Fatima BELLAHCENE[1]

APPLICATION OF THE POLYBLOCK METHOD
TO SPECIAL INTEGER CHANCE CONSTRAINED PROBLEM

The focus in this paper is on a special integer stochastic program with a chance constraint in which, with a given probability, a sum of independent and normally distributed random variables is bounded below. The objective is to maximize the expectation of a linear function of the random variables. The stochastic program is first reduced to an equivalent deterministic integer nonlinear program with monotonic objective and constraints functions. The resulting deterministic problem is solved using the discrete polyblock method which exploits its special structure. A numerical example is included for illustration and comparisons with LINGO, COUENNE, BONMIN and BARON solvers are performed.

Keywords: *stochastic programming, integer nonlinear programming, monotone optimization, polyblock method*

1. Introduction

We consider a stochastic program with integer decision variables x_j, $j = 1, ..., n$ where, the expectation of a linear function of the random variables $Y_j(x)$ is maximized under a chance constraint in which, with a predefined probability level α, is $\sum_{j=1}^{n} Y_j(x)$ bounded above by a positive constant b. The problem is formulated as

---

[1]LAROMAD Laboratory, Faculty of Sciences, Mouloud Mammeri University, BP 17 RP, 15000 Tizi-Ouzou, Algeria, e-mail address: belfat.ummto@gmail.com

$$\max E\left(\sum_{j=1}^{n} \beta_j Y_j(x)\right)$$

subject to

$$\Pr\left(\sum_{j=1}^{n} Y_j(x) \leq b\right) \geq \alpha \tag{1}$$

$$x \in X = \left\{x \in \mathbb{Z}_+^n \mid l \leq x \leq d\right\}$$

where $l = (l_1, l_2, ..., l_n)$ and $d = (d_1, d_2, ..., d_n)$ are vectors of integer numbers with $l_j \leq d_j$ for $j = 1, ..., n$ and β_j, $j = 1, ..., n$ are positive numbers.

Such a decision problem may arise in higher education applications when an institution wishes to maximize the quality or the desirability of a class of students while satisfying a constraint on the number of students who enrol due to its limited capacity. Students are also categorized into different types, $j = 1, ..., n$. The decision variables x_j represent the number of applicants of the type j who are admitted and the random variables $Y_j(x)$ represent the number of students of type j who enrol. It is assumed that the institution imposes an upper bound b on enrolment that must be met with a high degree of probability α. If an admitted student of type j enrols with a probability p_j independent of other students, then each random variable $Y_j(x)$ has a binomial distribution with parameters p_j, x_j and therefore has mean $\mu_j = p_j x_j$ and variance $\sigma_j^2 = p_j(1 - p_j) x_j = v_j x_j$.

Similar problems may arise in other applications such as airline overbooking problems, where x_j is the number of reservations accepted and $Y_j(x)$ is the number of passengers or guests who show up; marketing, where x_j is the number of solicitations issued and $Y_j(x)$ is the number of responses and manufacturing, where x_j is the number of items produced and $Y_j(x)$ is the number of non-defective items.

In each of these applications, the number of items is generally large. Then, the binomial random variables are easily approximated by normal random variables. Of course, the lower bounds l_j of the variables x_j should be sufficiently large to allow this approximation because we usually approximate a binomial random variable $B(p_j, x_j)$ with a normal distribution from the time where $p_j x_j > 15$ which yields that $x_j > 30$.

Under the assumption that the item types act independently, $\sum_{j=1}^{n} Y_j(x)$ is approximately normal with mean μ and variance σ^2, where

$$\mu = \sum_{j=1}^{n} \mu_j = \sum_{j=1}^{n} p_j x_j \quad \text{and} \quad \sigma^2 = \sum_{j=1}^{n} \sigma_j^2 = \sum_{j=1}^{n} v_j x_j \tag{2}$$

The objective function can be written as

$$E\left(\sum_{j=1}^{n} \beta_j Y_j(x) \right) = \sum_{j=1}^{n} \beta_j p_j x_j = \sum_{j=1}^{n} c_j x_j \tag{3}$$

Furthermore, as introduced by Charnes and Cooper [5], the lower-bound chance constraint in (1) may be written as

$$\Pr\left(\sum_{j=1}^{n} Y_j(x) \le b \right) \ge \alpha \Leftrightarrow \Phi\left(\frac{b-\mu}{\sigma} \right) \ge \alpha \Leftrightarrow -b + \mu + \sigma K_\alpha \le 0$$

where Φ is the cumulative function of the standard normal distribution and K_α is its α-quantile. We assume that $\alpha \ge 1/2$, which is justified by the fact that we are interested in the politics which guarantees the achievement of the constraints with high probability.

Using this approximation and taking into account the expressions (2) and (3), the stochastic program (1) becomes

$$\max \sum_{j=1}^{n} c_j x_j$$

subject to

$$g_0(x) = -b + \sum_{j=1}^{n} p_j x_j + K_\alpha \sqrt{\sum_{j=1}^{n} v_j x_j} \le 0 \tag{4}$$

$$x \in X = \left\{ x \in \mathbb{Z}_+^n \mid l_j \le x_j \le d_j, \ j = 1, ..., n \right\}$$

As a result, g_0 is a nonconvex and increasing function of x_j on $\left[l_j, d_j \right]$.

Note that problem (1) may not be necessarily restricted to a single-row chance constraint. Other technical constraints $g_i(x) \le 0$, $i = 1, ..., m$ with linear or nonlinear increasing functions g_i's can be added to the constraint set without changing anything in the algorithm that we present in Section 2. In sum, we are dealing with the following nonlinear integer monotonic program:

$$\max f(x)$$
$$\text{subject to}$$
$$g_i(x) \le 0, \quad i = 0, ..., m \tag{5}$$
$$x \in X = \left\{ x \in \mathbb{Z}_+^n \mid l_j \le x_j \le d_j, \ j = 1, ..., n \right\}$$

where both f and g_i are increasing functions.

For monotonic programming problems, Tuy [20] develops an outer approximation algorithm called the polyblock algorithm. Li et al. [15] propose pth power convexification and concavification schemes that transform a monotone function into either a concave or a convex function. According to Tuy [20], monotonic programming approaches have been demonstrated to be efficient for solving multiplicative programming problems, while alternative methods hardly handle them. The efficiency of the monotonic programming approaches has been reported with computational results on various classes of global optimization problems, such as linear or polynomial fractional programming [10, 21], and discrete nonlinear programming [22].

Mixed-integer nonlinear programming (MINLP) problems have been studied in the literature. In the early 1970s, Geoffrion [9] introduces the generalized benders decomposition (GBD) method. The GBD method uses duality theory to derive one single constraint that combines the linearizations derived from all the original problem constraints and solves a mixed-integer linear programming (MILP) master problem. In the 1980s, Duran and Grossmann [7] introduce the outer approximation (OA) decomposition algorithm. The OA method is very similar to the GBD method, differing only in the definition of the MILP master problem. Specifically, instead of combining the linearizations derived from all the original problem constraints, it uses linearizations for each nonlinear constraint. The fundamental insight behind the algorithm is that the MINLP problem is equivalent to a MILP problem of finite size. This latter algorithm was subsequently improved in the 1990s by Fletcher and Leyfer [8]. Quesada and Grossmann propose the LP/NLP-Based Branch-and-Bound (LP/NLP–BB) algorithm [17]. This method is an extension of the OA method but instead of solving a sequence of master problems, the master problem is dynamically updated in a single branch-and-bound tree that closely resembles the branch-and-cut method for MILP. In the same period, a related method called the Extended Cutting Plane (ECP) method, which is an extension of Kelley's cutting plane method [11] for solving convex NLPs, was proposed by Westerlund and Pettersson [24]. The main feature of the ECP method is that it does not require the use of an NLP solver. The algorithm is based on the iterative solution of a reduced master problem (RMP). Linearizations of the most violated constraint at the optimal solution of RMP are added at every iteration. Indeed, there are many modern software packages implementing the cited algorithms (For more details, see, for example, Abhishek et al. [1], Bonami et al. [3], Bonami et al. [4]).

Since our problem is nonconvex, it cannot be solved by the MINLP methods mentioned above. Therefore, solvers such as SCIP [23], COUENNE [6], BARON [12], ANTIGONE [16], BONMIN [6], LindoGlobal [13, 14] must be used. However, the special structure of the studied problem encouraged us to design the discrete polyblock method [18] to solve it. This method applies only to monotonic problems where all the decision variables are integers, which is the main concern of this paper. Moreover, it does not require linearization or convexification or any other properties except monotonicity of the functions. At each stage of the technique, the feasible domain is divided into two sub-domains and each of them is analyzed in order to discard the one not containing promising solutions. The results found by the polyblock algorithm are compared with those given by LINGO, COUENNE, BONMIN, and BARON in terms of running time and number of iterations.

1.1. The discrete polyblock method

Let us first review some properties of monotonic functions from the general results in [19, 20, 22].

For any two vectors $x, y \in \mathbb{R}^n$ we write $x \le y$ to mean $x_j \le y_j$ for $j = 1, ..., n$. If $l \le d$ then the box $[l, d]$ is the set of all $x \in \mathbb{R}^n$ satisfying $l \le x \le d$. The function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be increasing on $\mathbb{R}^n_+$ if $f(x) \le f(y)$ whenever $x \le y$. Many functions encountered in various applications are increasing in this sense. Outstanding examples are the production functions and the utility functions in mathematical economics (under the assumption that all goods are useful). The sum of two increasing functions is increasing and a product λf is increasing if $\lambda > 0$ and f is increasing. Consequently, polynomials with non-negative coefficients and posynomials $\sum_{j=1}^{m} c_j \prod_{i=1}^{n} (x_i)^{a_{ij}}$ with $c_j \ge 0$ and $a_{ij} \ge 0$, such as the well-known Gobb–Douglas function $f(x) = \prod_{i=1}^{n} x_i^{a_i}$, $a_i \ge 0$ are increasing.

Letting

$$G(x) = \max_{i=0,...,m} \{ g_i(x) \} \tag{7}$$

the boundary of the feasible set $S = \{ x \in X \mid g_i(x) \le 0, \quad i = 0, ..., m \}$ of problem (5) can be expressed as $\Gamma = \{ x \in X \mid G(x) = 0 \}$.

Let $[\alpha, \beta]$ be an integer box in X with $\alpha \in S$ and $\beta \notin S$. Since $G(\alpha) < 0$ and $G(\beta) > 0$, there must exist a boundary point x_b in X that satisfies $G(x_b) = 0$ (i.e., $g_i(x_b) \leq 0$, $0 = 1, ..., m$ and there exists at least one i such that $g_i(x_b) = 0$). x_b is the intersection point of the line $x = \lambda^* \alpha + (1 - \lambda^*) \beta$, $0 \leq \lambda^* \leq 1$ and the boundary Γ.

Assume now that x_b is not integral. Denote by $\lfloor x_b \rfloor$ the integer vector with its jth component being the maximum integer less than or equal to $x_{b,j}$, $j = 1, ..., n$ and denote by $\lceil x_b \rceil$ the integer vector with its jth component being the minimum integer greater than or equal to $x_{b,j}$, $j = 1, ..., n$. Let $x^F = \lfloor x_b \rfloor$, and $x^I = \lceil x_b \rceil$. It is easy to see that x^F is a feasible point ($x^F \in S$) and x^I is infeasible ($x^I \notin S$). The monotonicity of f and g_i implies that there are no feasible points better than x^F in $[\alpha, x^F]$ and there are no feasible points in $[x^I, \beta]$. Therefore, we can remove the integer boxes $[\alpha, x^F]$ and $[x^I, \beta]$ from $[\alpha, \beta]$ for further consideration after comparing x^F with the incumbent solution.

The following theorem shows how to cut a revised domain into sub-boxes.

Theorem 1. Let $A = [\alpha, \beta]$, $B = [\alpha, \gamma]$, and $C = [\gamma, \beta]$ integers boxes where $\alpha \leq \gamma \leq \beta$. Then both $A \setminus B$ and $A \setminus C$ can be partitioned into at most n new integer boxes (Xun et al., [18]).

$$A \setminus B = \bigcup_{j=1}^{n} \left(\prod_{k=1}^{j-1} [\alpha_k, \gamma_k] \times [\gamma_j + 1, \beta_j] \times \prod_{k=j+1}^{n} [\alpha_k, \beta_k] \right) \qquad (8)$$

$$A \setminus C = \bigcup_{j=1}^{n} \left(\prod_{k=1}^{j-1} [\gamma_k, \beta_k] \times [\alpha_j, \gamma_j - 1] \times \prod_{k=j+1}^{n} [\alpha_k, \beta_k] \right) \qquad (9)$$

The discrete polyblock method consists of finding a feasible point x^F and an infeasible point x^I and generating integer boxes using the formulas (8) and (9). The best feasible solution obtained during the generation of integer boxes is kept as an incumbent solution. Moreover, by the monotonicity of the problem, an integer box $[\alpha, \beta]$ with $f(\beta)$ less than the objective value $f(x^F)$ of the incumbent x^F can be discarded.

1.2. The solution algorithm

The algorithm of the polyblock method uses the bisection method or Newton's method to find the boundary point. At x_b each iteration of the algorithm, the feasible domain is divided into two subdomains. Each subdomain is analysed in order to discard the one not containing a feasible point which gives an objective function value greater than the value yielded by the incumbent solution.

Initialization. Let $l = (l_1, l_2, ..., l_n)$ and $d = (d_1, d_2, ..., d_n)$.
If l is infeasible, then problem (5) has no feasible solution.
If d is feasible, then d is the optimal solution to (5), stop.
Otherwise, set $x_{opt} = l$, $f_{opt} = f(x_{opt})$, $X^{(1)} = [l, d]$ and set $k = 1$.

Step 1 (box selection and boundary point).
Select a box $[\alpha, \beta] \in X^k$ with the highest objective value $f(\beta)$.
Set $X^{(k)} = X^{(k)} \setminus [\alpha, \beta]$.
Find the root λ^* of the following equation:

$$G[\lambda\alpha + (1-\lambda)\beta] = 0, \quad \lambda \in [0,1]$$

where G is defined in (7).
Set $x_b^{(k)} = \lambda^*\alpha + (1-\lambda^*)\beta$. Set $x^{Fk} = \lfloor x_b^{(k)} \rfloor$ and $x^{Ik} = \lceil x_b^{(k)} \rceil$. x^{Fk} is defined by rounding down and x^{Ik} by rounding up the components of $x_b^{(k)}$.
If $x^{Fk} = x_b^{(k)}$, set $x^{Ik} = x_b^{(k)} + e_j$, where e_j is the jth unit vector in $\mathbb{R}^n$ with $x_b^{(k)} + e_j \leq \beta$.
If $f(x^{Fk}) > f_{opt}$, set $x_{opt} = x^{Fk}$ and $f_{opt} = f(x^{Fk})$.

Step 2 (partition and remove).
Apply the formula (9) to partition the set $\Omega_1^{(k)} = [\alpha, \beta] \setminus [x^{Ik}, \beta]$ into a union of integer boxes.
Let $[\hat{\alpha}, \hat{\beta}]$ the integer box containing the point x^{Fk}. Set $\Omega_1^{(k)} = \Omega_1^{(k)} \setminus [\hat{\alpha}, \hat{\beta}]$.
Apply the formula (8) to partition the set $\Omega_2^{(k)} = [\hat{\alpha}, \hat{\beta}] \setminus [\hat{\alpha}, x^{Fk}]$.
Set $Y^{(k)} = \Omega_1^{(k)} \cup \Omega_2^{(k)}$.
Perform the following for each integer box $[\alpha, \beta]$ generated in the partition process:

- If β is feasible, remove $[\alpha, \beta]$ from $Y^{(k)}$. Furthermore, if $f(\beta) > f_{opt}$, set $x_{opt} = \beta$ and $f_{opt} = f(\beta)$.

- If α is infeasible, remove $[\alpha, \beta]$ from $Y^{(k)}$.

- If $f(\beta) \le f_{opt}$, remove $[\alpha, \beta]$ from $Y^{(k)}$.

- If α is feasible, β is infeasible and $f(\alpha) > f_{opt}$, set $x_{opt} = \alpha$ and $f_{opt} = f(\alpha)$.

Denote $z^{(k)}$ the set of integer boxes after the above removing process.

Step 3 (updating integer boxes).
Remove all integer boxes $[\alpha, \beta]$ in $X^{(k)}$ with $f(\beta) \le f_{opt}$.

Set $X^{(k+1)} = X^{(k)} \cup Z^{(k)}$.

If $X^{(k+1)} = \varnothing$, stop. Otherwise, set $k = k+1$ and go to Step 1.

Two box-selection strategies can be used in Step 1. The first strategy is to select the integer box in $X^{(k)}$ with the maximum objective function value of the upper bound point. The second strategy is to select the last integer box included in $X^{(k)}$. The finite convergence of the algorithm can be easily seen from the finiteness of X and the fact that at each iteration at least the integer points x^{Fk} and x^{lk} are removed from $X^{(k)}$. The algorithm proceeds successively by refining the partition and removing integer boxes that do not contain promising solutions and finally terminates in a finite number of iterations.

1.3. Illustrative example

Let us consider the following example with a similar structure to that of the problem (1), with $\alpha = 0.998$, $b = 125$, $p_1 = 0.80$, $p_2 = 0.90$, $\beta_1 = 400$, $\beta_2 = 200$, $l_1 = 55$, $l_2 = 70$, $d_1 = 60$, $d_2 = 76$. The termination step in the bisection method is set to $\varepsilon = 0.0002$. The two last constraints are added to show that this method applies also to multiple constraints. Problem (5) is formulated as:

$$\max f(x) = 320x_1 + 180x_2$$
$$\text{subject to}$$
$$g_0(x) = 0.80x_1 + 0.90x_2 + 2.9\sqrt{0.16x_1 + 0.09x_2} - 125 \le 0$$
$$g_1(x) = x_1 x_2 + 2x_1 - x_2 - 4380 \le 0$$
$$g_2(x) = x_1 + 5x_2 - 435 \le 0$$
$$x \in X = \left\{ x \in \mathbb{Z}_+^2 \mid 55 \le x_1 \le 60, \quad 70 \le x_2 \le 76 \right\}$$

The feasible set of this problem is given in Fig. 1.

Initialization. Let $l = (55, 70)$, $d = (60, 76)$

$l = (55, 70)$ is feasible, $d = (60, 76)$ is infeasible.

Set $x_{opt} = (55, 70)$ and $f_{opt} = 30\ 200$.

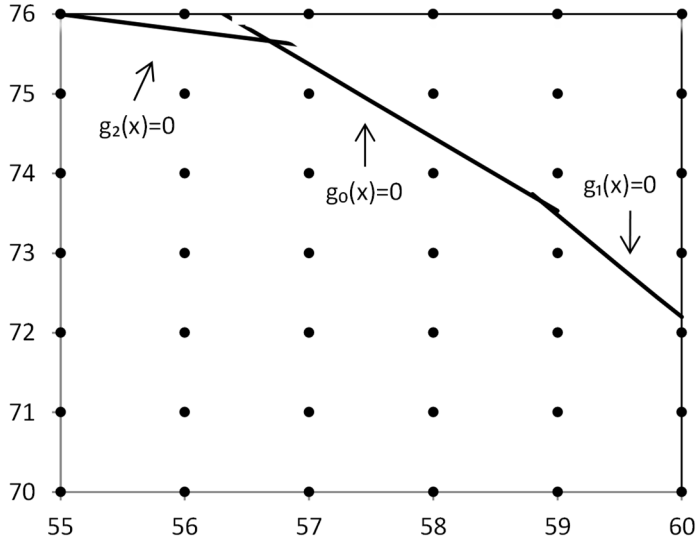$X^{(1)} = [l, d] = [(55, 70), (60, 76)]$; $k = 1$.

Fig. 1. Feasible region

Iteration 1. Step 1. Select $[\alpha, \beta] = [l, d] = [(55, 70), (60, 76)]$. Set

$$X^{(1)} = X^{(1)} \setminus [\alpha, \beta] = \varnothing$$

We use the bisection procedure to solve the equation

$$G[\lambda(55, 70) + (1 - \lambda)(60, 76)] = 0$$

This method finds out the root $\lambda^* = 0.31960$. Then the intersection point is given by: $x_b^{(1)} = (58.4020, 74.0824)$. The points $x^{F1} = (58, 74)$ and $x^{I1} = (59, 75)$ are represented in Fig. 2.

Since $f(x^{F1}) = 31\ 880 > 30\ 200 = f_{opt}$, set $x_{opt} = (58, 74)$ and $f_{opt} = 31\ 880$.

Step 2. Partition the set $\Omega_1^{(1)} = [\alpha, \beta] \setminus [x^{I1}, \beta]$ into two integer boxes.

The increase in f implies that there are no feasible points in $\left[x^{I1}, \beta\right] = [(59, 75), (60, 76)]$. Therefore, we can remove this box from the box $\left[\alpha, \beta\right] = [(55, 70), (60, 76)]$. Formula (9) gives us:

$$\begin{aligned}
\Omega_1^{(1)} &= \left[(55, 70), (60, 76)\right] \setminus \left[(59, 75), (60, 76)\right] \\
&= \left\{\left[\alpha_1, \gamma_1 - 1\right] \times \left[\alpha_2, \beta_2\right]\right\} \cup \left\{\left[\gamma_1, \beta_1\right] \times \left[\alpha_2, \gamma_2 - 1\right]\right\} \\
&= \left\{\left[55, 58\right] \times \left[70, 76\right]\right\} \cup \left\{\left[59, 60\right] \times \left[70, 74\right]\right\} \\
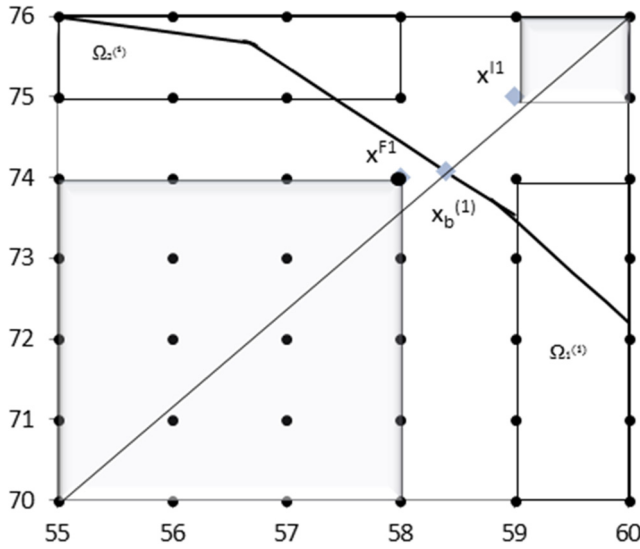&= \left[(55, 70), (58, 76)\right] \cup \left[(59, 70), (60, 74)\right]
\end{aligned}$$

Fig. 2. Illustration for iteration 1

Since $x^{F1} \in \left[\hat{\alpha}, \hat{\beta}\right] = [(55, 70), (58, 76)]$, set $\Omega_1^{(1)} = \Omega_1^{(1)} \setminus \left[\hat{\alpha}, \hat{\beta}\right] = [(59, 70), (60, 74)]$. Since f is an increasing function, there are no feasible points better than x^{F1} in $\left[\hat{\alpha}, x^{F1}\right]$ = [(55, 70), (58, 74)].

Then, we can remove the box $\left[\hat{\alpha}, x^{F1}\right]$ from the box $\left[\hat{\alpha}, \hat{\beta}\right]$ by applying formula (8).

$$\Omega_2^{(1)} = \left[\hat{\alpha}, \hat{\beta}\right] \setminus \left[\hat{\alpha}, x^{F1}\right]$$
$$= \left[(55, 70), (58, 76)\right] \setminus \left[(55, 70), (58, 74)\right]$$
$$= \left\{[\gamma_1 + 1, \beta_1] \times [\alpha_2, \beta_2]\right\} \cup \left\{[\alpha_1, \gamma_1] \times [\gamma_2 + 1, \beta_2]\right\}$$
$$= \left\{[59, 58] \times [70, 76]\right\} \cup \left\{[55, 58] \times [75, 76]\right\}$$
$$= \emptyset \cup \left\{[55, 58] \times [75, 76]\right\}$$
$$= \left[(55, 75), (58, 76)\right]$$

Set $Y^{(1)} = \Omega_1^{(1)} \cup \Omega_2^{(1)} = \left[(59, 70), (60, 74)\right] \cup \left[(55, 75), (58, 76)\right]$.

None of the integer boxes in $Y^{(1)}$ is removed, then, $Z^{(1)} = Y^{(1)}$. Only the coloured parts are removed.

Step 3. Set $X^{(2)} = X^{(1)} \cup Z^{(1)} = \emptyset \cup Z^{(1)} = \left[(59, 70), (60, 74)\right] \cup \left[(55, 75), (58, 76)\right]$

.

Iteration 2. Step 1 (we use the strategy which selects the integer box in $X^{(k)}$ with the maximum objective function value of the upper bound point).

Since $f(60, 74) = 32\ 520 > 32\ 240 = f(58, 76)$ select $[\alpha, \beta] = [(59, 70), (60, 74)]$ and set $X^{(2)} = X^{(2)} \setminus [\alpha, \beta] = [(55, 75), (58, 76)]$.
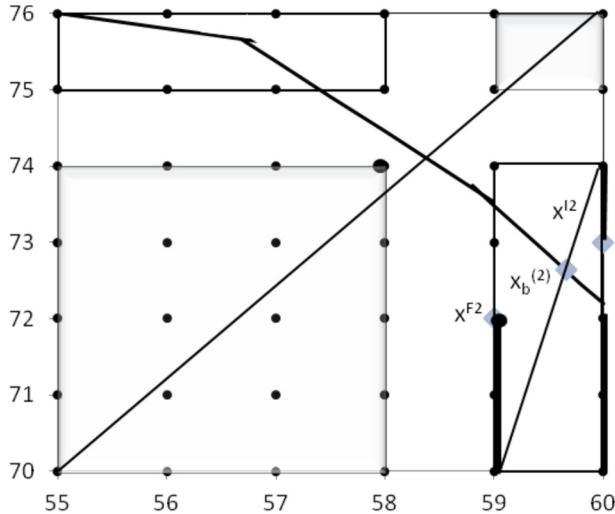

Fig. 3. Illustration for iteration 2

The bisection procedure applied to the equation $G\left[\lambda(59, 70) + (1 - \lambda)(60, 74)\right] = 0$ finds out $\lambda^* = 0.34123$ and $x_b^{(2)} = (59.65877, 72.63508)$. Then, $x^{F2} = (59, 72)$ and $x^{I2} = (60, 73)$ (see Fig. 3).

Since $f(x^{F2}) = 31\ 840 < 31\ 880$ then, $x_{\mathrm{opt}} = (58, 74)$ and $f_{\mathrm{opt}} = 31\ 880$.

$$\Omega_2^{(2)} = \left[\hat{\alpha}, \hat{\beta}\right] \backslash \left[\hat{\alpha}, x^{F2}\right]$$

$$= \left[(59, 70), (59, 74)\right] \backslash \left[(59, 70), (59, 72)\right]$$

$$= \left\{[60, 59] \times [70, 74]\right\} \cup \left\{[59, 59] \times [73, 74]\right\}$$

$$= \varnothing \cup \left\{[59, 59] \times [73, 74]\right\}$$

$$= \left[(59, 73), (59, 74)\right]$$

Set $Y^{(2)} = \Omega_1^{(2)} \cup \Omega_2^{(2)} = \left[(60, 70), (60, 72)\right] \cup \left[(59, 73), (59, 74)\right]$.

The point $(60, 72)$ is feasible and $f(60, 72) = 32\ 160 > 31\ 880$. Set $x_{\mathrm{opt}} = (60, 72)$ and $f_{\mathrm{opt}} = 32\ 160$. Remove the box $\left[(60, 70), (60, 72)\right]$.

Since $f(59, 74) = 32\ 200 > 32\ 160 = f_{\mathrm{opt}}$, the box $\left[(59, 73), (59, 74)\right]$ must be scanned. So, we do not remove it from $Y^{(2)}$. Therefore, $Z^{(2)} = \left[(59, 73), (59, 74)\right]$ (coloured parts and bold lines are removed).

Step 3. For the box $\left[(55, 75), (58, 76)\right] \in X^{(2)}$, we have $f(58, 76) = 32\ 240 > 32\ 160 = f_{\mathrm{opt}}$. Then this box must be explored.

Set $X^{(3)} = X^{(2)} \cup Z^{(2)} = \left[(55, 75), (58, 76)\right] \cup \left[(59, 73), (59, 74)\right]$.

Iteration 3. Step 1. Since $f(58, 76) > f(59, 74)$, select $\left[\alpha, \beta\right] = \left[(55, 75), (58, 76)\right]$ and set $X^{(3)} = X^{(3)} \backslash \left[\alpha, \beta\right] = \left[(59, 73), (59, 74)\right]$.

The bisection procedure applied to the equation $G[\lambda(55, 75) + (1 - \lambda)(58, 76) = 0$ finds out $\lambda^* = 0.41162$ and $x_b^{(3)} = (56.76514, 75.58838)$. Figure 4 shows two points $x^{F3} = (56, 75)$ and $x^{I3} = (57, 76)$. $f(x^{F3}) = 31\ 420 < 32\ 160 = f_{\mathrm{opt}}$ then, $x_{\mathrm{opt}} = (60, 72)$, and $f_{\mathrm{opt}} = 32\ 160$.
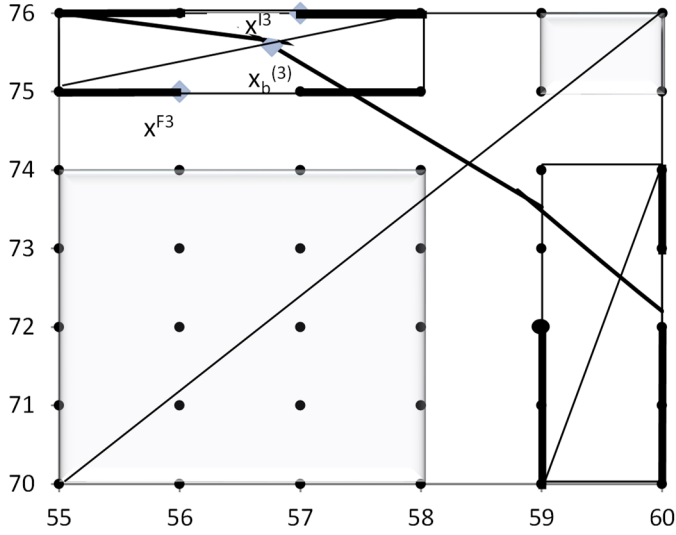
Fig. 4. Illustration for iteration 3

Step 2. Partition the set $\Omega_1^{(3)} = [\alpha, \beta] \setminus [x^{I3}, \beta]$ into two integer boxes.

$$\Omega_1^{(3)} = [(55, 75), (58, 76)] \setminus [(57, 76), (58, 76)]$$
$$= \{[55, 56] \times [75, 76]\} \cup \{[57, 58] \times [75, 75]\}$$
$$= [(55, 75), (56, 76)] \cup [(57, 75), (58, 75)]$$

Since $x^{F3} \in [\hat{\alpha}, \hat{\beta}] = [(55, 75), (56, 76)]$ set $\Omega_1^{(3)} = \Omega_1^{(3)} \setminus [\hat{\alpha}, \hat{\beta}] = [(57, 75), (58, 75)]$.

$$\Omega_2^{(3)} = [\hat{\alpha}, \hat{\beta}] \setminus [\hat{\alpha}, x^{F3}]$$
$$= [(55, 75), (56, 76)] \setminus [(55, 75), (56, 75)]$$
$$= \{[57, 56] \times [75, 76]\} \cup \{[55, 56] \times [76, 76]\}$$
$$= \varnothing \cup [55, 76] \times [56, 76]$$
$$= [55, 76] \times [56, 76]$$

Set $Y^{(3)} = \Omega_1^{(3)} \cup \Omega_2^{(3)} = [(57, 75), (58, 75)] \cup [(55, 76), (56, 76)]$.

$f(58, 75) = 32\ 060 < 32\ 160 = f_{opt}$ and $f(56, 76) = 31\ 600 < 32\ 160 = f_{opt}$, therefore, remove the two boxes $[(57, 75), (58, 75)]$ and $[(55, 76), (56, 76)]$ from $Y^{(3)}$. Consequently, $Z^{(3)} = \emptyset$.

Step 3. For the box $[(59, 73), (59, 74)] \in X^{(3)}$, we have $f(59, 74) = 32\ 220 > 32\ 160 = f_{opt}$. This box must be explored. Then, $X^{(4)} = X^{(3)} \cup Z^{(3)} = [(59, 73), (59, 74)]$.

Iteration 4. Step 1. Select $[\alpha, \beta] = [(59, 73), (59, 74)]$ and set $X^{(4)} = X^{(4)} \setminus [\alpha, \beta] = \emptyset$.

The bisection procedure applied to the equation $G[\lambda(59, 73) + (1 - \lambda)(59, 74)] = 0$ finds out $\lambda^* = 0.51724$ and $x_b^{(4)} = (59, 73.48276)$. The two points $x^{F4} = (59, 73)$ and $x^{I4} = (59, 74)$ are given in Fig. 5. $f(x^{F4}) = 32\ 020 < 32\ 160 = f_{opt}$, then the point $(60, 72)$ is still the incumbent optimal solution.
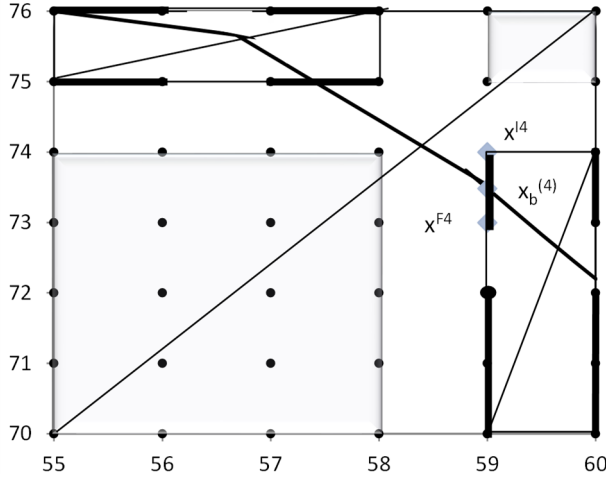

Fig. 5. Illustration for iteration 4

Step 2. Partition the set $\Omega_1^{(4)} = [\alpha, \beta] \setminus [x^{I4}, \beta]$ into two integer boxes.

$$\begin{aligned}
\Omega_1^{(4)} &= [(59, 73), (59, 74)] \setminus [(59, 74), (59, 74)] \\
&= \{[59, 58] \times [73, 74]\} \cup \{[59, 59] \times [73, 73]\} \\
&= \emptyset \cup [(59, 73), (59, 73)] \\
&= [(59, 73), (59, 73)]
\end{aligned}$$

$$x^{F4} \in \left[\hat{\alpha}, \hat{\beta} \right] = \left[(59, 73), (59, 73) \right], \text{ set } \Omega_1^{(4)} = \Omega_1^{(4)} \setminus \left[\hat{\alpha}, \hat{\beta} \right] = \varnothing .$$

$$
\begin{aligned}
\Omega_2^{(4)} &= \left[\hat{\alpha}, \hat{\beta} \right] \setminus \left[\hat{\alpha}, x^{F4} \right] \\
&= \left[(59, 73), (59, 73) \right] \setminus \left[(59, 73), (59, 73) \right] \\
&= \varnothing
\end{aligned}
$$

$Y^{(4)} = \Omega_1^{(4)} \cup \Omega_2^{(4)} = \varnothing$. Consequently, $Z^{(4)} = \varnothing$ and $X^{(5)} = X^{(4)} \cup Z^{(4)} = \varnothing$. The optimal solution of the problem is $x_{\text{opt}} = (60, 72)$ with objective value $f_{\text{opt}} = 32\ 160$.

2. Numerical experience and comparison

In order to investigate the potential of the discrete polyblock algorithm when applied to the considered chance-constrained problem, we implemented it and tested it on small problems similar to the mathematical model (4) where supplementary constraints are added. Our results are compared in terms of running time (CPU time) and the number of iterations (N/It) to those given by four solvers: LINGO, COUENNE, BONMIN, and BARON. The first three solvers are designed to solve general mixed-integer nonlinear problems, while the last solver is designed for nonconvex mixed-integer nonlinear ones.

The first problem corresponds to the example given in this article. For problems 2 and 3, we kept the same constraints as in problem 1 and modify the variation ranges of the decision variables. In problem 4, the variance and the intervals are modified. Three additional constraints are introduced in problem 4 to obtain problem 5 (see Table 1).

After performing our tests presented in Tables 2 and 3, we can confirm that the polyblock method gives exactly the same solution as LINGO, COUENNE, BONMIN and BARON. Table 2 shows that polyblock algorithm is efficient, compared with LINGO and BONMIN in terms of the number of iterations required to reach the optimum. The number of bisection calls (N/BC) is equal to the number of constraints multiplied by the number of iterations. The results in Table 3 show that polyblock algorithm is faster than LINGO but slower than COUENNE, BONMIN, and BARON. Much of the computational time was spent on locating the boundary point of the feasible set. We have used two approaches for finding the boundary points: Newton's technique and the Bolzano's bisection procedure. The two approaches produced similar results.

Table 1. Test problems

Problem 1	$\max f(x) = 320x_1 + 180x_2$ subject to $g_0(x) = 0.80x_1 + 0.90x_2 + 2.9\sqrt{0.16x_1 + 0.09x_2} - 125 \leq 0$ $g_1(x) = x_1x_2 + 2x_1 - x_2 - 4380 \leq 0$ $g_2(x) = x_1 + 5x_2 - 435 \leq 0$ $55 \leq x_1 \leq 60, \quad 70 \leq x_2 \leq 76$
Problem 2	$\max f(x) = 320x_1 + 180x_2$ subject to $g_0(x) = 0.80x_1 + 0.90x_2 + 2.9\sqrt{0.16x_1 + 0.09x_2} - 125 \leq 0$ $g_1(x) = x_1x_2 + 2x_1 - x_2 - 4380 \leq 0$ $g_2(x) = x_1 + 5x_2 - 435 \leq 0$ $3 \leq x_1 \leq 77, \quad 35 \leq x_2 \leq 100$
Problem 3	$\max f(x) = 320x_1 + 180x_2$ subject to $g_0(x) = 0.80x_1 + 0.90x_2 + 2.9\sqrt{0.16x_1 + 0.09x_2} - 125 \leq 0$ $g_1(x) = x_1x_2 + 2x_1 - x_2 - 4380 \leq 0$ $g_2(x) = x_1 + 5x_2 - 435 \leq 0$ $1 \leq x_1 \leq 100, \quad 1 \leq x_2 \leq 66$
Problem 4	$\max f(x) = 320x_1 + 180x_2$ subject to $g_0(x) = 0.80x_1 + 0.90x_2 + 2.9\sqrt{x_1 + 0.09x_2} - 125 \leq 0$ $g_1(x) = x_1x_2 + 2x_1 - x_2 - 4380 \leq 0$ $g_2(x) = x_1 + 5x_2 - 435 \leq 0$ $1 \leq x_1 \leq 100, \quad 1 \leq x_2 \leq 100$
Problem 5	$\max f(x) = 320x_1 + 180x_2$ subject to $g_0(x) = 0.80x_1 + 0.90x_2 + 2.9\sqrt{x_1 + 0.09x_2} - 125 \leq 0$ $g_1(x) = x_1x_2 + 2x_1 - x_2 - 4380 \leq 0$ $g_2(x) = x_1 + 5x_2 - 435 \leq 0$ $g_3(x) = 5x_1x_2 - 4x_1 - 4.5x_2 - 320 \leq 0$ $g_4(x) = x_1^3 - 3900 \leq 0$ $g_5(x) = (x_1 - 2)^3 + x_2 - 650 \leq 0$ $1 \leq x_1 \leq 100, \quad 1 \leq x_2 \leq 100$

Table 2. Efficiency of various algorithms in terms of the number of iterations

Problem	Solution	POLYBLOCK		LINGO N/It	COUENNE N/It	BONMIN N/It	BARON N/It
		N/It	N/BC				
1	(60, 72)	4	12	41	8	6	1
2	(77, 55)	5	15	49	16	24	1
3	(100, 35)	6	18	95	11	44	1
4	(100, 17)	5	15	55	3	28	1
5	(1, 86)	3	18	54	1	22	1

Table 3. Efficiency of various algorithms in terms of running time [s]

Problem	Solution	POLYBOCK	LINGO	COUENNE	BONMIN	BARON
1	(60, 72)	0.185	0.44	0.00	0.01	0.02
2	(77, 55)	0.178	0.78	0.00	0.05	0.02
3	(100, 35)	0.203	0.36	0.00	0.05	0.03
4	(100, 17)	0.197	0.55	0.00	0.05	0.03
5	(1, 86)	0.093	0.11	0.02	0.09	0.02

In sum, to solve this particular problem, it is better to apply BARON or the specific algorithm polyblock that takes into account its special structure instead of solving it with solvers designed for general mixed integer nonlinear problems such as LINGO, COUENNE, and BONMIN. This allows us to reduce the number of iterations and/or to reach the optimal solution in a minimum of time.

3. Conclusion

In this study, a special stochastic integer program is considered. The problem is first reformulated into a deterministic monotonic integer problem introducing the chance-constraint programming approach. The resulting monotonic program is solved by the polyblock method which exploits its special structure. This method does not require specific mathematical properties to be satisfied by the objectives or the constraints and treats the cutting plane gap which appear in solving integer programming problems. Indeed, no linearization or convexification of the nonlinear functions is needed, which distinguishes it from the other known methods. Its efficiency is due to the fact that one has just to solve simple equations rather than complex optimization problems in order to find the optimal solutions for the main problem. Even if this algorithm is not fast compared to some solvers, it can be used as a support for other research works.

Acknowledgements

References

[1] ABHISHEK K., LEYFFER S., LINDEROTH J., *FilMINT. An outer approximation-based solver for convex mixed-integer nonlinear programs*, INFORMS J. Comp., 2010, 22, 555–567.

[2] BELOTTI P., *Couenne. A user's manual*, https://projects.coin-or.org/Couenne

[3] BONAMI P., BIEGLER L.T., CONN A.R., CORNUJOLS G., GROSSMANN I.E., LAIRD C.D., LEE J., LODI A., MARGOT F., SAWAYA N., WACHTER A., *An algorithmic framework for convex mixed integer nonlinear programs*, Disc. Opt., 2008, 5, 186–204.

[4] BONAMI P., KILINC M., LINDEROTH J.T., *Algorithms and software for solving convex mixed integer nonlinear programs, mixed integer nonlinear programming*, IMA Vol. Math. Appl., 2012, 154, 1–39.

[5] CHARNES A., COOPER W.W., *Deterministic equivalents for optimizing and satisfying under chance constraints*, Oper. Res., 1963, 11, 18–39.

[6] COIN-OR. https://www.coin-or.org › Bonmin

[7] DURAN M.A., GROSSMANN I., *An outer-approximation algorithm for a class of mixed-integer nonlinear programs*, Math. Progr., 1986, 36, 307–339.

[8] FLETCHER R., LEYFFER S., *Solving mixed integer nonlinear programs by outer approximation*, Math. Progr., 1994, 66, 327–349.

[9] GEOFFRION A., *Generalized benders decomposition*, J. Opt. Theory Appl., 1972, 10, 237–260.

[10] HOAI-PHUONG N.T., TUY H., *A unified monotonic approach to generalized linear fractional programming*, J. Global Opt., 2003, 26, 229–259.

[11] KELLEY J.E., *The cutting plane method for solving convex programs*, J. SIAM, 1960, 8, 703–712.

[12] KILINÇ M.R., SAHINIDIS N.V., *Exploiting integrality in the global optimization of mixed-integer nonlinear programming problems with BARON*, Opt. Meth. Soft., 2018, 33 (3), 540–562.

[13] LINDO Systems, Inc., *LINGO. The modeling language and optimizer*, 2017. http://www.lindo.com

[14] LINDO Systems, Inc., *Optimization modeling with LINGO*, Technical Support in China, 2015. http://www.lindochina.com

[15] LI D., SUN X.L., BISWAL M.P., GAO F., *Convexification, concavification and monotonization in global optimization*, Ann. Oper. Res., 2001, 105, 213–226.

[16] Neos-server. https://neos-server.org/neos/solvers/go:ANTIGONE

[17] QUESADA I., GROSSMANN I.E., *An LP/NLP based branch-and-bound algorithm for convex MINLP optimization problems*, Com. Chem. Eng., 1992, 16, 937–947.

[18] SUN X.L., LI J.L., *Nonlinear integer programming*, Springer, 2006.

[19] TAWARMALANI M., SAHINIDIS N.V., *Global optimization of mixed integer nonlinear programs. A theoretical and computational study*, Math. Progr., 2004, 99, 563–591.

[20] TUY H., *Monotonic optimization. Problems and solution approaches*, SIAM J. Opt., 2000, 11 (2), 464–494.

[21] TUY H., THACH P.T., KONNO H., *Optimization of polynomial fractional functions*, J. Global Opt., 2004, 29, 19–44.

[22] TUY H., MINOUX M., HOAI-PHUONG N.T., *Discrete monotonic optimization with application to a discrete location problem*, SIAM J. Opt., 2006, 17 (1), 78–97.

[23] VIGERSKE S., GLEIXNER A., *SCIP. Global optimization of mixed-integer nonlinear programs in a branch-and-cut framework*, Opt. Meth. Soft., 33 (3), 2018, 536–539.

[24] WESTERLUND T., PETTERSSON F., *A cutting plane method for solving convex MINLP problems*, Comp. Chem. Eng., 1995, 19, 131–136.