sample approximation to the CVaR constraint

$$\text{CVaR}_\beta(\zeta^T x) \leq b \quad (5)$$

in problem CSLP.

**Proof:** For  $\forall x \in \mathbf{R}^n$ , constraint (5) can be approximated by

$$\frac{1}{N(1-\beta)} \sum_{i=1}^N \xi_i^T x I_{B(x)}(\xi_i) \leq b \quad (6)$$

where  $B(x) \in \Phi$  is a subset of  $\{\xi_1, \xi_2, \dots, \xi_N\}$ , with size  $N - K + 1$ , such that for

$\forall \xi \in B(x)$  and  $\forall \bar{\xi} \notin B(x)$ ,  $\xi^T x \geq \bar{\xi}^T x$ , i.e.,

$$B(x) = \{A \in \Phi : \xi^T x \geq \bar{\xi}^T x, \quad \forall \xi \in A \text{ and } \forall \bar{\xi} \notin A\} \quad (7)$$

We call  $B(x)$  the dominant set induced by  $x$ . In essence,  $B(x)$  contains the top

$N - K + 1$  samples that have the largest  $\xi^T x$  values.

Now we claim that for  $\forall A \in \Phi - B(x)$ , if  $x$  satisfies (6), it also satisfies all the rest of the constraints, shown in (8).

$$\frac{1}{N(1-\beta)} \sum_{i=1}^N \xi_i^T x I_A(\xi_i) \leq b, \quad \forall A \in \Phi - B(x) \quad (8)$$

By the definition of  $B(x)$ , we know that

$$\sum_{i=1}^N \xi_i^T x I_A(\xi_i) \leq \sum_{i=1}^N \xi_i^T x I_{B(x)}(\xi_i), \quad \forall A \in \Phi - B(x)$$

as replacing  $\forall \xi \in B(x)$  by  $\forall \bar{\xi} \notin B$  always results in a smaller  $\xi^T x$  value. Constraints (6)

and (8) imply constraint set (4).  $\square$

**Proposition 2:** IEM solves the following linear program, which is a sample approximation to problem CSLP,

$$\begin{aligned} & \max c^T x \\ \text{(SA-CSLP)} \quad & \frac{1}{N(1-\beta)} \sum_{i=1}^N \xi_i^T x I_A(\xi_i) \leq b, \quad \forall A \in \Phi \\ & x \in D \end{aligned}$$

**Proof:** Let  $\lambda_A$  denote the dual variable corresponding to the constraint indexed by  $A \in \Phi$ . Consider applying the simplex method to solve the dual of problem SA-CSLP. In each simplex pivot step, a column that has the minimum reduced cost

$$b - \frac{1}{N(1-\beta)} \sum_{i=1}^N \xi_i^T x I_A(\xi_i)$$

is put into the base. Finding the minimum reduced cost is equivalent to solving the following column generation problem

$$\max_A \left\{ \sum_{i=1}^N \xi_i^T x I_A(\xi_i) : A \in \Phi \right\}$$

It is clear that for a given  $x$ , its dominant set  $B(x)$  solves the above column generation problem. Finding  $B(x)$  is easy, because we only need to compute  $\xi_i^T x$  for each  $\xi_i, i = 1, 2, \dots, N$ , and then sort them and pick up the top  $N - K + 1$  ones. This is what IEM does in steps 3a and 3b. The time to sort  $N$  values of  $\xi_i^T x$  is  $O(N \log N)$ , which is the overhead we need to pay in each iteration. After identifying  $B(x)$ , the simplex method selects  $\lambda_{B(x)}$  and adds it into the basis. In the primal problem, this is equivalent to constructing a constraint like (3) and adding it to the preceding linear program. This is exactly what IEM does in step 3c. Note that in each simplex pivot step, only one dual variable  $\lambda_{B(x)}$  that is zero will enter the basis, while the rest of the dual variables that are zeros will continue to be kept as zeros. Those dual variables that are continued to be kept as zeros correspond to non-binding constraints in the primal, and thus can be safely removed without affecting the solution of the next iteration.  $\square$

### 3. CVaR-constrained Stochastic Convex Programs

Extending IEM for problem CSCP is straightforward. In steps 3a, 3b and 3c, we sort  $g(\xi_i, x), i = 1, 2, \dots, N$  and add the following convex constraint to the preceding convex optimization problem

$$\frac{1}{N(1-\beta)} \sum_{i=K}^N g(\xi_{(i)}, x) \leq b$$

where the definition of  $\xi_{(i)}$  is similar to that in (1), or more precisely,

$$g(\xi_{(1)}, x) \leq g(\xi_{(2)}, x) \leq \dots \leq g(\xi_{(N)}, x).$$

**Proposition 3:** Constraint set

$$\frac{1}{N(1-\beta)} \sum_{i=1}^N g(\xi_i, x) I_A(\xi_i) \leq b, \quad \forall A \in \Phi \quad (9)$$

is a sample approximation to the CVaR constraint

$$\text{CVaR}_\beta[g(\xi, x)] \leq b$$

in problem CSCP.

**Proof:** Similar to the proof of proposition 1.  $\square$

**Proposition 4:** IEM solves the following sample approximation to problem CSCP,

$$\begin{aligned} & \max h(x) \\ \text{(SA-CSCP)} \quad & \frac{1}{N(1-\beta)} \sum_{i=1}^N g(\xi_i, x) I_A(\xi_i) \leq b, \quad \forall A \in \Phi \\ & x \in D \end{aligned}$$

**Proof:** Let  $x^*$  denote the optimal solution of problem SA-CSCP. If for  $\forall A \in \Phi$ , none of the constraints in constraint set (9) is binding at  $x^*$ , IEM obviously solves problem SA-CSCP as IEM steps 3, 4 and 5 have no effect on the final solution.

Now assume that at least one constraint in constraint set (9) is binding at  $x^*$ . We claim that as IEM progresses,  $h(x^t) > h(x^{t+1})$  for any iteration  $t$ , i.e., IEM keeps getting strictly better approximations to  $x^*$ , and the sequence of solutions is strictly monotonically decreasing across all iterations in IEM. We prove this by contradiction.

Suppose that in an iteration  $t+1$ , we get a solution  $x^{t+1}$ , such that  $h(x^t) < h(x^{t+1})$ .

Clearly,  $x^{t+1}$  can not lie in the feasible region of the convex problem IEM just solved in the preceding  $t$ -th iteration, because otherwise  $x^{t+1}$  would be the optimal solution to the  $t$ -th iteration problem. Therefore,  $x^{t+1}$  must be infeasible for the  $t$ -iteration problem. Since in iteration  $t+1$ , IEM keeps all binding constraints in the  $t$ -th iteration problem, and removes all non-binding constraints, it follows that  $x^{t+1}$  must violate some of the non-

binding constraints that have been removed. On the other hand,  $x^t$  obviously satisfies all these non-binding constraints as it is the optimal solution to the  $t$ -th iteration problem. Now we have two points  $x^t$  and  $x^{t+1}$ , lying on two sides of some non-binding constraints of the  $t$ -th iteration problem. It follows that there exist a point  $\bar{x}$  lying on the line segment connecting  $x^t$  and  $x^{t+1}$ , such that  $\bar{x}$  is feasible for the  $t$ -th iteration problem. Since  $h(x)$  is strictly concave, we have  $h(\bar{x}) > h(x^t)$  as well, which contradicts the assumption  $x^t$  is the optimal solution to the  $t$ -th iteration problem.

Given that  $h(x^t) > h(x^{t+1})$  is always true as IEM progresses, we know that IEM must terminate after a finite number of iterations. When it terminates at iteration  $t + 1$ , we have  $x^{t+1} = x^t$  (or more precisely,  $\|x^{t+1} - x^t\|$  is small enough). Now consider the constraint IEM added in iteration  $t + 1$ . It is a constraint in constraint set (9) indexed by  $B(x^t)$  (recall that  $B(x^t)$  is the dominant set induced by  $x^t$ ). Given that IEM solves the  $(t + 1)$ -th iteration problem with a constraint indexed by  $B(x^t)$ , and the optimal solution is equal to  $x^t$ , it follows that  $x^t$  satisfies each and every constraint in constraint set (9), as the constraint indexed by  $B(x^t)$  dominates all the other constraints. This means that  $x^t$  is the optimal solution to problem SA-CSCP.  $\square$

IEM solves a sequence of nonlinear programs iteratively. It is desirable to warm-start an iteration based on the solution obtained in the previous one. As IEM converges close to the optimal solution, warm-starting an iteration is especially helpful as the consecutive solutions are close to each other. The most widely used interior point methods for nonlinear programs are not suitable for this task, as our IEM algorithm converges to the optimal solution from the infeasible region. On the other hand, active-set methods are well appropriate to be embedded into IEM to warm-start new iterations. There are considerable new developments on active-set methods in recent years (see, e.g., Byrd and Waltz (2007), Chen et al. (2006), Byrd et al. (2004)). Though the current available implementations of active-set methods still can not compete with those of interior point methods, we expect them to improve significantly in the near future.

## 4. Bound on Sample Size

Problem SA-CSLP increasingly better approximates problem CSLP as the number of samples  $N$  grows. In this section, we give a bound on  $N$  such that the CVaR constraint holds with high probability at the optimal solution  $x^*$ . For the sake of presentation simplicity, we will derive the bound for problem CSLP. The exact same technique applies to problem CSCP.

To make sure that the CVaR constraint is satisfied with higher probability, we tighten the r.h.s. of the CVaR constraint, and solve the sample approximation problem corresponding to the tightened version. More specifically, we replace the CVaR constraint in problem CSLP by

$$\text{CVaR}_\beta(\xi^T x) \leq b - \varepsilon,$$

where  $\varepsilon$  is a small positive number, and solve the corresponding sample approximation problem. Let  $x^*$  denote the optimal solution, then we want the following inequality to hold

$$\Pr\{\text{CVaR}_\beta(\xi^T x^*) - b > 0\} < \delta, \quad (10)$$

where  $0 < \delta < 1$  is a small probability threshold.

When IEM terminates, we get an estimator of  $\text{CVaR}_\beta(\xi^T x^*)$ , i.e.,

$$\text{CVaR}_\beta(\xi^T x^*) \approx \frac{1}{N(1-\beta)} \sum_{i=K}^N \xi_{(i)}^T x^*,$$

where  $\xi_{(i)}$  and  $K$  are defined in definitions (1) and (2) respectively. Let  $H(x^*)$  denote the above estimator. There are two possibilities that can happen:  $H(x^*) = b - \varepsilon$  (the approximated CVaR constraint is binding), or  $H(x^*) < b - \varepsilon$  (the approximated CVaR constraint is not binding). We consider the binding case only, since it will produce a conservative bound on  $N$  that holds even for the non-binding case.

Manistre and Hancock (2005) showed that  $H(x^*)$  is an asymptotically unbiased estimator of  $\text{CVaR}_\beta(\zeta^T x^*)$ , and furthermore,  $H(x^*) - \text{CVaR}_\beta(\zeta^T x^*)$  is asymptotically normal distributed with mean 0 and variance  $\sigma^2$ , which can be approximated by

$$\sigma^2 \approx \frac{\sigma^2(\zeta_{(K)}^T x^*, \zeta_{(K+1)}^T x^*, \dots, \zeta_{(N)}^T x^*) + \beta(H(x^*) - \zeta_{(K)}^T x^*)^2}{N(1-\beta)}, \quad (11)$$

where  $\sigma^2(\zeta_{(K)}^T x^*, \zeta_{(K+1)}^T x^*, \dots, \zeta_{(N)}^T x^*)$  is the empirical variance of  $N - K + 1$  data points  $\zeta_{(K)}^T x^*, \zeta_{(K+1)}^T x^*, \dots, \zeta_{(N)}^T x^*$ . As  $N$  grows large, both the empirical variance and the term  $(H(x^*) - \zeta_{(K)}^T x^*)^2$  will converge to some constants. The former converges to the true variance of the tail distribution, and the later converges to the square of the true difference between the CVaR value and the VaR value. Therefore, we can simply equation (11) as

$$\sigma^2 \approx \frac{C_1 + \beta C_2}{N(1-\beta)}.$$

To satisfy inequality (10), we need

$$N > \frac{[\phi^{-1}(1-\delta)]^2 (C_1 + \beta C_2)}{\varepsilon^2 (1-\beta)}, \quad (12)$$

Where  $\phi^{-1}(\cdot)$  is the inverse cumulative distribution function of the standard normal distribution.

Monte Carlo methods usually converge at rate  $O(N^{-1/2})$ . Result (12) is consistent with that rate. Also note that  $N$  grows proportionally to  $1/(1-\beta)$ . This matches our intuition, as we actually only use  $N(1-\beta)$  data points to estimate CVaR. Importance sampling can reduce the number of sample points required. We will investigate this issue in another paper.

## 5. Computational Performance

In order to characterize the computational performance of IEM in practice, we conducted a comparison study on a set of CSLP problem instances that were generated using an instance generator. We describe the instance generator and report the observed computational performance of IEM, in comparison with the linear programming reformulation proposed by Rockafellar and Uryasev (2000).

The instance generator creates instances of CSLP with the following set of parameters. The number of columns (variables) was fixed at 30, while the number of rows, each of which is a CVaR constraint as described in the definition of CSLP, was varied from 2 to 200. The random constraint matrix (in the Left Hand Side, of the constraint set) was generated using a random seed, and the following equation, which generates a set of positive, random, constraint coefficients.

$$randseed = 8$$

$$A_{ij} = \max(0.1, \text{Normal}(\text{Uniform}(1,10), \text{Uniform}(5,10)))$$

The objective function coefficient vector was fixed; the constraint right-hand-side coefficient was fixed to take a value of 1 for each constraint.

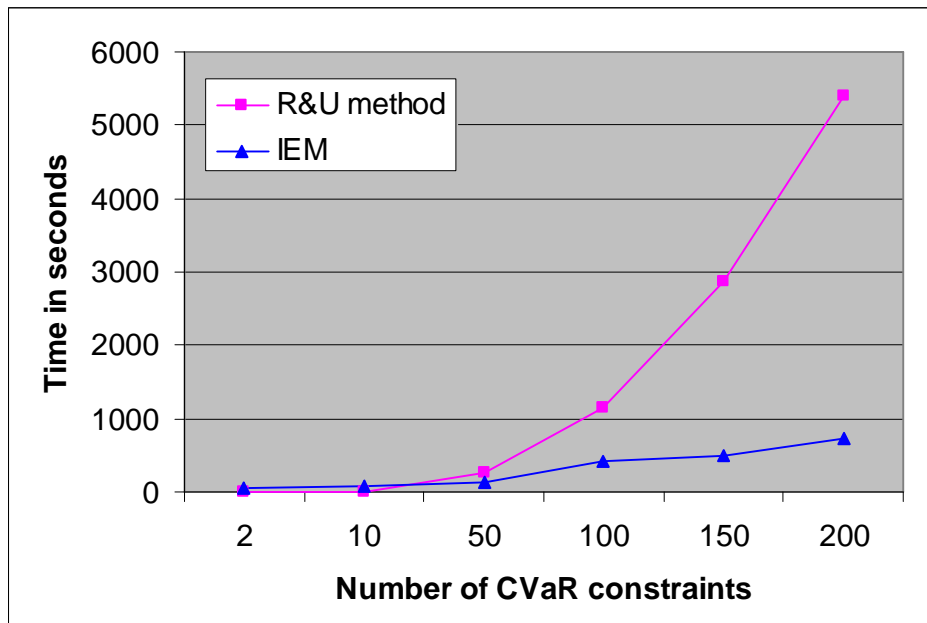
Note that the above instance generator will generate an instance of CSLP, upon specification of the number of rows, i.e. the number of CVaR constraints. We tested the computational performance of IEM, as well as the linear programming reformulation proposed by Uryasev and Rockafellar, on the following set of instances, which differ in the number of rows that range from 2 to 200, using a sample size of 1000 to approximate each problem instance. For the purposes of comparison, we implemented IEM using AMPL along with a C function call in order to enable the sorting procedure required in Step 3 of IEM. Further, the tolerance parameter for termination of IEM in Step 6 was chosen to be 1E-6. Similarly, we also implemented the linear programming reformulation of CSLP proposed by Uryasev and Rockafellar using AMPL. 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.

Instance	Number of Rows	Rockafellar and Uryasev (secs)	IEM (secs)
1	2	0.1	45
2	10	9.4	81.1
3	50	273.3	129.2
4	100	1148.4	420
5	150	2880.5	494.1
6	200	>5400, Terminated	730.2

**Table 1: Computational performance of IEM and Rockafellar and Uryasev’s method.**

The above results are compared in the following figure.



**Figure 1: Compare performance of IEM and Rockafellar and Uryasev’s method.**

The table shows that the performance of IEM is significantly superior, when the number of rows (constraints) is large. This is because, as mentioned earlier, the linear programming reformulation proposed by Rockafellar and Uryasev introduces a large number of additional variables and constraints, whose size scales linearly with the number of samples, for each original constraint in CSLP. In other words, if there are  $m$  original constraints in any of the above CSLP instances, the reformulation introduces



about  $1000m$  auxiliary variables and additional constraints, since the number of samples has been fixed at 1000 for the above instances. As shown in Figure 1, when  $m$ , which is the number of rows, increases from 2 to 200, the additional burden imposed by the increased complexity of the reformulation weighs in significantly on its computational performance.

On the other hand, IEM does not suffer from such an increase in complexity, because in each iteration of IEM, we solve a linear programming problem that is almost the same size and complexity as the original CSLP, in terms of number of constraints, and exactly the same size and complexity as the original CSLP, in terms of the number of variables. This aspect of IEM 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 sample average approximation quality, because the algorithmic complexity of IEM is relatively insensitive to the number of samples that are chosen in the sample average approximation.

## 6. Conclusion

IEM is an efficient algorithm for stochastic linear and convex programs with CVaR constraints. A fundamental characterization of coherent risk measures in general, and CVaR in specific, is that they can be represented as the worst-case expectation in some probability space (Artzner et al. (1999)). IEM exploits this property to sequentially approximate CVaR constraints and solve CVaR-constrained stochastic linear and convex problems simultaneously.

## References

Philippe Artzner, Freddy Delbaen, Jean-Marc Eber & David Heath 1999. Coherent Measures of Risk. *Mathematical Finance*, 9(3), p.203-228.

Richard H. Byrd & Richard A. Waltz, 2007. An Active-Set Algorithm for Nonlinear Programming Using Parametric Linear Programming. Technical Report, University of Southern California, Department of Industrial and Systems. Available at: [http://www.optimization-online.org/DB\\_HTML/2007/09/1768.html](http://www.optimization-online.org/DB_HTML/2007/09/1768.html)

Richard H. Byrd, Nicholas I. M. Gould, Jorge Nocedal & Richard A. Waltz, 2004. An Active-Set Algorithm for Nonlinear Programming Using Linear Programming and Equality Constrained Subproblems . *Mathematical Programming*, 100(1), p.27-48.

Lifeng Chen, Yongli Wang & Guoping He, 2006. A Feasible Active Set QP-Free Method for Nonlinear Programming. *SIAM Journal on Optimization*, 17(2), p.401-429.

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

Pavlo Krokmal, Jonas Palmquist & Stanislav Uryasev, 2002. Portfolio Optimization with Conditional Value-At-Risk Objective and Constraints. *The Journal of Risk*, 4(2), p.11-17.

B. John Manistre & Geoffrey H. Hancock, 2005. Variance of the CTE Estimator. *North American Actuarial Journal*, 9(2), p.129-156.

Arkadi Nemirovski & Alexander Shapiro, 2006. Convex Approximations of Chance Constrained Programs. *SIAM Journal on Optimization*, 17(4), p.969-996.

R. Tyrrell Rockafellar & Stanislav Uryasev, 2000. Optimization of Conditional Value-At-Risk. *The Journal of Risk*, 2(3), p.21-41.