RUTCOR
RESEARCH
REPORT

An Improved Cutting Plane method for the solution of Probabilistic Constrained Problem with Discrete Random Variables

Emre Yamangil[a] András Prékopa[b]

RRR 12-2012, February 2012

RUTCOR
Rutgers Center for
Operations Research
Rutgers University
640 Bartholomew Road
Piscataway, New Jersey
08854-8003
Telephone: 732-445-3804
Telefax: 732-445-5472
Email: rrr@rutcor.rutgers.edu
http://rutcor.rutgers.edu/∼rrr

[a]RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway NJ 08854-8003; (emreyamangil@gmail.com)
[b]RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway NJ 08854-8003; (prekopa@rutcor.rutgers.edu)

Rutcor Research Report
RRR 12-2012, February 2012

An Improved Cutting Plane method for the solution of Probabilistic Constrained Problem with Discrete Random Variables

Emre Yamangil, András Prékopa

Abstract. We consider a probabilistic constrained stochastic programming problem with discrete random variables. Two methods, a cutting plane and a column generation method have already been developed for the solution of the problem. In this paper we blend them together and obtain a method that is faster than the earlier ones. We also present a refined algorithm for the generation of p-level efficient points of a discrete distribution.

Keywords: Stochastic programming, probabilistic constraint, p-efficient point, column generation.

1 Introduction

In this paper we consider the probabilistic constrained stochastic programming problem,

$$\begin{array}{llcccl} \min & c^T x & & & & \\ \text{s.t.} & Ax & & & = & b \\ & P(Tx & \geq & \xi) & \geq & p \\ & x & & & \geq & 0, \end{array} \tag{1}$$

where $\xi = (\xi_1, \ldots, \xi_r)^T$ is a random vector with finite support. Problem (1) has a long history. A simpler formulation was given in[2], where probabilistic constraints were taken individually. Joint probabilistic constraints under independent random variables was formulated in [5]. The general case was first studied in [6].

The notion of a *p-level efficient point* of a discrete distribution was introduced in [7], where a dual method, for the solution of problem (1), was also proposed. Assuming the knowledge of all p-efficient points, a cutting plane method was proposed in [9]. First we enumerate all p-efficient points and then apply the cutting plane method that subsequently generates the facets of the cunvex hull of the p-efficient points. Not long after that a column generation method, to generate tight lower and upper bounds was presented in [3], making the numerical solvability of problem (1) more realistic, as the enumeration of all p-efficient points frequently led to intractability. The method was later extended to general convex programming with probabilistic constraints by the same authors [4]. Problem (1) was reformulated as a large scale mixed integer programming problem with knapsack constraints in [10]. Using bounds on the probability of the union of events, new valid inequalities for these mixed integer programming problems have been derived. A general framework to use probability bounds for the solution of probabilistic constrained stochastic programming problem was presented in [8]. We also mention [11], where the algorithm in [3] was further explained.

In this paper we

- present a refined method to generate p-efficient points,
- apply a column generation procedure that generates the p-efficient points simultaneously with the solution algorithm,
- present a different formulation of the cutting plane for the disjunctive hull, defined in [1].

The remainder of this paper is organized as follows. In Section 2 we give a brief description of pLEP's. In Section 3 we present an equivalent formulation of problem (1). Then, in Section 4 we show how to use outer approximation approach to generate cutting planes for the feasible region. We finalize the description of our algorithm with the column generation in Section 5. Numerical examples and concluding remarks are provided in Sections 6 and 7, respectively.

2 p-Level Efficient Points

Let $\mathcal{Z}$ denote the possible values of the random vector ξ. A point $z \in \mathcal{Z}$ is called p-level efficient point (pLEP) of the probability distribution of ξ, if $P(\xi \leq z) \geq p$ and there is no $y \in \mathcal{Z}$ satisfying $y \leq z$, $y \neq z$, $P(\xi \leq y) \geq p$. Let $\Omega = \{z^{(1)}, \ldots, z^{(N)}\}$ be the set of all pLEP's. For algorithmic enumeration of all pLEP's we use a modification of the method presented in [9]. Instead of calculating the smallest value subsequently for every random variable such that the probabilistic constraint is not violated, we enumerate a tree where every node corresponds to a permutation of $\{1, \ldots, r\}$. At every node of this tree we carry out the minimization in the order given by the permutation. Figure 1 visualizes the pLEP's on 3 dimensional space, grey shaded region represents $P(\xi \leq p_1) \geq p$. Also some of the facets from the convex hull of pLEP's are highlighted in Figure 1.

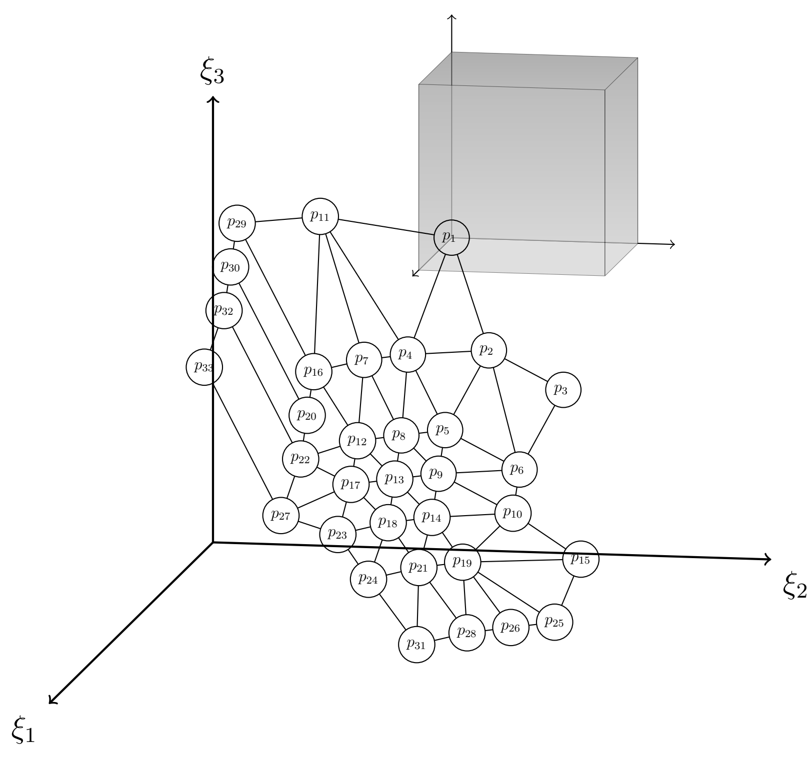

Figure 1: pLEP's on $\mathbb{R}^3$

We note that $O(Mr!) = O(Mr^r)$ time is required to calculate all pLEP's (M denotes the maximum range of the components of the random vector) which further emphasizes the importance of the use of a column generation method to solve problem (1).

3 Different Formulations

An equivalent formulation of problem (1) is the following,

$$\begin{array}{llll} \min & c^T x & & \\ \text{s.t.} & Ax & = & b \\ & Tx & \geq & z^{(i)} \quad \text{for at least one } i \in \{1, \ldots, N\} \\ & x & \geq & 0. \end{array} \tag{2}$$

Problem (2) is a disjunctive programming problem. Following Balas [1], we formulate a relaxation of Problem (2) as follows:

$$\begin{array}{llrll} \min & c^T x & & & \\ \text{s.t.} & Ax & & = & b \\ & Tx & -\sum_{i=1}^N z^{(i)}\lambda_i & \geq & 0 \\ & & \sum_{i=1}^N \lambda_i & = & 1 \\ & x & & \geq & 0 \\ & & \lambda & \geq & 0. \end{array} \tag{3}$$

In many cases it is impractical to enumerate all pLEP's and solve the above linear programming problem as it is proposed in [9]. In fact, the number of p-efficient points can be very large even if the number of components of ξ is moderate. For example, in our numerical example we have obtained more than 14000 p-efficient points for a 6-component ξ. In this paper, we propose a column generation and a cutting plane algorithm, where instead of enumerating all pLEP's, our algorithm finds an initial subset of pLEP's and at every iteration either generates a new pLEP or a cutting plane, which cuts off a part of the pLEP-free region.

4 Cutting Plane Algorithm

We adapt the cut generation strategy of [9] with a new formulation of the cut. First we convert problem (3) to the following:

$$\begin{array}{llllrrll} \min & c^T x & & & & & & \\ \text{s.t.} & Ax & & & & & = & b \\ & Tx & -z & & & -u & = & 0 \\ & & z & -\sum_{i=1}^N z^{(i)}\lambda_i & & & = & 0 \\ & & & \sum_{i=1}^N \lambda_i & & & = & 1 \\ & x & & & & & \geq & 0 \\ & & & \lambda & & & \geq & 0 \\ & & & & & u & \geq & 0. \end{array} \tag{4}$$

We generate facets to the set of feasible solutions of the disjunctive polyhedron (Balas, 1975).

$$K = \{z = \sum_{i=1}^N z^{(i)}\lambda_i | \sum_{i=1}^N \lambda_i = 1, \lambda \geq 0\}. \tag{5}$$

Let $\Phi' = \{z^{(1)}, \dots, z^{(p)}\}$ be an initially generated subset of pLEP's with the following property. Let $\overline{z}_j = \max\{z_j^{(i)} | i = 1, \dots, N\}, \forall j = 1, \dots, r$ and $\underline{z}_j = \min\{z_j^{(i)} | i = 1, \dots, N\}, \forall j = 1, \dots, r$. Then $\forall z \in \Phi$ we have $\underline{z} \le z \le \overline{z}$. Clearly such a subset can be generated efficiently. Also introduce $z^{(0)} = \frac{1}{p} \sum_{i=1}^{p} z^{(i)}$, the arithmetic mean of Φ.

Since the set Φ can be concentrated on an affine manifold with dimension smaller than r, we append the following constraint to the constraints in problem (4):

$$w_l^T (z - z^0) = 0, \quad l = 1, \dots, h,$$

where $[w_1, \dots, w_h]$ is a basis of the vectors $z^{(1)} - z^{(0)}, \dots, z^{(p)} - z^{(0)}$.

To create the cutting plane in the kth iteration, we formulate the problem:

$$\begin{array}{lrcl} \min & & e^T \lambda & \\ \text{s.t.} & \sum_{i=1}^{p} (z^{(i)} - z^{(0)}) \lambda_i & = & z^k - z^0 \\ & \lambda & \ge & 0, \end{array} \tag{6}$$

where $e = (1, \dots, 1)^T$ and (x^k, u^k, z^k) is the current optimal solution. Let α be the optimum value of problem (6). If $\alpha \le 1$, then we terminate the procedure. The current optimal solution (x^k, u^k, z^k) is an optimal solution to problem (4). Else we generate a cut as follows.

Let $\left\{ z^{(i_1)} - z^{(0)}, \dots, z^{(i_{r-h})} - z^{(0)} \right\}$ be an optimal basis to problem (6). Then we find a $w \neq 0$ such that,

$$\begin{aligned} w^T w_i = 0, &\quad i = 1, \dots, h \\ w^T (z^{(i_j)} - z^{(i_1)}) = 0, &\quad j = 2, \dots, r - h. \end{aligned}$$

If w is determined in such a way that $w^T (z^{(0)} - z^{(i_1)}) > 0$, then we supplement the following cut to our problem:

$$w^T z - w^T z^{(i_1)} \ge 0. \tag{7}$$

5 Column Generation Method

We incorporate the column generation method proposed by [3]. In that paper, instead of generating cuts with the method of [9], the authors successively generate upper and lower bounds to the probabilistic constrained problem. When these bounds coincide the algorithm terminates. Here we propose to generate columns that are better approximates to the disjunctive hull as follows.

Let $\left\{ z^{(i_1)} - z^{(0)}, \dots, z^{(i_{r-h})} - z^{(0)} \right\}$ be an optimal basis to problem (6) and π the corresponding dual vector. Since π is dual feasible, we have

$$(z^{(i)} - z^0)^T \pi \le 1, \quad \forall i \in \Phi'. \tag{8}$$

			Dimension r			
			3	4	5	6
	Enumeration	# of pLEPs	33	187	1461	14856
		pLEP time	0.0	0.3	16.2	5638.7
Algorithm	Initialization	# of initial pLEPs	6	24	120	709
		Initial pLEP time	0.0	0.0	0.0	0.3
	Execution	# of gen. pLEPs	7	35	60	85
		# of gen. Rows	4	10	11	9
		CPU time	0.4	1.9	3.9	9.1

Table 1: Computational Study

We look for a new pLEP, $z' \in \Phi$ subject to the condition that (8) is not satisfied, i.e. $(z' - z^0)^T \pi > 1$. Let $\overline{z}_j = \max\{z_j^{(k)} | k = i_1, \ldots, i_{r-h}\}, \forall j = 1, \ldots, r$ and $\underline{z}_j = \min\{z_j^{(k)} | k = i_1, \ldots, i_{r-h}\}, \forall j = 1, \ldots, r$. We can attain such a z' by solving the following problem,

$$\begin{array}{llcl} \max & \pi^T z & & \\ \text{s.t.} & P(z \geq \xi) & \geq & p \\ & z & \leq & \overline{z} \\ & z & \geq & \underline{z} \end{array} \tag{9}$$

If we assume that the random variables are independent, we can further simplify problem (9) in the following way.

$$\begin{array}{lrcll} \max & \sum_{i=1}^{r} \sum_{k=\underline{z}_i}^{\overline{z}_i} k\pi_i \delta_{ik} & & & \\ \text{s.t.} & \sum_{i=1}^{r} \sum_{k=\underline{z}_i}^{\overline{z}_i} \log(P(k \geq \xi_i))\delta_{ik} & \geq & \log(p) & \\ & \sum_{k=\underline{z}_i}^{\overline{z}_i} \delta_{ik} & = & 1, & \forall i \\ & \delta_{ik} & \in & \{0,1\}, & \forall i,k \end{array} \tag{10}$$

If z' does not satisfy (8) then we let $\Phi' \leftarrow \Phi' \cup \{z'\}$, else we generate the cut as described in Section 4.

6 Numerical Results

We coded all our algorithms in MATLAB 7.11.0. Also we employ the use of linear programming and binary integer programming solver provided by MATLAB Optimization Toolbox V5.1. We provide the output of our computational test in Table 1.

Computational study strongly encourages the use of column generation cutting plane method, as we can see in cases of even modestly sized problems. We frequently observe that the enumeration phase is exhaustive and it quickly becomes intractable. However, the initial subset of pLEP's are easy to obtain to start the algorithm, which quickly generates the required subset of pLEP's and cuts into the feasible region. The matrices $A \in \mathbb{R}^{5\times15}$, $T \in \mathbb{R}^{r\times15}$, $c \in \mathbb{R}^{15\times1}$ and $b \in \mathbb{R}^{5\times1}$ have been uniformly generated between 0 and 10.

The components of the random vector $\xi = (\xi_1, \dots, \xi_r)^T$ are independent and have Poisson distribution in all numerical examples and $p = 0.9$.

7 Concluding Remarks and Future Research

Quite often the number of pLEP's grows exponentially, since the complexity of the proposed method of enumeration is $O(Mr^r)$. This explains why enumerative approaches quickly become intractable due to the size of the problem. The Column Generation / Cutting Plane method presented in this paper gives us an elegant solution method, where we just need to enumerate a small subset of pLEP's to start the algorithm and we need to generate a subset of them, which points to the gradient of the objective function.

As regards future work first we mention that the Column Generation procedure is NP-*hard* (similarly to the Knapsack Problem). A more efficient way of generating the columns may be necessary for solving large scale stochastic programming problems. Also, the proposed method heavily depends on the initially generated subset of pLEP's which is also of exponential complexity. Increasing the efficiency of this method can greatly contribute to the performance of our algorithm. However, we realize that solving the Disjunctive Programming problem is NP-*hard*. Therefore a great increase in efficiency is unlikely.

References

[1] E. Balas. Disjunctive programming: Cutting planes from logical conditions. *Nonlinear Programming*, 2:279–312, 1975.

[2] A. Charnes, W.W. Cooper, and G.H. Symonds. Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management Science*, 4:235–263, 1958.

[3] D. Dentcheva, A. Prékopa, and A. Ruszczyński. Concavity and efficient points of discrete distributions in probabilistic programming. *Mathematical Programming Series A*, 89:55–77, 2000.

[4] D. Dentcheva, A. Prékopa, and A. Ruszczyński. On convex probabilistic programming with discrete distributions. *Nonlinear Analysis*, 47:1997–2009, 2001.

[5] L.B. Miller and H. Wagner. Chance-constrained programming with joint constraints. *Operations Research*, 13:930–945, 1965.

[6] A. Prékopa. On probabilistic constrained programming. In *Proceedings of the Princeton Symposium on Mathematical Programming*, pages 113–138. Princeton University Press, 1970.

[7] A. Prékopa. Dual method for a one-stage stochastic programming problem with random rhs obeying a discrete probability distribution. *Zeitschrift f ur Operations Research*, 34:441–461, 1990.

[8] A. Prékopa. The use of discrete moment bounds in probabilistic constrained stochastic programming models. *Annals of Operations Research*, 85:21–38, 1999.

[9] A. Prékopa, B. Vizvári, and T. Badics. Programming under probabilistic constraint with discrete random variable. *New Trends in Mathematical Programming*, pages 235–255, 1998.

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

[11] B. Vizvári. The integer programming background of a stochastic integer programming algorithm of dentcheva, prékopa, ruszczyński. *Optimization Methods and Software*, 7:543–559, 2002.