A Branch-Reduce-Cut Algorithm for the Global Optimization of Probabilistically Constrained Linear Programs*

Myun-Seok Cheon, Shabbir Ahmed† and Faiz Al-Khayyal

School of Industrial & Systems Engineering

Georgia Institute of Technology, 765 Ferst Drive, Atlanta, GA 30332.

September 23, 2004

Abstract

We consider probabilistic constrained linear programs with general distributions for the uncertain parameters. These problems generally involve non-convex feasible sets. We develop a branch and bound algorithm that searches for a global solution to this problem by successively partitioning the non-convex feasible region and by using bounds on the objective function to fathom inferior partitions. This basic algorithm is enhanced by domain reduction and cutting plane strategies to reduce the size of the partitions and hence tighten bounds. The proposed *branch-reduce-cut* algorithm exploits the monotonicity properties inherent in the problem, and requires solving linear programming subproblems. We provide convergence proofs for the algorithm. Some illustrative numerical results involving problems with discrete distributions are presented.

1 Introduction

Various applications in reliability and risk management (cf.[15]) give rise to probabilistically-constrained linear programs (PCLP) of the following form

$$\min_{x} \quad c^T x \tag{1}$$

$$\text{s.t.} \quad Ax = b \tag{2}$$

$$\mathbb{P}\{Tx \geq \xi(\omega)\} \geq \alpha \tag{3}$$

$$x \geq 0. \tag{4}$$

In the above model, $x \in \mathbb{R}^n$ is a vector of decision variables; the parameters $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$ and $T \in \mathbb{R}^{m \times n}$ represent deterministic problem data; ω is a random vector from the probability space $(\Omega, \Sigma, \mathcal{P})$ and $\xi : \Omega \mapsto \mathbb{R}^m$ represent stochastic right-hand-side parameters; $\mathbb{P}\{S\}$ denotes the probability of the event $S \in \Sigma$ under the probability measure $\mathcal{P}$; and $\alpha \in (0, 1)$ is a scalar. The probabilistic constraint (3) requires that the inequalities $Tx \geq \xi(\omega)$, involving random data, hold with a probability of at least α.

Beginning with the seminal work of Charnes and Cooper [4], many different solution techniques have been proposed for different versions of PCLP. Most of the existing approaches rely on specific assumptions on the distribution of the stochastic parameters that render convex the feasible region defined by the probabilistic constraints (3). For example, if the stochastic parameters have a continuous log-concave distribution, then PCLP is guaranteed to have a convex feasible set [13], and hence may be solvable with standard convex programming techniques (see [15] for a comprehensive review). For general distributions, in particular for

---

†Corresponding author. E-mail: `sahmed@isye.gatech.edu`

discrete distributions, the feasible region defined by (3) is non-convex. This non-convexity is illustrated with the following simple example from [17] (a similar example is presented in [20]).

Example 1: Consider the following PCLP:

$$\min_{x_1, x_2} \quad c_1 x_1 + c_2 x_2$$

$$\text{s.t.} \quad \mathbb{P}\left\{ \begin{array}{c} x_1 + x_2 \geq \xi_1 \\ x_1 + 3x_2 \geq \xi_2 \end{array} \right\} \geq 0.5$$

$$x_1, \; x_2 \geq 0,$$

where ξ_1 and ξ_2 are random variables with the joint distribution $\mathbb{P}\{\xi_1 = 2, \xi_2 = 4\} = 0.5$ and $\mathbb{P}\{\xi_1 = 3, \xi_2 = 0\} = 0.5$. The feasible region is given by union *of the polyhedra*

$$P_1 = \{(x_1, x_2) \in \mathbb{R}_+^2 : \; x_1 + x_2 \geq 2, x_1 + 3x_2 \geq 4\} \; \text{ and}$$
$$P_2 = \{(x_1, x_2) \in \mathbb{R}_+^2 : \; x_1 + x_2 \geq 3, x_1 + 3x_2 \geq 0\},$$

and is clearly non-convex.

For discrete distributions, a PCLP can be immediately reformulated as a mixed-integer linear program (milp) (see [15]). Ruszczyński [16] has developed specialized cutting planes for this milp reformulation, and embedded these within a branch-and-cut algorithm. Unless the random variables are independent, the milp reformulation involves a huge number of binary variables – one for every possible joint realization of the random parameters – and may be computationally impractical in general. An improved milp formulation for the PCLP can be constructed if the set of p-efficient points (PEPs) can be identified. A point $z \in \mathbb{R}^m$ is said to be a PEP for the random variable $\xi(\omega)$ if $\mathbb{P}\{z \geq \xi(\omega)\} \geq \alpha$ and there is no y such that $y \leq z$, $y \neq z$ and $\mathbb{P}\{y \geq \xi(\omega)\} \geq \alpha$[14]. For discrete distributions with certain concavity properties, Dentcheva et al. [5, 6] have developed efficient convex programming relaxations for PEP-based milp formulations for PCLPs. Such relaxations have been used within exact branch and bound algorithms for some classes of probabilistically constrained milps [1, 2]. Sen [19] has suggested convex relaxations for PCLPs with general discrete distributions using disjunctive programming techniques.

In this paper, we develop an algorithm for obtaining global optimal solutions to PCLPs. Unlike prior work, we do not make any concavity or continuity assumptions on the underlying distributions. The proposed algorithm is a branch and bound scheme that searches for a global solution by successively partitioning the non-convex feasible region and by using bounds on the objective function to fathom inferior partitions. The basic scheme is enhanced by domain reduction and cutting plane strategies to reduce the size of the partitions and hence tighten bounds. We refer to this strategy as branch-reduce-cut. The proposed method exploits the monotonicity properties inherent in the problem, and requires solving linear programming subproblems. Convergence analysis of the algorithm in case of discrete and continuous distributions is provided. We also present numerical results for PCLPs involving discrete distributions and demonstrate that the proposed algorithm is significantly superior to a straight-forward milp approach.

The remainder of this paper is organized as follows. In Section 2, we reformulate PCLP to reveal the inherent monotonicity in the problem. The proposed branch-reduce-cut algorithm is developed in Section 3, and its convergence analysis is presented in Section 4. Finally, in Section 5 we present some numerical results with the proposed algorithm on randomly generated PCLP instances involving discrete distributions.

2 Problem reformulation and structural properties

Consider the following problem:

$$\min_{y} \quad f(y) \tag{5}$$

$$\text{s.t.} \quad y \in G \cap H, \tag{6}$$

where $y \in \mathbb{R}^m$ is a vector of decision variables; $f : \mathbb{R}^m \mapsto \mathbb{R} \cup \{-\infty, +\infty\}$ is the linear programming value function

$$f(y) = \min_{x}\{c^T x : \; Ax = b, Tx \geq y, x \geq 0\} \tag{7}$$

(we let $f(y) = -\infty$ and $f(y) = +\infty$ when the linear program (7) is unbounded and infeasible, respectively); the set G is the set of y's for which the linear program (7) is feasible, i.e.,

$$G = \{y \in \mathbb{R}^m : \ f(y) < +\infty\}; \tag{8}$$

and the set H is defined as

$$H = \{y \in \mathbb{R}^m : \ F(y) \geq \alpha\}, \tag{9}$$

where $F : \mathbb{R}^m \mapsto [0,1]$ is the cumulative density function of the random vector $\xi(\omega)$, i.e., $F(y) = \mathbb{P}\{y \geq \xi(\omega)\}$.

The following result establishes the equivalence between PCLP (1)-(4) and the problem (5)-(6). The proof is straight-forward and is omitted.

Proposition 1.

(i) If x^* is an optimal solution of the PCLP (1)-(4), then $y^* = Tx^*$ is an optimal solution of (5)-(6), and both problems have the same optimal objective function value.

(ii) If y^* is an optimal solution of (5)-(6), then $x^* \in \mathrm{argmin}\{c^T x : \ Ax = b, Tx \geq y^*, x \geq 0\}$ is an optimal solution of the PCLP (1)-(4), and both problems have the same optimal objective function value.

Using Proposition 1, we consider solving the reformulated PCLP (5)-(6) throughout the remainder of this paper. Henceforth, the acronym PCLP will refer to the reformulation (5)-(6). The following assumptions are required to ensure that PCLP is well-defined.

Assumption 1. *The set of dual solutions to the linear program (7) is non-empty, i.e.,*

$$\{(\pi, \rho) \in \mathbb{R}^{p+m} : \ \pi A + \rho T \leq c, \rho \geq 0\} \neq \emptyset.$$

By weak duality, the above assumption guarantees that $f(y) > -\infty$ for all y.

Assumption 2. *The constraint (6) in PCLP can be replaced by*

$$y \in G \cap H \cap [y^L, y^U],$$

for some $y^L \leq y^U$.

The above assumption is required to make the feasible region bounded.

Proposition 2 (see, e.g., [3]). *Under Assumption 1,*

(i) *the set G is non-empty, closed and convex, and*

(ii) *the function f is continuous, piece-wise linear, convex, and non-decreasing over G.*

Proposition 3. *Under Assumptions 1 and 2, if the feasible region of the PCLP (5)-(6) is non-empty, then there exists an optimal solution.*

Proof. The set H is closed due to the upper semi-continuity of the cumulative density function F. Thus the feasible region of PCLP is compact, and the objective function is continuous. The result then follows from the Weirstrass Theorem. $\square$

Note that the set G satisfies the following property

$$x \leq y \text{ and } y \in G \Rightarrow x \in G.$$

Such a set is called a *normal* set [23]. Owing to the non-decreasing property of F, the set H satisfies

$$x \geq y \text{ and } y \in H \Rightarrow x \in H.$$

Such a set is called a *reverse normal* set [23]. Thus the problem (5)-(6) involves minimizing a non-decreasing function over the intersection of a normal and a reverse normal set. Such problems belong to the class of non-convex monotonic optimization problems recently studied by Tuy [23], Li et al. [10] and Toh [22]. Following are some important properties adapted to our setting (the proofs follow immediately from the general results in [23]).

Proposition 4. *Under Assumptions 1 and 2,*

(i) *if the problem is feasible then $y^L \in G$ and $y^U \in H$;*

(ii) *if $y^L \in H$, then either y^L is an optimal solution or the problem is infeasible;*

(iii) *if $y^L \notin H$ and the problem is feasible, then there exists an optimal solution on the relative boundary of H;*

(iv) *if $\widehat{y} \in [y^L, y^U]$ is on the relative boundary of H, then there cannot be a solution that is better (i.e., feasible with a smaller objective value) in the sets $Q_A := \{y : y^L \le y < \widehat{y}\}$ and $Q_B := \{y : \widehat{y} \le y \le y^U\}$. Consequently, the sets Q_A and Q_B can be removed from further consideration.*

In the case of a (finite) discrete distribution of the random vector $\xi(\omega)$, PCLP has some additional properties. Suppose $\xi(\omega)$ has K possible realizations $\{\xi^1, \ldots, \xi^K\}$ with probabilities $\{p^1, \ldots, p^K\}$. Let

$$\Xi_j = \cup_{k=1}^{K}\{\xi_j^k\} \text{ for } j = 1, \ldots, m \text{ and } \mathcal{C} = \prod_{j=1}^{m} \Xi_j. \tag{10}$$

Note that the set $\mathcal{C}$ is finite.

Lemma 1. *If $\xi(\omega)$ has a discrete distribution, then for any $y \in \mathbb{R}^m$ such that $y \in H$, there exists $\widehat{y}$ such that*

$$\widehat{y} \le y, \ \widehat{y} \in \mathcal{C} \text{ and } \widehat{y} \in H.$$

Proof. Let $\mathcal{K}(y) = \{k \in \{1, \ldots, K\} : \xi^k \le y\}$, then $y \in H \Leftrightarrow \sum_{k \in \mathcal{K}(y)} p^k \ge \alpha$. Let $\widehat{y}_j = \max_{k \in \mathcal{K}(y)}\{\xi_j^k\}$ for $j = 1, \ldots, m$. Then $\widehat{y} \le y$, $\widehat{y} \in \mathcal{C}$, and $\widehat{y} \ge \xi^k$ for all $k \in \mathcal{K}(y)$, thus $\widehat{y} \in H$. $\qquad\square$

Proposition 5. *If $\xi(\omega)$ has a discrete distribution and PCLP has an optimal solution, then there exists an optimal solution $y^* \in \mathcal{C}$.*

Proof. Let $y' \in H \cap G$ be any optimal solution of PCLP, then by Lemma 1, there exists $y^* \in \mathcal{C}$ such that $y^* \le y'$ and $y^* \in H$. Since G is a normal set, we have $y^* \in G$, thus y^* is feasible. By the monotonicity of the objective function, $f(y^*) \le f(y')$, thus y^* is also an optimal solution. $\qquad\square$

By virtue of Proposition 5, in case of discrete distributions, we can restrict our search of the optimal solution to the finite set $\mathcal{C}$ intersected with $G \cap H$.

In the following section, we develop a branch-and-bound algorithm for solving PCLP by exploiting the properties outlined above. In addition to revealing the monotonicity properties, the reformulation (5)-(6) has the added advantage (over the original formulation (1)-(4)) that the problem dimension is m rather than n, and often $m < n$. We conclude this section with an example to illustrate the properties of the reformulation.

Example 2: Consider the following PCLP:

$$\min_{x_1, x_2} \quad -x_1 - 2x_2$$

$$s.t. \quad \mathbb{P}\left\{ \begin{array}{c} -x_1 - x_2 \ge \xi_1 \\ x_1 + \frac{1}{2}x_2 \ge \xi_2 \end{array} \right\} \ge 0.5$$

$$x_1, x_2 \ge 0,$$

where ξ_1 and ξ_2 are random variables with a discrete distribution consisting of the 10 realizations given in Table 1.

It can be verified that the three p-efficient points of the distribution are $(-5, 3)$, $(-3, 2)$ and $(0, 1.5)$. The feasible region of PCLP is then give by the union of the polyhedra

$$P_1 = \{(x_1, x_2) \in \mathbb{R}_+^2 : -x_1 - x_2 \ge -5, \ x_1 + \frac{1}{2}x_2 \ge 3\},$$

$$P_2 = \{(x_1, x_2) \in \mathbb{R}_+^2 : -x_1 - x_2 \ge -3, \ x_1 + \frac{1}{2}x_2 \ge 2\} \quad and$$

$$P_3 = \{(x_1, x_2) \in \mathbb{R}_+^2 : -x_1 - x_2 \ge 0, \ x_1 + \frac{1}{2}x_2 \ge 1.5\},$$

k	1	2	3	4	5	6	7	8	9	10
ξ_1^k	-7	-6	-6	- 5.5	-5	-3	-3	-2	0	1
ξ_2^k	1.5	1	2	3	1	1	5.5	3	1	2
p^k	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

Table 1: Distribution of (ξ^1, ξ^2)

and is illustrated in Figure 1 (a).
The reformulation of the problem is

$$\min_{y_1,y_2} \quad f(y_1, y_2)$$
$$s.t. \quad (y_1, y_2) \in G \cap H \cap [y^L, y^U],$$

where $f(y_1, y_2) = \min_{x_1,x_2}\{-x_1 - 2x_2 : \; -x_1 - x_2 \geq y_1, \; x_1 + \frac{1}{2}x_2 \geq y_2, \; x_1 \geq 0, \; x_2 \geq 0\}$, $G = \{(y_1, y_2) : \; f(y_1, y_2) < +\infty\}$, $H = \{(y_1, y_2) : \; F(y_1, y_2) \geq \alpha\}$, $y^L = (-7, 1.5)^T$, $y^U = (1, 5.5)^T$. *The feasible region in the reformulated problem is shown in Figure 1 (b). The points of intersection of the dotted grid constitute the finite set* $\mathcal{C}$.
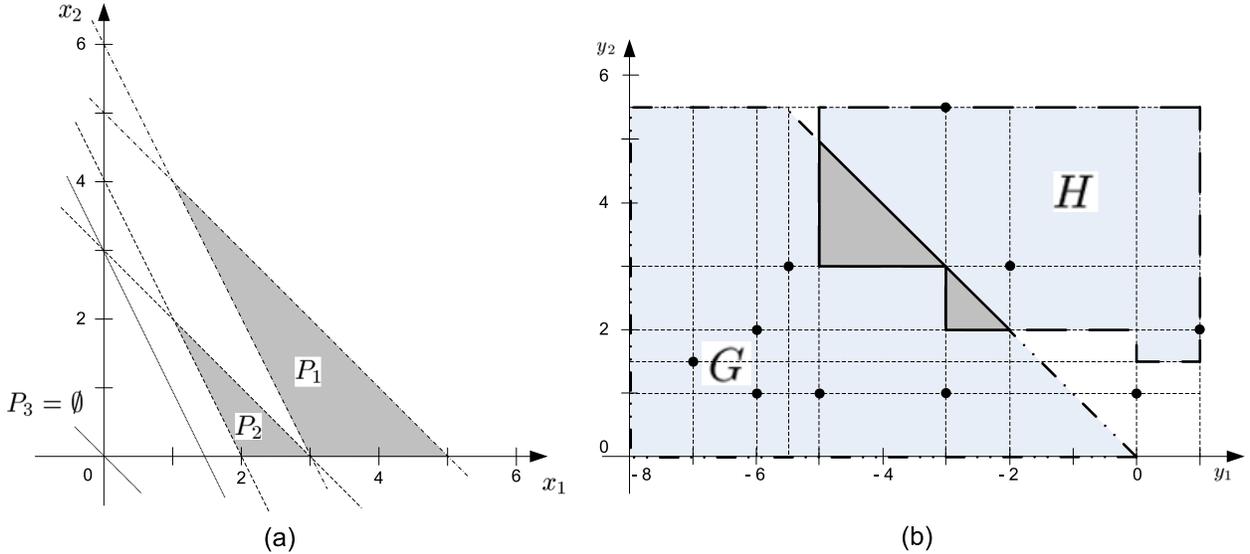

Figure 1: (a) The feasible region in the x-space, (b) The feasible region in the y-space.

3   A Branch-Reduce-Cut algorithm

Owing to the non-convexity of the set H (recall that the normal set G and the function f are convex), PCLP is a non-convex optimization problem. Initially proposed to deal with non-convexities arising from discreteness [9], the branch-and-bound scheme has since evolved into a general purpose global optimization technique for a wide class of non-convex problems [7, 8, 21]. The scheme proceeds by recursively partitioning (branching) the feasible region in search of a global optimal solution. For a given partition, bounds on the optimal objective value, based on efficiently solvable relaxations of the problem, are used to decide whether to examine the partition further or to fathom it, i.e., discard it from further consideration (bounding). For

example, in the case of milps, linear programming relaxation based bounding and integrality-based branching rules are used (cf. [12]). For general non-convex non-linear problems, developing tight and tractable (convex) relaxations pose a crucial challenge [21]. Furthermore, since the feasible region is typically (semi)continuous, special care has to be taken to decide how a particular partition is to be further refined in order to ensure convergence.

A variety of branch-and-bound schemes has been proposed for various classes of non-convex optimization problems. These methods rely on exploiting the analytical form of the objective and constraints of the problem in order to develop convex relaxations. Several sophisticated global optimization software that automatically construct such relaxations and use these within enhanced branch-and-bound search are also available [11, 18]. Unfortunately, none of these existing global optimization methods can be used for PCLP since neither the objective function f, nor the set G, and in some cases, the set H defining the constraints is available in closed analytic form.

In this section, we exploit the monotonicity properties of PCLP outlined in the previous section to develop a specialized branch-and-bound algorithm. The algorithm recursively partitions the hyper-rectangle $[y^L, y^U]$, that contains the feasible region, into smaller and smaller hyper-rectangles. The monotonicity of the objective function f guarantees that for any hyper-rectangular partition $[a, b]$, $f(a)$ is a valid lower bound and can be used for fathoming. The monotonicity property of the set H is used for branching and to find feasible solutionss. This basic branch-and-bound scheme is enhanced by cutting plane strategies to successively approximate the set G and the set of optimal solutions contained in G. The cutting planes along with the monotonicity of the set H is used within a domain reduction step to reduce the size of the current hyper-rectangle. We refer to this algorithm as branch-reduce-cut and provide its detailed description next.

In the following description, $\mathcal{R}$ denotes a hyper-rectangular partition in $\mathbb{R}^m$, i.e. $\mathcal{R} = [a, b]$ with $a, b \in \mathbb{R}^m$ and $a \leq b$; $v(\mathcal{R}) = ||b - a||_\infty$ is a measure of the size of $\mathcal{R}$; $LB(\mathcal{R})$ denotes the lower bound on the optimal value over $\mathcal{R}$; UB denotes a global upper bound on the optimal value; $\mathcal{L}$ is a list of un-fathomed hyper-rectangles; y^* denotes the best candidate solution; $\overline{G}$ denotes a normal set approximation of the set of optimal solutions contained in G (the algorithm will ensure that this approximation is a normal set); and ϵ denotes a pre-specified tolerance. Recall that, given a vector $y \in \mathbb{R}^m$, the function $f(y)$ is evaluated by solving the linear program (7), and if $f(y) = +\infty$ then $y \notin G$. We assume that we have an oracle to check, given $y \in \mathbb{R}^m$, whether $y \in H$, i.e., we can evaluate the cumulative density function F of the random vector $\xi(\omega)$ and check if $F(y) \geq \alpha$. An algorithmic description of the proposed branch-cut-reduce scheme is presented in Figure 2. The key steps are highlighted in italics, and are described below.

The algorithm starts by considering the initial parition $[y^L, y^U]$ given by the bounds on the variables and checks the necessary feasibility condition in Proposition 4(i) and the sufficient optimality condition in Proposition 4(ii). If neither of these two conditions are satisfied, the algorithm starts its main loop.

3.1 Selection and branching

The first main step is to select a partition from the list $\mathcal{L}$ of unfathomed partitions. This choice is based upon the least-lower bound rule (line 2 of the main loop in Figure 2). This guarantees that the bounding process is *bound improving* [8].

After the selection step, the selected hyper-rectangle is partitioned into two hyper-rectangles. In the case of a continuous distribution, the branching rule is to bisect the longest edge. That is, given $\mathcal{R} = [a, b]$ of size $v(\mathcal{R}) > 0$ with the longest edge index $\hat{\jmath} \in \operatorname{argmax}_{j=1,\ldots,m}\{b_j - a_j\}$, $\mathcal{R}$ is partitioned into $\mathcal{R}_1 = [a, b - e^{\hat{\jmath}} v(\mathcal{R})/2]$ and $\mathcal{R}_2 = [a + e^{\hat{\jmath}} v(\mathcal{R})/2, b]$, where $e^j \in \mathbb{R}^m$ is the j-th unit vector. The longest-edge bisection rule guarantees that the branching process is *exhaustive* [8].

Since, in the discrete case we can restrict our search to the set $\mathcal{C}$ (Proposition 5), our branching rule is as follows. Given a hyper-rectangle $\mathcal{R} = [a, b]$ of size $v(\mathcal{R}) > 0$ suppose the longest edge has an index $\hat{\jmath} \in \operatorname{argmax}_{j=1,\ldots,m}\{b_j - a_j\}$. Let $\Xi_{\hat{\jmath}} = \{\sigma_1, \sigma_2, \ldots, \sigma_K\}$ with $\sigma_k < \sigma_{k+1}$ for all k. Choose $\{\sigma_p, \sigma_q\} \in \Xi_{\hat{\jmath}}$ such that $\sigma_p < (a_{\hat{\jmath}} + b_{\hat{\jmath}})/2 \leq \sigma_q$, and let $b_1 = b - (b_{\hat{\jmath}} + \sigma_p)e^{\hat{\jmath}}$ and $a_2 = a + (a_{\hat{\jmath}} + \sigma_q)e^{\hat{\jmath}}$. Then $\mathcal{R}$ is partitioned into $\mathcal{R}_1 = [a, b_1]$ and $\mathcal{R}_2 = [a_2, b]$. Note that, if $\mathcal{R} = [a, b]$ is such that

$$a, b \in \mathcal{C} \tag{11}$$

Initialization:

1: **if** $f(y^L) = +\infty$ or $y^U \notin H$ **then**
2:    STOP the problem is infeasible
3: **else** $\{f(y^L) < +\infty$ and $y^U \in H\}$
4:    **if** $y^L \in H$ **then**
5:       STOP y^L is an optimal solution
6:    **else**
7:       set $\mathcal{R} = [y^L, y^U]$, $\mathcal{L} = \{\mathcal{R}\}$, $LB(\mathcal{R}) = f(y^L)$, $UB = +\infty$, $y^* = \emptyset$, and $\overline{G} = \mathbb{R}^m$
8:    **end if**
9: **end if**

Main loop:

1: **while** $\mathcal{L} \neq \emptyset$ **do**
2:    *select* $\mathcal{R} \in \mathcal{L}$ such that $LB(\mathcal{R}) = \min_{\mathcal{R}' \in \mathcal{L}}\{LB(\mathcal{R}')\}$
3:    *branch* $\mathcal{R}$ into $\mathcal{R}_1$ and $\mathcal{R}_2$
4:    **for** $i = 1, 2$ **do**
5:       *reduce the domain* of $\mathcal{R}_i$ using H and $\overline{G}$
6:       suppose $\mathcal{R}_i = [a, b]$
7:       **if** $f(a) = +\infty$ or $b \notin H$ **then**
8:          in case of $f(a) = +\infty$, *add a feasibility cut* to refine $\overline{G}$
9:          *fathom* $\mathcal{R}_i$ (it contains no feasible solutions)
10:       **else** $\{f(a) < +\infty$ and $b \in H\}$
11:          set $LB(\mathcal{R}_i) = f(a)$
12:          **if** $a \in H$ **then**
13:             a is a feasible solution, *add an optimality cut* to refine $\overline{G}$
14:             **if** $f(a) < UB$ **then**
15:                $UB \leftarrow f(a)$ and $y^* \leftarrow a$
16:             **end if**
17:             *fathom* $\mathcal{R}_i$ (a is the best feasible solution in $\mathcal{R}_i$)
18:          **else** $\{a \notin H\}$
19:             set $\mathcal{L} \leftarrow \mathcal{L} \cup \{\mathcal{R}_i\}$
20:             *search for feasible solutions:* select a finite set $\mathcal{S} \subset \mathcal{R}_i$
21:             **for** each $y \in \mathcal{S}$ **do**
22:                **if** $f(y) = +\infty$ **then**
23:                   *add a feasibility cut* to refine $\overline{G}$
24:                **else if** $f(y) < +\infty$ and $y \in H$ **then**
25:                   y is a feasible solution, *add an optimality cut* to refine $\overline{G}$
26:                   **if** $f(y) < UB$ **then**
27:                      $UB \leftarrow f(y)$ and $y^* \leftarrow y$
28:                   **end if**
29:                **end if**
30:             **end for**
31:          **end if**
32:       **end if**
33:    **end for**
34:    **for** each $\mathcal{R} \in \mathcal{L}$ **do**
35:       **if** $LB(\mathcal{R}) > UB$ or $v(\mathcal{R}) \leq \epsilon$ **then**
36:          *fathom* $\mathcal{R}$, i.e, set $\mathcal{L} \leftarrow \mathcal{L} \setminus \{\mathcal{R}\}$
37:       **end if**
38:    **end for**
39: **end while**

Figure 2: A Branch-Reduce-Cut algorithm for PCLP

then σ_p and σ_q always exist and the above branching rule is well-defined. Moreover, the resulting partitions $\mathcal{R}_1$ and $\mathcal{R}_2$ also satisfy condition (11). We shall show that the domain reduction step on any partition preserves condition (11). Thus, if the initial partition $[y^L, y^U]$ satisfies condition (11) (this can be ensured by domain reduction), then all subsequent partitions produced by the algorithm will satisfy (11).

3.2 Domain reduction

Suppose we have a current normal set approximation $\overline{G}$ of the set of optimal solutions, then the region of interest is $\overline{G} \cap H$. Given a partition $\mathcal{R} = [a, b]$ such that $a \in \overline{G}$ and $b \in H$, the idea of domain reduction is to find a hyper-rectangle $\mathcal{R}' \subset \mathcal{R}$ such that $\mathcal{R}' \supseteq \mathcal{R} \cap \overline{G} \cap H$. This reduction helps to obtain improved bounds and reduces the size of the search tree.

The domain reduction on $\mathcal{R} = [a, b]$, with $a \in \overline{G}$ and $b \in H$, can be carried out as follows. Let

$$\lambda_j^* = \max\{\lambda \in [0,1]: \ a + \lambda(b_j - a_j)e^j \in \overline{G}\} \qquad \text{for } j = 1, \ldots, m \tag{12}$$

$$\mu_j^* = \max\{\mu \in [0,1]: \ b - \mu(b_j - a_j)e^j \in H\} \qquad \text{for } j = 1, \ldots, m. \tag{13}$$

The above problems both have optimal solutions, since the sets $\overline{G}$ and H are closed. Construct

$$a' = b - \sum_{j=1}^m \mu_j^*(b_j - a_j)e^j \tag{14}$$

$$b' = a + \sum_{j=1}^m \lambda_j^*(b_j - a_j)e^j. \tag{15}$$

Proposition 6. $[a', b'] \supseteq [a, b] \cap \overline{G} \cap H$.

Proof. Consider $y \in [a, b] \cap \overline{G} \cap H$, such that $y \notin [a', b']$. Then there exists an index $\widehat{\jmath} \in \{1, \ldots, m\}$, such that either $a_{\widehat{\jmath}} \leq y_{\widehat{\jmath}} < a'_{\widehat{\jmath}}$ or $b'_{\widehat{\jmath}} < y_{\widehat{\jmath}} \leq b_{\widehat{\jmath}}$. Consider the first case (the second case is analogous). Let $z = y + \sum_{j=1, j \neq \widehat{\jmath}}^m (b_j - y_j)e^j$, i.e., $z_j = b_j$ for all $j = 1, \ldots, m, j \neq \widehat{\jmath}$ and $z_{\widehat{\jmath}} = y_{\widehat{\jmath}}$. Let $\widehat{\mu}_{\widehat{\jmath}} = (b_{\widehat{\jmath}} - z_{\widehat{\jmath}})/(b_{\widehat{\jmath}} - a_{\widehat{\jmath}})$. Clearly $\widehat{\mu} \in [0,1]$, and we can write $z = b - \widehat{\mu}_{\widehat{\jmath}}(b_{\widehat{\jmath}} - a_{\widehat{\jmath}})e^{\widehat{\jmath}}$. Note that, by construction, $z \geq y$, and since $y \in H$, we have that $z \in H$ (since H is a reverse normal set). Thus $\widehat{\mu}_{\widehat{\jmath}}$ is a feasible solution to the problem (13) corresponding to $\widehat{\jmath}$. Then

$$b_{\widehat{\jmath}} - \widehat{\mu}_{\widehat{\jmath}}(b_{\widehat{\jmath}} - a_{\widehat{\jmath}}) = z_{\widehat{\jmath}} = y_{\widehat{\jmath}} < a'_{\widehat{\jmath}} = b_{\widehat{\jmath}} - \mu_{\widehat{\jmath}}^*(b_{\widehat{\jmath}} - a_{\widehat{\jmath}}).$$

The above implies that $\widehat{\mu}_{\widehat{\jmath}} > \mu_{\widehat{\jmath}}^*$ and we have a contradiction to the fact that $\mu_{\widehat{\jmath}}^*$ is an optimal solution to the problem (13) corresponding to $\widehat{\jmath}$. $\qquad\square$

Proposition 6 establishes that the domain reduction process is valid, i.e., no feasible solutions are lost. In case of continuous distributions, the one-dimensional optimization problems (12) and (13) can be solved by bisection. For discrete distributions, the problems simplify to

$$\lambda_j^* = \max\{\lambda \in [0,1]: \ a + \lambda(b_j - a_j)e^j \in \overline{G} \cap \mathcal{C}\} \qquad \text{for } j = 1, \ldots, m$$

$$\mu_j^* = \max\{\mu \in [0,1]: \ b - \mu(b_j - a_j)e^j \in H \cap \mathcal{C}\} \qquad \text{for } j = 1, \ldots, m,$$

and can be solved by sorting the elements in Ξ_j for all $j = 1, \ldots, m$. It is immediately seen that, in this case, the resulting partition $[a', b']$ will satisfy condition (11).

3.3 Feasibility and optimality cuts

Whenever a solution $\widehat{y}$ is found such that $f(\widehat{y}) = +\infty$, i.e., the linear program (7) is infeasible, a feasibility cut is added to the description of the set $\overline{G}$ (see lines 8 and 22 of the main loop in Figure 2). The cut is generated as follows. Recall that dual polyhedron for the linear program (7) is

$$\{(\pi, \rho) \in \mathbb{R}^{p+m}: \ \pi A + \rho T \leq c, \ \rho \geq 0\}.$$

If $f(\widehat{y}) = +\infty$, then the dual to (7) is unbounded (since dual feasibility is assumed), i.e., there exists an extreme ray $(\widehat{\pi}, \widehat{\rho})$ of the above dual polyhedron such that

$$\widehat{\pi}b + \widehat{\rho}\widehat{y} > 0.$$

Thus any $y \in G$ should satisfy the *feasibility cut*

$$\widehat{\rho}y \leq -\widehat{\pi}b. \tag{16}$$

Whenever a solution $\widehat{y} \in H$ such that $f(\widehat{y}) < +\infty$, i.e., $\widehat{y}$ is feasible, an optimality cut is added to the description of $\overline{G}$ (see lines 12 and 24 of the main loop in Figure 2). The optimality cut is generated as follows. Let $(\widehat{\pi}, \widehat{\rho})$ be an optimal dual solution for the linear program (7) defining $f(\widehat{y})$. Then any y such that $f(y) \leq f(\widehat{y})$ should satisfy the *optimality cut*

$$\widehat{\rho}y \leq \widehat{\rho}\widehat{y}. \tag{17}$$

To see this, first note that $(\widehat{\pi}, \widehat{\rho})$ is a feasible dual solution for the linear program (7) defining $f(y)$, thus $f(y) \geq \widehat{\pi}b + \widehat{\rho}y$. Since $f(\widehat{y}) = \widehat{\pi}b + \widehat{\rho}\widehat{y}$, the requirement $f(y) \leq f(\widehat{y})$ then implies the optimality cut (17).

At any point in the algorithm, the set $\overline{G}$ is defined by a set of feasibility and optimality cuts of the forms (16) and (17), respectively. Note that the cut-coefficients are always non-negative. Thus the set $\overline{G}$ is always maintained to be a normal set.

Example 2 (contd.): Here, we illustrate the cutting plane and domain reduction steps using Example 2. Figure 3(a) shows a feasibility cut. Consider the initial partition defined by $[(-5, 1.5), (1, 5.5)]$ (shown as the dashed rectangle). While searching for feasible solutions (see the subsection 3.4), the algorithm considers the $(1, 5.5)$ and its two adjacent vertices $(-5, 1.5)$ and $(1, 1.5)$. All of these points are infeasible. The sloped dotted line in Figure 3(a) is the feasibility cut corresponding to the point $(1, 1.5)$. Upon adding this cut, the domain reduction step is able to reduce the partition $[(-5, 1.5), (1, 5.5)]$ to $[(-5, 2), (-2, 5)]$ (shown as the solid rectangle).

Figure 3(b) shows two optimality cuts. Consider now the partition $[(-5, 2), (2, 5)]$. The vertices $(-2, 2)$ and $(-5, 5)$ both have finite optimal objective values, and consequently the two optimality cuts shown as sloped dotted lines in Figure 3(b) are generated. The optimality cut corresponding to $(-5, 5)$ helps to reduce the partition $[(-5, 2), (-2, 5)]$ to $[(-5, 2), (-3, 5)]$.
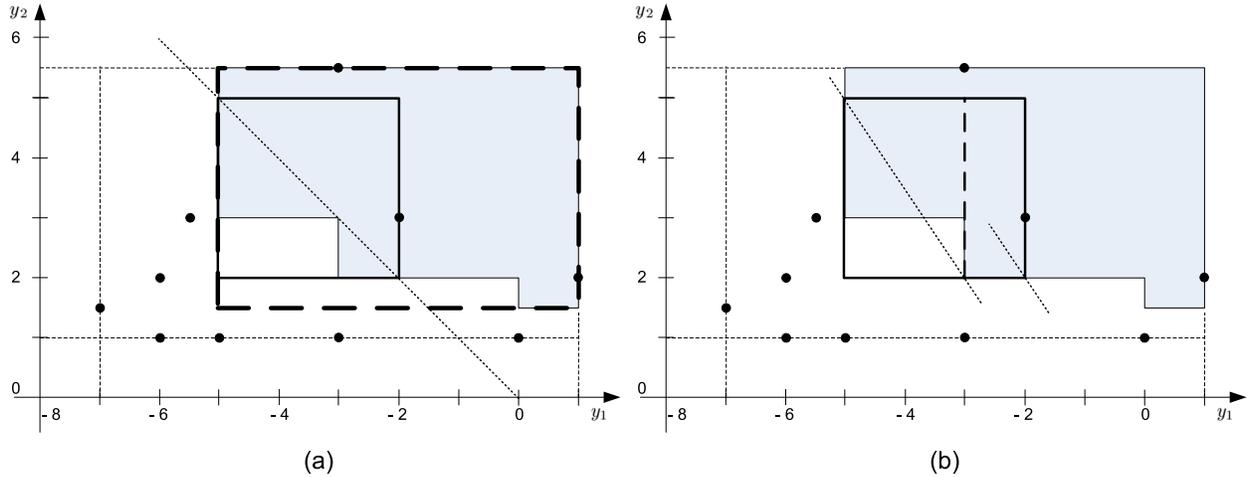

Figure 3: (a) Feasibility cut corresponding to $y = (1, 1.5)^T$ and (b) Optimality cuts corresponding to $y = (-5, 5)^T$ and $y = (-2, 2)^T$

3.4 Upper bounding and searching for feasible solutions

Whenever a feasible solution is found, the corresponding objective value is checked against the upper bound UB, i.e., the objective value of the best solution y^* found thus far, and UB and y^* are updated appropriately. Also, as mentioned earlier, an optimality cut is added to the description of $\overline{G}$.

For a given partition $\mathcal{R}$, the algorithm first checks if a is feasible, i.e., $f(a) < +\infty$ and $a \in H$. If it is, then this partition is fathomed, since it cannot contain any better feasible solution.

To search for feasible solutions within an unfathomed partition $\mathcal{R} = [a, b]$, the algorithm (see lines 20-30 of the main loop in Figure 2) checks b and the set of m vertices of $\mathcal{R}$ adjacent to b (this is the candidate set $\mathcal{S}$). The adjacent vertices are checked since these might have smaller objective value than b. We might also search for solution on the relative boundary of H on the line segment joining a to b, however, this might be computationally expensive.

Suppose we encounter a solution $\widehat{y}$ such that $f(\widehat{y}) < +\infty$ but $\widehat{y} \notin H$. We may then attempt to "lift" this solution to be feasible as follows. Let $\widehat{x}$ be an optimal solution to the linear program (7) defining $f(\widehat{y})$. Then, $T\widehat{x} \geq \widehat{y}$. If some of these inequalities are strict, we can consider the solution $y' = \min\{T\widehat{x}, b\}$ (the min is taken component-wise). Clearly, $f(y') = f(\widehat{y})$, and since $y' \geq \widehat{y}$, it might be that $y' \in H$, in which case, y' is a feasible solution.

3.5 Fathoming

A given partition $\mathcal{R} = [a, b]$ can be fathomed, or discarded from further consideration, if one of the following conditions are satisfied.

Infeasibility: If the partition fails to satisfy the necessary feasibility condition $a \in G$ and $b \in H$, then it can fathomed since contains no feasible solutions.

Feasibility: If a or a solution obtained by "lifting" a is feasible, then the partition can be fathomed since it contains no better solutions.

Inferiority: If $LB(\mathcal{R}) > UB$, the partition does not contain any solution whose objective function value is smaller than that of the best solution already found. Hence the partition can be fathomed.

Tolerance: If the $v(\mathcal{R}) \leq \epsilon$ and has not been fathomed according to the rules above, then the partition is not considered further. In this case, we have $f(a) < +\infty$, $a \notin H$ and $b \in H$. In this case, we choose a to be an "approximately feasible" solution and compare it with the best candidate solution found so far. Note that, if $\epsilon = 0$ (as in the case of discrete distributions) this fathoming rule is never encountered.

4 Convergence analysis

4.1 Discrete distribution

We shall show that, in case of a discrete distribution of the random vector ξ, the proposed branch-reduce-cut algorithm (with the tolerance $\epsilon = 0$), either finds a global optimal solution to PCLP or resolves that the problem is infeasible. We shall need the following concept.

Definition 1. [8] *The bounding operation in a branch and bound algorithm is called* finitely consistent *if, (i) at every step, any unfathomed partition element can be further refined, and if (ii) any nested sequence of successively refined partition elements is finite.*

Lemma 2. *The bounding operation in the Branch-reduce-cut algorithm is finitely consistent.*

Proof. Recall that the algorithm always generates partitions satisfying condition (11). Consider, any unfathomed partition $[a, b]$ satisfying (11). As mentioned earlier, since $\{a, b\} \in \mathcal{C}$, the branching rule is well-defined, and so $[a, b]$ can be further refined. Also, the branching rule guarantees that any nested sequence of unfathomed partitions will reduce to a point after a finite number of steps, whence such a partition will have to be fathomed. Thus any nested sequence of successively refined partition elements is finite. $\square$

Theorem 1. *Given a PCLP with discrete distributions for the random parameters, the Branch-reduce-cut algorithm with $\epsilon = 0$ terminates after finitely many steps either with a global optimal solution or by resolving that the problem is infeasible.*

Proof. From Theorem IV.1 in [8], a branch-and-bound algorithm where the bounding operation is finitely consistent terminates in a finite number of steps. Thus by Lemma 2, the proposed Branch-reduce-cut algorithm terminates after finitely many steps.

Consider, first, the case that the algorithm found a feasible solution y^*. In this case, the feasible region is non-empty, therefore PCLP has an optimal solution. Suppose $y' \in \mathcal{C}$ is an optimal solution to PCLP. We shall show that $f(y^*) = f(y')$, so that y^* is an optimal solution of PCLP. Suppose, for contradiction, that $f(y') < f(y^*)$. Let $\mathcal{R}_i = [a^i, b^i]$ for $i = 1, \ldots, I$ be the set of partitions corresponding to the leaf nodes of the branch-and-bound search tree upon termination. By the exhaustiveness of the branching rule and the validity of the domain reduction rules, there exists i' such that $y' \in \mathcal{R}_{i'} = [a^{i'}, b^{i'}]$. Then $a^{i'} \in G$ and $b^{i'} \in H$, therefore $\mathcal{R}_{i'}$ cannot be fathomed due to infeasibility. Moreover, by $f(a^{i'}) \leq f(y') \leq f(b^{i'})$, we must have $a^{i'} \notin H$, since otherwise $a^{i'}$ would be an optimal solution, and the algorithm would have discovered it and set $y^* = a^{i'}$. So $\mathcal{R}_{i'}$ cannot be fathomed by feasibility. Finally, since $f(a^{i'}) \leq f(y') < f(y^*) = UB$, therefore $\mathcal{R}_{i'}$ cannot be fathomed by inferiority. Thus, $\mathcal{R}_{i'}$ can be further refined, and the algorithm should not have terminated. Hence $f(y') = f(y^*)$.

Consider now the second case, that the algorithm did not find any feasible solution. We then claim that PCLP is infeasible. Suppose, for contradiction, that y' is a feasible solution to PCLP. As before, let $\mathcal{R}_i = [a^i, b^i]$ for $i = 1, \ldots, I$ be the set of partitions corresponding to the leaf nodes of the branch-and-bound search tree upon termination, and let i' be such that $y' \in \mathcal{R}_{i'} = [a^{i'}, b^{i'}]$. Then $a^{i'} \in G$ and $b^{i'} \in H$, therefore $\mathcal{R}_{i'}$ cannot be fathomed due to infeasibility. Moreover, $a^{i'} \notin H$, since otherwise $a^{i'}$ would be a feasible solution, and the algorithm would have discovered it. So $\mathcal{R}_{i'}$ cannot be fathomed by feasibility. Finally, since no feasible solution has been found, $UB = +\infty$, and $f(a^{i'}) < UB$, therefore $\mathcal{R}_{i'}$ cannot be fathomed by inferiority. Thus, $\mathcal{R}_{i'}$ can be further refined, and the algorithm should not have terminated. Hence, PCLP cannot have any feasible solution. $\square$

4.2 Continuous distribution

In the continuous distribution case, if $\epsilon = 0$, then the algorithm may not terminate finitely and can only be guaranteed to converge in the limit. To see this, suppose that the feasible set is a singleton, then an infinite number of branching operations may be needed to attain a partition $[a, b]$ where the single feasible point coincides with a.

Theorem 2. *Given a PCLP with continuous distributions for the random parameters, if the Branch-reduce-cut algorithm with $\epsilon = 0$ terminates, then it terminates with a global optimal solution or by resolving that the problem is infeasible.*

Proof. Analogous to the proof of Theorem 1. $\square$

Consider now the case when the algorithm does not terminate. Let k denote the index for the iterations, UB^k denote the upper bound at iteration k, and $[a^k, b^k]$ be the partition considered in iteration k.

Theorem 3. *Given a PCLP with continuous distributions for the random parameters, if the Branch-reduce-cut algorithm with $\epsilon = 0$ does not terminate, then PCLP has an optimal solution y^*, and*

$$\lim_{q \to \infty} a^{k_q} = y^*.$$

Proof. Note that $a^k \in G$ and $b^k \in H$ for all k. Also by the exhaustiveness of the branching rule, we have $\lim_{k \to \infty}(b^k - a^k) = 0$. Since the sets G and H are closed, for any convergent subsequence $\lim_{q \to \infty} b^{k_q} = \lim_{q \to \infty} a^{k_q} = a^* \in G \cap H \cap [y^L, y^U]$. Thus PCLP has a feasible solution a^*, and hence an optimal solution y^*.

The least lower bound selection rule guarantees that $f(a^{k_q}) \leq f(y^*)$ for all q. Thus $\lim_{q \to \infty} f(a^{k_q}) = f(a^*) \leq f(y^*)$, where the first equality follows from the continuity of f over G. Therefore, a^* is also an optimal solution. $\square$

To ensure finite termination, a positive tolerance ($\epsilon > 0$) is required. The tolerance-based fathoming rule then guarantees that the algorithm terminates. In this case, we might end up with a δ-feasible solution y^*, i.e., $\alpha - F(y^*) \leq \delta$ for some $\delta > 0$. For a PCLP with an absolutely continuous cumulative density $F : \mathbb{R}^m \mapsto [0,1]$ for the random vector $\xi(\omega)$ with $f_i : \mathbb{R}^m \mapsto \mathbb{R}_+$ for $i = 1, \ldots, m$ as the corresponding marginal probability density functions, ϵ and δ are related as $\delta \leq \epsilon L$, where $L = \max_{i=1,\ldots,m} \max_{y \in [y^L, y^U]} \{f_i(y)\}$.

5 Computational results

In this section we report on some computational experience in using the proposed Branch-cut-reduce algorithm for randomly generated instances of PCLP with discrete distributions, i.e., $\xi(\omega)$ has K possible realizations $\{\xi^1, \ldots, \xi^K\}$ with probabilities $\{p^1, \ldots, p^K\}$. Such a problem can be immediately reformulated into the following milp (see, e.g., [15])

$$\min_{x,y,\lambda} \quad c^T x$$
$$\text{s.t.} \quad Ax = b$$
$$Tx \geq y$$
$$\sum_{k=1}^{K} p^k \lambda_k \geq \alpha$$
$$\xi_j^k \lambda_i \leq y_j \quad \text{for } j = 1, \ldots, m, \ k = 1, \ldots, K$$
$$\lambda_k \in \{0,1\} \quad \text{for } k = 1, \ldots, K$$
$$x \geq 0, y \geq 0.$$

The above milp formulation can be improved by adding the constraints

$$\lambda_{k_1} \geq \lambda_{k_2} \quad \text{if } \xi^{k_1} \leq \xi^{k_2} \text{ for } k_1 = 1, \ldots, K \text{ and } k_2 = k_1 + 1, \ldots, K.$$

We compare the performance of the proposed algorithm with that of solving the above milp formulation using CPLEX 8.0.

The proposed algorithm was implemented in C++ with CPLEX 8.0 as the linear programming solver. All computations were on a UltraSparc-III-Cu UNIX workstation 2x900MHz CPUs and 2GB RAM.

For generating our test problems, we replaced the constraint $Ax = b$ by simple bounds. The number n of x-variables was fixed at $n = 50$, and the number m of y-variables was varied in the set $\{3, 6, 9\}$. The number of scenarios K was varied in the set $\{100, 300, 500\}$. The desired probability level α was set to 0.9. For each combination of m and K, five test problems were randomly generated as follows:

1. K realizations of the ξ vector were randomly sampled from a uniform distribution over $[0, 100]^m$. Each realization was assigned a probability of $1/K$.

2. Each component of c and T were independently randomly sampled from a uniform distribution over $[0, 20]$.

Table 2 compares, for each test problem, the CPU seconds required to solve the milp reformulation using CPLEX 8.0 (under columns labelled "MIP"), the CPU seconds required by a branch-and-reduce algorithm, i.e., the proposed scheme without the enhancements offered by cutting planes (under the columns labelled "BR"), and the CPU seconds required by the proposed branch-reduce-cut algorithm (under the columns labelled "BRC"). In the majority of the test problems, the proposed branch-cut-reduce algorithm performed better that the other two approaches. In particular, in only two out of 45 test problems, the milp approach showed better performance.

In Figure 4, we compare the growth of the CPU time required by the milp approach and that required by the proposed algorithm, with and without the cutting plane enhancements, as the number K of realizations in the distribution of the uncertain parameters grows for PCLPs with $m = 5$ and $n = 50$. Each data point

	$K = 100$			$K = 300$			$K = 500$		
Test No.	MIP	BR	BRC	MIP	BR	BRC	MIP	BR	BRC
1	0.04	0.00	0.00	2.78	0.00	0.00	240.84	0.19	0.04
2	0.04	9.00	1.59	7.67	0.28	0.08	16.26	0.00	0.00
3	0.03	0.00	0.00	42.77	0.25	0.00	175.48	0.59	0.12
4	0.09	0.06	0.00	2.81	0.00	0.00	16.36	0.00	0.00
5	0.04	0.00	0.00	5.35	0.00	0.00	18.71	0.17	0.05
AVG	0.05	2.27	0.40	14.65	0.13	0.02	56.70	0.19	0.04

$$m = 3$$

	$K = 100$			$K = 300$			$K = 500$		
Test No.	MIP	BR	BRC	MIP	BR	BRC	MIP	BR	BRC
1	0.03	0.00	0.00	10.46	0.32	0.00	7.91	0.00	0.00
2	0.14	0.15	0.00	23.62	0.77	0.13	8.20	0.00	0.00
3	0.05	0.17	0.00	51.43	3.16	0.39	59.11	6.50	4.01
4	0.05	0.37	0.00	2.12	0.62	0.22	4.94	0.00	0.00
5	0.24	0.09	0.00	1.70	0.00	0.00	202.57	3.47	1.51
AVG	0.10	0.16	0.00	17.87	0.97	0.15	56.55	1.99	1.10

$$m = 6$$

	$K = 100$			$K = 300$			$K = 500$		
Test No.	MIP	BR	BRC	MIP	BR	BRC	MIP	BR	BRC
1	2.21	1.04	0.00	1.91	0.00	0.00	445.89	9.61	5.09
2	0.65	0.46	0.00	82.90	3.18	0.63	307.91	3.94	0.16
3	5.34	29.13	8.61	3.94	0.32	0.00	2261.06	37.55	13.64
4	5.39	0.72	0.08	5.00	0.53	0.00	89.97	1.45	0.38
5	3.25	3.05	0.62	2.73	1.38	0.64	5216.37	21.54	16.75
AVG	3.37	6.88	1.86	19.30	1.08	0.25	1664.24	14.82	7.20

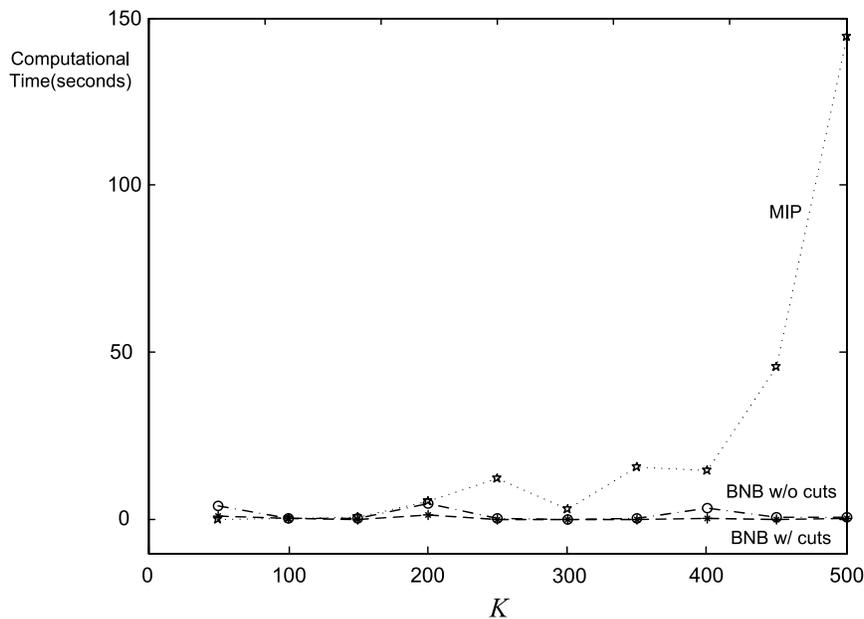$$m = 9$$

Table 2: Comparison of CPU seconds

Figure 4: CPU seconds versus K ($m = 5$, $n = 50$).

in the graph corresponds to an average over 5 randomly generated instances. It is evident that the proposed algorithm offers significant advantages over the milp approach as the the number of realizations increase.

References

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

[2] P. Beraldi and A. Ruszczyński. The probabilistic set covering problem. *Operations Research*, pages 956–967, 2002.

[3] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, Massachusetts, 1997.

[4] A. Charnes and W. W. Cooper. Chance-constrained programming. *Management Science*, 6:73–89, 1959.

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

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

[7] R. Horst, P. M. Pardalos, and N. V. Thoai. *Introduction to Global Optimization*. Kluwer Academic Publishers, Dordrecht, 2001.

[8] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer-Verlag, Berlin, Germany, 1996.

[9] A.H. Land and A.G. Doig. An automatic method for solving discrete programming problems. *Econometrica*, pages 497–520, 1960.

[10] D. Li, X. L. Sun, M. P. Biswal, and F. Gao. Convexification, concavification and monotonization in global optimization. *Annals of Operations Research*, 105:213–226, 2001.

[11] LINDO Systems Inc. LINDO API 2.0 and LINGO 8.0. http://www.lindo.com/.

[12] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience, New York, USA, 1999.

[13] A. Prékopa. Contributions to the theory of stochastic programming. *Mathematical Programming*, 4:202–221, 1973.

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

[15] A. Prékopa. *Stochastic Programming*. Kluwer Academic Publishers, Dordrecht, Ther Netherlands, 1995.

[16] A. Ruszczyński. Probabilistic programming with discrete distributions and precedence constrained knapsack polyhedra. *Mathematical Programming*, Ser. A93:195–215, 2002.

[17] N. V. Sahinidis. Optimization under uncertainty: state-of-art and opportunities. *Computers & Chemical Engineering*, 28:971–983, 2004.

[18] N.V. Sahinidis. BARON: A global optimization software. http://archimedes.scs.uiuc.edu/baron/baron.html.

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

[20] S. Sen and J. L. Higle. An introductory tutorial on stochastic linear programming models. *Interfaces*, 29(2):31–61, 1999.

[21] M. Tawarmalani and N. V. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*. Kluwer Academic Publishers, Dordrecht, 2002.

[22] K.-A. Toh. Global optimization by monotonic transformation. *Computational Optimization and Applications*, 23:77–99, 2002.

[23] H. Tuy. Monotonic optimization: Problems and solution approaches. *SIAM Journal of Optimization*, 11(2):464–494, 2000.