Pesquisa Operacional (2026) 46: e301123 p.1-31
doi: 10.1590/0101-7438.2026.046.00301123
© 2026 Brazilian Operations Research Society
Printed version ISSN 0101-7438 / Online version ISSN 1678-5142
www.scielo.br/pope
ARTICLES

EXACT ALGORITHMS FOR GENERATING THE EFFICIENT SET OF MULTI-OBJECTIVE STOCHASTIC INTEGER LINEAR PROGRAMS

Sarra Kentache[1*] and Djamal Chaabane[2]

Received September 20, 2025 / Accepted September 1, 2026

ABSTRACT. In this paper, we propose two new methods for generating the set of all efficient solutions to a discrete multi-objective optimization problem under uncertainty, also known as stochastic multiple objective integer linear programming problem. The suggested approaches follows the standard state-of-the-art formulation requiring a two-stage integer stochastic formulation with fixed recourse and using the L-shaped method as a main framework. Furthermore, the suggested method can be perceived as adaptations of deterministic discrete multi-objective optimization methods to the stochastic context. Finally, the obtained computational results show that our methods can solve large-scale randomly generated instances. A comprehensive computational study on structured benchmark problems, including the well known George B. Dantzig (GBD) benchmark instances and the newly introduced stochastic bi-objective knapsack problem (SBKP) benchmark instances, further demonstrates the practical efficiency of the proposed approaches.

Keywords: stochastic programming, integer programming, multi-objective programming, L-shaped method, linear programming.

1 INTRODUCTION

In real-world problems, multi-objective modeling is inevitable. Therefore, finding a solution that simultaneously optimizes all objectives is almost impossible due to conflicting objectives. Subsequently, the aim is to find a subset of feasible solutions that realize the best compromise between objectives. The multi-objective linear problems in the continuous case have been widely studied and several resolution methods have been introduced by researchers. The reader may refer to Branke (2008) for a full literature review. The introduction of discrete variables makes them significantly more complex to solve. However, a major issue with this type of problem is that the admissible set is not convex, which induces significant difficulties. In the last decade, several methods have been developed to solve the MOILP, among them we mention: Zionts (1977),

*Corresponding author
[1]USTHB, Faculty of Mathematics, Department of Operations Research, Laboratory AMCD-RO, Bab-Ezzouar, Algiers, Algeria – E-mail: skentache@usthb.dz / https://orcid.org/0009-0004-8514-5707
[2]USTHB, Faculty of Mathematics, Department of Operations Research, Laboratory AMCD-RO, Bab-Ezzouar, Algiers – E-mail: djamel.chaabane@usthb.edu.dz / https://orcid.org/0000-0002-7722-311X

Klein & Hannan (1982), Marcotte & Soland (1986), Teghem Jr et al. (1986), Sylva & Crema (2004), Sylva & Crema (2007), Chankong & Haimes (2008), Lokman & Köksalan (2013) and Helfrich et al. (2023). Additionally, multi-objective problems have been addressed from several point-of-views, according to practical requirements such as decision maker's preferences, called the optimization over the efficient set problem. We mention, for example, Zaidi et al. (2025) in deterministic case, and Badaoui et al. (2024) in stochastic case. We note that generating the entire efficient set and optimizing over an efficient set are different problems in multi-objective optimization. The former aims to enumerate all efficient solutions, whereas the latter aims to optimize a given preference function over the efficient frontier. In particular, Badaoui et al. (2024) address the optimization over the efficient set problem, relying on the branch-and-bound framework of Amrouche & Moulaï (2012) as their underlying search strategy. In contrast, our work focuses on efficiently generating the complete efficient set using novel algorithmic strategies, positioning the method of Amrouche & Moulaï (2012) as our primary point of comparison.In the rest of this section, we present an overall view over the research work related to this paper followed by a section of background.

1.1 Literature review

The main characteristic of the problems that emerge from industry is uncertainty. Thus, in this situation, stochastic optimization is a potential alternative for solving such problems. Stochastic optimization was pioneered by George Dantzig (1955). The problem essentially involves two directions in terms of formulation and solving approaches. First, we should mention that stochastic programming problems are commonly recognized as being difficult, both theoretically and computationally. To be more precise, their difficulty manifests in the mathematical modeling of these problems. This latter consists of two categories of data: stochastic and deterministic. Consequently, for a stochastic optimization problem, the mathematical model is not necessarily unique. The commonly used formulations and approaches in the literature are: a model with chance constraints see Charnes et al. (1958), Abdelaziz (2012). Such a model has at least one constraint that can be violated with a certain probability, which somehow represents the reliability of the system to be modeled. Furthermore, we mention two alternative stochastic linear programming modeling approaches: "Wait and see" and "here and now". In "Wait and See" (W-S) models, one would actually be allowed to wait for the realization of the random experiment, then solves the deterministic (non-stochastic) linear programming problem based on the corresponding values that are observed. This is known as a situation of a priory information (see Madansky (1960), Bereanu (1967), and King (1986)). Regarding the "Here and Now" (H-N) class of stochastic programming, it deals primarily with problems where a decision must be made about the field under uncertainty without waiting for the realization of certain random variables. The combination of the last two models results in the new two-stage model.

In 1955, Dantzig, presented the two-stage linear program model under uncertainty. This approach was further developed for future investigation by by Wets (1966), Walkup & Wets (1967), and Walkup & Wets (1969). His strategy for the decision variable is based on an integration and

dynamic relation between the W-S and H-N models, also known as two-stage programming (see Slowinski & Teghem Jr (1988), Bakir & Byrne (1998), and Wets (2002)). This approach can be summarized as follows: in the first stage, some initial decisions are made that optimize current costs while minimizing recourse costs, and the feasibility issue is addressed in the second stage. This cutting plane technique is called the L-shaped method in stochastic programming. The idea of feasibility and optimality cuts in the dual decomposition method may be traced to Benders (1962). This method has been extended into stochastic programming under the name of the L-shaped method. The L-shaped method for two-stage stochastic linear programming with continuous variables in both stages was first proposed by Van Slyke & Wets (1969).Laporte & Louveaux (1993) presented the integer L-shaped method, which is also based on Benders decomposition methodology. Only a limited number of research papers focus on pure integer variables (see Schultz et al. (1998), Ahmed et al. (2004), and Kong et al. (2006)).

Stochastic optimization and multi-objective optimization have seen impressive progress. Notice that these two domains developed largely independently of one another, and their interaction is still covered by only a small number of publications including Goicoechea (1977), Teghem (1983), Stancu-Minasian (1984), Teghem Jr et al. (1986), Slowinski & Teghem Jr (1988), Urli & Nadeau (1990), Kall & Wallace (1994), and León et al. (2020). These methods have been applied to real problems such as planning electricity contracts, location under uncertainty, process engineering, and many more. Moreover, in the case of the stochastic multi-objective linear problem with integer variables, and to the best of our knowledge, this problem is solely addressed by Amrouche & Moulaï (2012). This method is based on classical branch and bound approach and L-shaped method. In this work, the main contribution is the development of two methods for Multi-Objective Integer Stochastic Linear Programming problem (SMOILP), for two-stage models with fixed-recourse. The suggested methods can be perceived as adaptations of state-of-the-art deterministic algorithms. The first method is based on Sylva & Crema (2004), we call it Stochastic Sylva and Crema (SSC), and the second is based on Lokman & Köksalan (2013), we call it Stochastic Lokman and Kökslan (SLK).

This paper is organized as follows: in the next section, we present an overview of the addressed problem and the related basic concepts. In Section 3, we present a description of the suggested methods with detailed steps and we illustrate them with didactic examples. In section 4, we report the results of the computational experiment, in which, we assess the performance of the proposed algorithms against the state-of-the-art algorithm introduced by Amrouche & Moulaï (2012). Finally, we conclude the paper with a discussion of the proposed methods and directions for future extensions.

2 BACKGROUND

This section introduces the basic notions required for the proposed methods, including: basic concepts of multi-objective programming in general and stochastic programming with multiple objective function, mainly, we recall the well known two stage approach with fixed recourse.

2.1 Multi-objective Programming

A Multi-Objective Integer Linear Problem (MOILP) can be stated as follows:

$$(MOILP)\begin{cases} \text{"max"} & z_k(x) = c^k x, \quad k \in \{1,\dots,p\}, \\ s.t., & x \in S. \end{cases} \tag{1}$$

Where, $S = D \cap \mathbb{Z}^n$, $D = \{x \in \mathbb{R}^n : Ax \leq b,\ x \geq 0\}$, $A \in \mathbb{Z}^{m\times n}$, $b \in \mathbb{Z}^m$, $c^k \in \mathbb{Z}^n$, and $p \geq 2$. We assume throughout this paper that S is not empty and D is a bounded convex polyhedron.

Efficiency and dominance can be defined as follows: a feasible solution $x^* \in S$ is an efficient solution if and only if there is no $x \in S$ such that $z_k(x) \geq z_k(x^*)$, for all objective function $k \in \{1,\dots,p\}$ and $z_k(x) > z_k(x^*)$ for at least one i. Otherwise, x^* is not an efficient solution, and $(z_1(x^*), z_2(x^*), \dots, z_p(x^*))$ is said to be dominated. A feasible solution $x^* \in S$ is weakly efficient if there is no other $x \in S$ such that, we have $z_k(x) > z_k(x^*)$, for all $k \in \{1,\dots,p\}$. If x^* is weakly efficient, then $z(x^*)$ is called a weakly non-dominated point.

The following proposition states that the optimal solution obtained from a single-objective linear program (P_t) below is also an efficient solution of MOILP. This result allows us to avoid weakly efficient solutions by introducing a small positive constant.

Proposition 1. *Consider the following problem:*

$$(P_t)\begin{cases} \max & c^t x + \varepsilon \displaystyle\sum_{k\neq t} c^k x, \\ s.t., & x \in S. \end{cases} \tag{2}$$

The optimal solution x^* of (P_t) is an efficient solution of (MOILP). To avoid weakly efficient solutions, we choose an ε sufficiently small strictly positive constant (see Steuer (1986)), thus this objective function generates the best efficient solution in terms of the t^{th} objective.

Using the program (P_t), we can construct a program that allows to enumerate all efficient solutions as follows :

Proposition 2. *Sylva & Crema (2004) Consider the following single-objective integer linear program $t = p$, at step (ℓ):*

$$(P^{(\ell)})\begin{cases} \max & c^p x + \varepsilon \displaystyle\sum_{k=1}^{p-1} c^k x, & & \\ s.t., & Ax \leq b, & & \\ & c^k x \geq (c^k x^i + 1) y_k^i - M_k(1 - y_k^i), & \forall\ i \in \{1,\dots,\ell\}\ ,\forall\ k \in \{1,\dots,p\}, \\ & \displaystyle\sum_{k=1}^{p-1} y_k^i \geq 1, & \forall\ i \in \{1,\dots,\ell\}, \\ & y_k^i \in \{0,1\}, & \forall\ i \in \{1,\dots,\ell\}\ ,\forall\ k \in \{1,\dots,p\}, \\ & x \in \mathbb{N}. & \end{cases} \tag{3}$$

Let $x^{(1)},\dots,x^{(\ell)}$ be efficient solutions to the problem (MOILP). If $\hat{x}$ is an optimal solution to $(P^{(\ell)})$, then $\hat{x}$ is an efficient solution to problem (MOILP). Furthermore, if problem $(P^{(\ell)})$ is infeasible,

$\{cx^i\}_{i=1}^{\ell}$ is the entire set of non-dominated criterion vectors for problem (MOILP).
Where $M = (M_1\ M_2 \ldots M_p)$ is a sufficiently large positive constant used to create a lower bound for $z_k(x) = c^k x$.

2.2 Stochastic programming

Consider the Stochastic Multiple Objective Integer Linear Programming problem (SMOILP) which is defined as follows Chaabane & Mebrek (2013):

$$(SMOILP)\begin{cases} \text{"max"} & z_k(\xi) = c^k(\xi)x, \quad k \in \{1,2,\ldots,p\}, \\ s.t., & Ax \leq b, \\ & T(\xi)x \leq h(\xi), \\ & x \in \mathbb{N}^n. \end{cases} \tag{4}$$

Here, the set $S = \{x \in \mathbb{R}^n : Ax \leq b, x \in \mathbb{N}^n\}$ represents the deterministic constraints, $c^k(\xi)$, T(ξ) and $h(\xi)$ are matrices of random variables of respective dimensions: $(1 \times n)$, $(m_1 \times n)$, and $(m_1 \times 1)$ with probability vector $\mathscr{P}^{(r)}$, where $r \in \{1,2,\ldots,R\}$ which is defined on a probability space $(\Omega, E, \mathbb{P})$.

The corresponding deterministic problem is defined as follows:

$$(P^0)\begin{cases} \text{"}\max_x\text{"} & \mathbb{E}(z_k) = \mathbb{E}(c^k),\ k \in \{1,\ldots,p\}, \\ s.t., & x \in S. \end{cases} \tag{5}$$

The general formulation of a two-stage stochastic problem with fixed recourse, with expected value $\tilde{z} = \mathbb{E}(z_k)$, is represented by the following mathematical program:

$$(PS)\begin{cases} \text{"}\max_x\text{"} & \tilde{z}_k = \tilde{c}^k x - Q(x), \quad k \in \{1,\ldots,p\}, \\ s.t., & x \in S. \end{cases} \tag{6}$$

Where, the function Q is the expected minimum penalty value given by

$$Q(x) = \sum_{r=1}^{R} \mathscr{P}^{(r)} \times Q(\xi^{(r)}, x), \tag{7}$$

and thus, the recourse problem is defined by

$$Q(\xi^{(r)}, x) = \min\left\{ q(\xi^{(r)})y \mid Wy = h(\xi^{(r)}) - T(\xi^{(r)})x,\ y \geq 0 \right\}. \tag{8}$$

In the above question, $q(\xi)$ represents the penalty cost vector relative to the scenario ξ. In our case, we assumes that the fixed recourse matrix $W \in \mathbb{N}^{m_1 \times n_1}$.

Note that, for a given scenario r, the stochastic constraints $T(\xi^{(r)})x \leq h(\xi^{(r)})$ of (4) form a single block, common to all the objective functions $z_k(\xi^{(r)}), k \in \{1,2,\ldots,p\}$. The recourse repairs a violation of these constraints; its cost is accordingly assumed common to all the objective function, the penalty cost vector $q(\xi^{(r)})$ not depending on the criterion index. The recourse problem (8) is then a single-objective linear program, so that the expected minimum penalty value $Q(x)$ depends only on the first stage decision x and is subtracted from each objective function in (6).

2.2.1 L-Shaped Method

The L-Shaped decomposition method, also known as the Benders decomposition method, is a fundamental approach for solving two-stage linear problems with recourse. It was initially introduced by Benders (1962) and later extended by Van Slyke & Wets (1969). This method consists mainly of dividing the problem into two-stages. The main problem uses a linear approximation of the recourse function through an auxiliary variable to determine the first stage decision. This will be used in the second stage to generate feasibility and optimality cuts based on the solution of the dual problem for different scenarios $\xi^{(r)}$. These cuts improve the approximation of the recourse function.

Feasibility test To verify the feasibility of the optimal solution x^* obtained by solving the problem (P^0) (5), we solve the dual of the second stage program (8) as stated below:

$$\mathscr{F}^{(r)}(x^*)\begin{cases} \max & \sigma^t[h(\xi^{(r)}) - T(\xi^{(r)})x^*], \\ s.t., & \sigma^t W \leq 0, \\ & \|\sigma\|_1 \leq 1. \end{cases} \tag{9}$$

According to Farkas's lemma, the set $\{x : Wx = h(\xi^{(r)}) - T(\xi^{(r)})x^*,\ x \geq 0\} \neq \varnothing$, if and only if, $\Big(\sigma^t \leq 0 \ \text{ implies } \ \sigma^t[h(\xi^{(r)}) - T(\xi^{(r)})x^*] \leq 0\Big)$. Therefore, the program $Q(\xi^{(r)}, x^*)$ is infeasible, if and only if, the set $P = \{\pi | \pi^t W \leq q^{(r)^t}\}$ has an extreme ray $\hat{\sigma}$ such that $\hat{\sigma}^{(r)^t}[h(\xi^{(r)}) - T(\xi^{(r)})]x^* > 0$. If a direction vector $\hat{\sigma}^{(r)}$ is an optimal solution to the program (9) then it verifies that the corresponding value of $\hat{\sigma}^{(r)^t}[h(\xi^{(r)}) - T(\xi^{(r)})x^*] \leq 0$, for all the scenarios r, then x^* is feasible.

The following proposition focuses on identifying scenarios in which a given solution fails to meet the requirements of a second-stage problem. To address this, the proposition introduces a feasibility cut that imposes additional constraints to ensure the solution's feasibility.

Proposition 3 (Feasibility cut). *If the inequality* $(\hat{\sigma}^{(r)})^t[h(\xi^{(r)}) - T(\xi^{(r)})x^*] > 0$, *is checked, for at least one* $r \in \{1, \ldots, R\}$, *where* $\hat{\sigma}^{(r)}$ *is a solution of (9), we have found a scenario* $\xi^{(r)}$ *for which* x^* *not yield a feasible second-stage problem. In this case, we create the feasibility cut:*

$$\sigma^{(r)^t}[h(\xi^{(r)}) - T(\xi^{(r)})x] \leq 0. \tag{10}$$

And $F^{(\ell)} = \bigcup_{i=1}^{\ell} \{\sigma_i^{(r)^t}[h(\xi^{(r)}) - T(\xi^{(r)})x] \leq 0\}$ *is the set of feasible cuts at step* ℓ, *with* $F^{(0)} = \varnothing$.

Optimality test To check if the solution $\hat{x}$ is optimal, we solve the following program:

$$\mathscr{O}_p^{(r)}(\hat{x})\begin{cases} \max & \pi^t[h(\xi^{(r)}) - T(\xi^{(r)})\hat{x}], \\ s.t., & \pi^t W \leq (q^{(r)})^t. \end{cases} \tag{11}$$

If $Q(\hat{x}) \leq \theta$, $\hat{x}$ is considered as an optimal solution. The following proposition presents a cut to enhance the value of the main problem when $Q(\hat{x}) > \theta$.

Proposition 4 (Optimality cut). *If $Q(\hat{x}) > \theta$, we introduce an optimality cut to the main problem which can be written as follows:*

$$\theta \geq \sum_{r=1}^{R} \mathscr{P}^{(r)}(\pi^{(r)^t}[h(\xi^{(r)} - T(\xi^{(r)}x]. \tag{12}$$

Such that, in the first step θ is set to $-\infty$. We define the set

$$O^{(\ell)} = \bigcup_{i=1}^{\ell} \{\theta \geq \sum_{r=1}^{R} \mathscr{P}^{(r)}(\pi_i^{(r)^t})[h(\xi^{(r)}) - T(\xi^{(r)})x]\},$$

as being the set of all optimatility cuts at step ℓ, such that $O^{(0)} = \varnothing$.

3 SUGGESTED METHODS

As it is mentioned above, we propose two exact solving approaches to determine all stochastic efficient solutions to SMOILPs (6): one is inspired by Sylva & Crema (2004) and the other is based on Lokman & Köksalan (2013), we call them Stochastic Sylva and Crema (SSC) and Stochastic Lokman and Kökslan (SLK).

3.1 Stochastic Sylva and Crema

We initiate the process by solving $(SP^{(0)})$ with $S^{(0)} = S$ and $\mathscr{X}_{Eff} = \varnothing$. We solve the deterministic relaxed problem (i.e., without the stochastic constraints) and let $x^{(0)}$ be an optimal integer solution. This is tested for feasibility (9) of $x^{(0)}$ for each scenario. If for at least one scenario it is not feasible, we add the feasibility cut proposition (3), re-optimize and retest feasibility until we find a feasible solution. Otherwise, if $x^{(0)}$ is feasible for all scenarios, we test its optimality. If $x^{(0)}$ is not optimal, we add the optimality cut proposition (4). Otherwise, $x^{(0)}$ is an efficient solution. As a result, $x^{(0)} \in \mathscr{X}_{Eff}$. The subsequent step involves reducing the feasible region $S^{(0)}$ to construct the new region $S^{(1)}$, which does not contain $x^{(0)}$ or any other solutions dominated by it. This reduction is achieved by introducing new binary variables $y_k^{(\ell)}$ that improve the optimal solution of the ℓ^{th} efficient solution and a set of constraints as shown in proposition (2). Therefore, at each iteration (ℓ), we solve $SP^{(\ell)}$ and a new efficient solution is found. The feasible region at each iteration is reduced in such a way that it does not contain the efficient solutions

already found. The method stops when the feasible region is empty (i.e., $SP^{(\ell)}$ is infeasible) and returns the set $\mathscr{X}_{Eff}$ of all stochastic efficient solutions of (SMOILP).

$$(SP^{(\ell)})\begin{cases} \max & \tilde{z} = \mathbb{E}(c^p(\xi))x + \varepsilon\sum_{k=1}^{p-1}\mathbb{E}(c^k(\xi))x, & \\ s.t., & x \in S, & \\ & \tilde{z}_k(x) \geq (\tilde{z}_k^i+1)y_k^i - M_k(1-y_k^i), & \forall i \in \{1,\ldots,\ell\}, \forall k \in \{1,\ldots,p-1\}, \\ & \sum_{k=1}^{p-1} y_k^i \geq 1, & \forall i \in \{1,\ldots,\ell\}, \\ & x \in F^{(i)}, & \forall i \in \{1,\ldots\ell\}, \\ & x \in O^{(i)}, & \forall i \in \{1,\ldots\ell\}, \\ & y_k^i \in \{0,1\}, & \forall i \in \{1,\ldots,\ell\}, \forall k \in \{1,\ldots,p-1\}, \end{cases} \tag{13}$$

where $\tilde{z}_k = \mathbb{E}(c^k(\xi))$ and $S^{(\ell)}$ is the feasible region of $SP^{(\ell)}$. The following theorem deals with the convergence and completeness of the proposed algorithms.

The general steps of the method are summarized in Algorithm 1, with clarifications in Appendix A.

Theorem 1. *Let $x_1,\ldots,x_\ell$ be the efficient solutions to (SP) and let x^* is the new efficient solution after adding all cuts to $S^{(\ell)}$, then x^* is an efficient solution to (SP). If ($SP^{(\ell)}$) is infeasible:*

1. *all efficient solutions are found,*
2. *the first proposed method terminates in finite iterations and generates all efficient solutions of the problem ($SP^{(\ell)}$).*

Proof. Let us suppose there exists $\dot{x} \in F(SP)$ such that $\mathbb{E}(cx^*) \leq \mathbb{E}(c\dot{x})$, with at least one strict inequality, where $F(SP)$ is feasible set of (SP) .

As x^* is an efficient integer stochastic solution to $F(SP) - \bigcup_{i=1}^{\ell}\{\mathscr{O}_p^{(i)} \cup \mathscr{F}^{(i)} \cup B_i\}$, where, $B_i = \{x \in \mathbb{Z}^n : \mathbb{E}(c^k(\xi))x \leq \mathbb{E}(c^k(\xi))x^i\}$, it follows that $\dot{x}$ cannot belong to feasible set $\{F(SP) - \bigcup_{i=1}^{\ell}\{\mathscr{O}_p^{(i)} \cup \mathscr{F}^{(i)} \cup B_i\}\}$, thus, $\dot{x} \in \bigcup_{i=1}^{\ell}\{\mathscr{O}_p^{(i)} \cup \mathscr{F}^{(i)} \cup B_i\}$, thus $\dot{x} \in \bigcup_{i=1}^{\ell}\{\mathscr{F}^{(i)}\}$ or $\dot{x} \in \bigcup_{i=1}^{\ell}\{B_i\}$ or $\dot{x} \in \bigcup_{i=1}^{\ell}\{\mathscr{O}_p^{(i)}\}$.

1^{st} case ($\dot{x} \in \bigcup_{i=1}^{\ell}\{B_i\}$): Let $\dot{x} \in \bigcup_{i=1}^{\ell}\{B_i\}$. Therefore, $\dot{x} \in B_i$ for at least one $i \in \{1\ldots\ell\}$. According to the definition of B_i, $\mathbb{E}(c\dot{x}) \leq \mathbb{E}(cx^i)$. As $\mathbb{E}(cx^*) \leq \mathbb{E}(c\dot{x}) \leq \mathbb{E}(cx^i)$, we have $x^* \in B_i$ contradicting that $x^* \in Eff(MOILP) - \bigcup_{i=1}^{\ell}\{\mathscr{O}_p^{(i)} \cup \mathscr{F}^{(i)} \cup B_i\}$.

2^{nd} case ($\dot{x} \in \bigcup_{i=1}^{\ell}\{\mathscr{F}^{(i)}\}$): Let $\dot{x} \in \bigcup_{i=1}^{\ell}\{\mathscr{F}^{(i)}\}$. So, $\dot{x} \in \bigcup_{i=1}^{\ell}\{(\sigma_{(r)})^t[h(\xi^{(r)}) - T(\xi^{(r)})x^{(i)}] \leq 0\}$. Therefore, it exists at least one direction vector $\hat{\sigma}_{(r)}$ such that $\hat{\sigma}^t_{(r)}[h(\xi^{(r)}) - T(\xi^{(r)})\dot{x}] > 0$, contradicting that $\dot{x} \in F(SP)$.

3^{rd} case ($\dot{x} \in \bigcup_{i=1}^{\ell}\{\mathscr{O}_p^{(i)}\}$): Let $\dot{x} \in \bigcup_{i=1}^{\ell}\{\mathscr{O}_p^{(i)}\}$, it implies that it exists a step $i \in \{1,\ldots,\ell\}$, such that

$$\theta \geq \sum_{r=1}^{R} \mathscr{P}^{(r)}(\dot{\pi}_{(r)})^t[h(\xi^{(r)}) - T(\xi^{(r)})\dot{x}],$$

therefore, $Q(\dot{x}) > \theta$, so $\dot{x}$ is not optimal in terms of stochasticity, contradicting that $\dot{x} \in F(SP)$. □

3.2 Stochastic Lokman and Kökslan

The idea of the second method is to reduce computational time by avoiding the use of binary variables. Instead, we generate sub-models for all possible combinations of y_k. By dividing the feasible region into sub-regions, we ensure that each region does not contain all the efficient solutions already found by computing all possible bounds as mentioned in (15). We solve the associated stochastic problem for each bound and add the best solution in terms of p^{th} objective to $\mathscr{X}_{Eff}$. Thus, at each step, a new efficient solution is found. The algorithm stops when all bounds are infeasible and provides the efficient set of the problem (14). To improve the performance of our algorithm, we used an approach proposed by Lokman & Köksalan (2013) for finding integer efficient solutions, which consist of subdividing the main model into sub-models. It is known that one constraint is sufficient to define the region that contains the non-dominated points with respect to the already available points. Based on this, it is concluded that the values of the right-hand side of each (p-1) objective for each sub-model can be determined more efficiently.

$$(P^{b^{s,\eta}}) \begin{cases} \max & z_p(x) + \varepsilon \sum_{k=1}^{p-1} z_k(x), \\ s.t., & \\ & z_k(x) \geq b_k^{s,\eta}, \qquad \forall k \in \{1,\ldots,p-1\}, \\ & x \in S. \end{cases} \tag{14}$$

Where, $b^{s,\eta} = (b_1^{s,\eta}, b_2^{s,\eta}, \ldots, b_{p-1}^{s,\eta})$ for $s \in \{0,\ldots,\eta\}$ defines a lower bound for each of the first $p-1$ criteria. If $(P^{b^{s,\eta}})$ is infeasible, then we assign $z^{(P^{b^{s,\eta}})} = (-M,\ -M,\ldots,-M)$.

Proposition 5. *(Lokman & Köksalan (2013))*

Let $S_\eta = \{z^\ell, 1 \leq \ell \leq \eta\}$ countain the avaible non dominated point and K the set of all possible combination of $s = (s_1, s_2, \ldots, s_{p-2})$ satisfying $s_k \in \{0,1,\ldots,\eta\}$ and $z_{s_j j} + 1 \leq z_{s_j k}$ for all indices $s_k, s_j \neq 0$ and $j < k$. Given the bounds vector $b^{s,\eta} = (b_1^{s,\eta}, b_2^{s,\eta}, \ldots, b_{p-1}^{s,\eta})$ and the set $S_\eta^s = \{z^\ell :$

Algorithm 1: Stochastic Sylva and Crema Algorithm for SMOILP

Result: The set of efficient solutions $\mathscr{X}_{Eff}$ for SP problem.

Input Data:

$\downarrow p, m, n, m_1, n_1$: the size of the problem.

$\downarrow R, \mathscr{P}^{(r)}$: the number of scenarios and corresponding probability r.

$\downarrow A_{(m\times n)}, b_{(m\times 1)}$: the deterministic constraints.

$\downarrow T_{(m_1\times n)}, h_{(m_1\times 1)}$: the stochastic constraints.

$\downarrow W_{(m_1\times n_1)}$: fixed recourse.

$\downarrow q_{(n_1\times 1)}$: penalties of constraint violations.

Output Data:

$\uparrow \mathscr{X}_{Eff}$: the set of stochastic efficient solutions.

$\uparrow Z_{Eff}$: the set of non dominated solutions.

Initialization $\mathscr{X}_{Eff} \leftarrow \varnothing$, $Z_{Eff} \leftarrow \varnothing$, $\theta \leftarrow -\infty$, $Exist \leftarrow True$, $S^0 \leftarrow S$, $Q \leftarrow 0, \ell \leftarrow 0$.

Compute $\tilde{c_k}$ (i.e., $\mathbb{E}(c^k(\xi))$ the expected value of $c^k(\xi)$.

Compute $\tilde{z}$ the main criterion vector.

Solve the deterministic relaxed problem $(SP^{(\ell)})$.

$\uparrow x^\ell, \uparrow \tilde{c} \times x^\ell = Pb_relaxed(\downarrow \tilde{z}, \downarrow A, \downarrow b)$.

if *the problem is not feasible* **then**
- **Exist** $\leftarrow$ **False.**

else
- **while** *Exist* **do**
  - **while** $r \leq R$ **do**
    - feasibility $\leftarrow$ False
    - **while** *feasibility=False* **do**
      - $\sigma^{(r)} \leftarrow$ Solve $\mathscr{F}^{(r)}(x^\ell)$
      - **if** $\sigma^{(r)}[h(\xi^{(r)}) - T(\xi^{(r)})x^\ell] > 0$ **then**
        - $S^\ell \leftarrow S^\ell \cup \{feasibility \;\; cut\}$
        - $x^\ell \leftarrow Solve(SP^\ell)$
        - $r \leftarrow 0$
      - feasibility$\leftarrow$True
    - **end**
    - $r \leftarrow r+1$
  - **end**
- **end**
- **for** $r \leftarrow 1$ **to** R **do**
  - $Q(x^\ell, \xi^{(r)}) \leftarrow$ Solve $\mathscr{O}_p^{(r)}(x^\ell)$
  - $Q \leftarrow Q + \mathscr{P}^{(r)} \times Q(x^\ell, \xi^{(r)})$
- **end**
- **if** $\theta < Q$ **then**
  - $S^\ell \leftarrow S^\ell \cup \{$Optimality cut$\}$
  - $x^{\ell+1} \leftarrow$ Solve (SP^ℓ).
- **else**
  - $x^{\ell+1} \leftarrow x^\ell$
- **end**
- $\mathscr{X}_{Eff} \leftarrow \mathscr{X}_{Eff} \cup \{x^{\ell+1}\}$
- $Z_{Eff} \leftarrow Z_{Eff} \cup \{\tilde{c} * x^{\ell+1}\}$
- **if** (SP^ℓ) *infeasible* **then**
  - **Exist** $\leftarrow$ **False.**
- **else**
  - $x^\ell \leftarrow x^{\ell+1}$
- **end**

end

$z_{\ell k} \geq b_i^{s,\eta}, k \in \{1,\ldots,p-2\}, \quad z^\ell \in S_\eta\}$, *let* s^* *be the chosen index* $z_p^{(P^{b^{s^*,\eta}})} = \max_{s\in K}\{z_p^{(P^{b^{s,\eta}})}\}$, *where,*

$$b_k^{s,\eta} = \begin{cases} -M, & \text{if } s_k = 0, \\ z_{s_k k}+1, \forall k \in \{1,\ldots,p-2\}, & \text{otherwise}, \end{cases} \qquad b_{p-1}^{s,\eta} = \begin{cases} -M, & \text{if } S_\eta^s = \varnothing, \\ \max\limits_{z^\ell \in S_\eta^s} z_{\ell(p-1)}+1, & \text{otherwise}, \end{cases} \tag{15}$$

where, S_η^s *contains the available non-dominated points, and* $b^{s,\eta} = (b_1^{s,\eta}, b_2^{s,\eta}, \ldots, b_{p-1}^{s,\eta})$ *denotes the corresponding bound vector for each sub-model. If* $z_p^{(P^{b^{s^*,\eta}})} = -M$, *then the sets* $S_\eta = \{z^\ell : 1 \leq \ell \leq \eta\}$ *contains all non-dominated points of the original problem* (*SMOILP*). *Otherwise* $z^{\eta+1} = z^{(P^{b^{s^*,\eta}})}$.

Description of the suggested method

The proposed method consists of finding all efficient solutions for a SMOILP with fixed recourse, see (*PS*), by going through several steps and adding a series of cuts to feasible region $SI^{(s)}$ of the problem $SIP^{(s)}$ where $SI^{(0)} = S$.

$$(SIP^{(s)}) \begin{cases} \max\ \tilde{z} = \quad \mathbb{E}(z_p(x)) + \varepsilon \sum\limits_{k=1}^{p-1} \mathbb{E}(z_k(x)), \\ s.t., \\ \qquad x \in S, \\ \qquad \tilde{z}_k(x) \geq b_k^{s,\eta} \quad \forall k \in \{1,..,p-1\}, \forall s \in \{0,..,\eta\}, \\ \qquad x \in F^{(s)}, \qquad \forall s \in \{0,..,\eta\}, \\ \qquad x \in O^{(s)}, \qquad \forall s \in \{0,..,\eta\}. \end{cases} \tag{16}$$

Where,

$$O^{(s)} = \bigcup_{i=1}^{s} \{\theta \geq \sum_{r=1}^{R} \mathscr{P}^{(r)} (\pi_i^{(r)^t})[h(\xi^{(r)}) - T(\xi^{(r)})x]\},$$

and

$$F^{(s)} = \bigcup_{i=1}^{s} \{\sigma_i^{(r)^t}[h(\xi^{(r)}) - T(\xi^{(r)})x] \leq 0\},$$

with $F^{(0)} = \varnothing$, $O^{(0)} = \varnothing$.

In each step, we optimize the total addition of the *pth* objective function and the sum of other objectives scaled by a small positive constant. We solve this single deterministic relaxed program using the L-Shaped method, i.e., we begin by calculating the expected value of the objectives of each scenario. Next, we solve the deterministic problem and let x^0 be the optimal solution. Note that we test the feasibility of x^0 for each scenario. If, for at least one scenario, it is not feasible, we add the feasibility cut (3) and reoptimize until we find a feasible solution. Otherwise, if x^0 is feasible for all the possible scenarios, we test it for optimality. If x^0 is not optimal, we add the optimality cut (4). Otherwise, x^0 is an efficient solution. Next, we start by separating the

feasible region into sub-regions, such that each region doesn't contain all the efficient solutions already found by calculating all possible bounds as mentioned in (15). We solve the associated deterministic problem (16) for each bound and add the best solution in terms of the p^{th} objective to Z_{Eff}. Therefore, in each step, a new non-dominated solution is found. When all bounds are infeasible, the algorithm stops and provides the entire non-dominated set of problem (14). The general steps of the method are summarized in Algorithm 2, while additional details are provided in Appendix B.

This method introduces key improvements to avoid extra model-solving by significantly reducing the number of bounds to solve through two specific conditions: (i) Bound Overlap: if b^{s_1,η_1} lies between $b^{s,\eta}$ and its relative solution $z^{(p^{b^{s,\eta}})}$, it produces the same criterion vector as $b^{s,\eta}$. (ii) Infeasible Bounds: if $b^{s,\eta}$ (stored in the *Infeasible* list) is infeasible and the $b^{s_1,\eta_1} > b^{s,\eta}$, then b^{s_1,η_1} is also infeasible.

4 NUMERICAL ILLUSTRATION

As an illustration of our second proposed method, we show how it computes all non-dominated points of a SMOILP on an instance with three objective functions and two scenarios. We present the following example:

$$(P_1)\begin{cases} \max & 3x_1+x_2, \\ \max & 7x_1-x_2, \\ \max & 2x_1-3x_2, \\ s.t., & \\ & 4x_1-2x_2 \leq 8, \\ & x_1+x_2 \leq 5, \\ & x_1+2x_2 \leq 3, \\ & -2x_1+x_2 \leq 5, \\ & x_1,x_2 \in \mathbb{N}. \end{cases} \qquad (P_2)\begin{cases} \max & -9x_1+x_2, \\ \max & 3x_1-3x_2, \\ \max & 2x_1+x_2, \\ s.t., & \\ & 4x_1-2x_2 \leq 8, \\ & x_1+x_2 \leq 5, \\ & x_1 \leq 6, \\ & 3x_1+4x_2 \leq 1, \\ & x_1,x_2 \in \mathbb{N}. \end{cases}$$

Two scenarios affect stochastic objectives and constraints. Assume that the realizations of $\xi^{(r)}$ are equiprobable, with:

$$\mathscr{P}(\xi^{(1)}) = \frac{1}{2} \quad \text{and} \quad \mathscr{P}(\xi^{(2)}) = \frac{1}{2}.$$

Penalties for Constraint Violations The penalties for the two scenarios are defined as follows:

- For scenario $\xi^{(1)}$:
$$q(\xi^{(1)}) = (1,0,6,2),$$
- For scenario $\xi^{(2)}$:
$$q(\xi^{(2)}) = (5,3,2,1).$$

Algorithm 2: Stochastic Lokman and Kökslan Algorithm for SMOILP

Result: The set of non dominated solutions Z_{Eff} for SMOILP problem.

Input Data:

$\downarrow p, m, n, m_1, n_1$: the size of the problem.

$\downarrow R, \mathscr{P}^{(r)}$: the number of scenarios and probability of each scenario r.

$\downarrow A_{(m\times n)}, b_{(m\times 1)}$: the deterministic constraints.

$\downarrow T_{(m_1\times n)}, h_{(m_1\times 1)}$: the stochastic constraints.

$\downarrow W_{(m_1\times n_1)}, q_{(n_1\times 1)}$: fixed recourse and penalties of constraint violations.

Output Data:

$\uparrow \mathscr{X}_{Eff}$: the set of efficient solutions. $\uparrow Z_{Eff}$: the set of non-dominated solutions.

Initialization:

$\mathscr{X}_{Eff} \leftarrow \varnothing, \quad Z_{Eff} \leftarrow \varnothing, \quad \theta \leftarrow -\infty, \quad Exist \leftarrow True, \quad S \leftarrow SI^0,$

$b^{0,0} \leftarrow (-M, -M, ..., -M), \quad Q \leftarrow 0, \quad Infeasible \leftarrow \varnothing, \quad \eta \leftarrow 0.$

Compute $\tilde{c_k}$ (i.e., $\mathbb{E}(c^k(\xi))$ the expected value of $c^k(\xi)$.

Compute $\tilde{z}$ the main criterion vector.

Solve the deterministic relaxed problem $(SIP^{(s)})$.

$\uparrow x^s, \uparrow \tilde{z}^s = Pb_relaxed(\downarrow \tilde{z}, \downarrow A, \downarrow b)$.

while *Exist* **do**

- Calculate the bounds $b^{s,\eta}$.
- $S_\eta \leftarrow \varnothing$.
- **for** $s \leftarrow 0$ **to** η **do**
  - **if** *(b^s is not solved yet $||b^s \notin$ Infeasible)* **then**
    - **for** $k \leftarrow 1$ **to** *p-1* **do**
      - Add cut $\tilde{z}_k(x) \geq b_k^s$ to $(SIP^{(s)})$.
    - $x^s, \tilde{z}^s \leftarrow$ solve bound $(SIP^{(s)})$.
    - **if** $(SIP^{(s)})$ *is infeasible* **then**
      - $Infeasible \leftarrow Infeasible \cup b^{s,\eta}$
      - $z^{s,\eta} \leftarrow -\infty$ Continu;
    - **while** $r \leq R$ **do**
      - feasibility $\leftarrow$ False.
      - **while** *feasiblility=False* **do**
        - $\sigma^{(r)} \leftarrow$ Solve $\mathscr{F}^{(r)}(x^s)$.
        - **if** $\sigma^{(r)}[h(\xi^{(r)}) - T(\xi^{(r)})x^s] > 0$ **then**
          - $SI^s \leftarrow SI^{s-1} \cup \{$feasibility cut $\}$.
          - $x^s \leftarrow Solve(SIP^s)$
          - $r \leftarrow 0$
        - **else**
          - feasiblility $\leftarrow$ True
      - $r \leftarrow r+1$
    - **for** $r \leftarrow 1$ **to** R **do**
      - $Q(x^s, \xi^{(r)}) \leftarrow$ Solve $\mathscr{O}_p^{(r)}(x^s)$.
      - $Q \leftarrow Q + \mathscr{P}^{(r)} \times Q(x^s, \xi^{(r)})$.
    - **if** $teta < Q$ **then**
      - $SI^s \leftarrow SI^s \cup \{optimality \quad cut\}$.
      - $x^s \leftarrow Solve(SIP^{(s)})$.
    - $z^{s,\eta} \leftarrow \tilde{c} \times x^s$.
  - **else**
    - $z^{s,\eta}$ relative to b_i^s.
  - $S_\eta \leftarrow S_\eta \cup z^{s,\eta}$.
- $\tilde{z} \longleftarrow \max_{z^{s,\eta} \in S_\eta} (z^{s,\eta}(p))$.
- **if** $\tilde{z} = -\infty$ **then**
  - **Exist** $\leftarrow$ **False.**
- **else**
  - $\mathscr{X}_{Eff} \leftarrow \mathscr{X}_{Eff} \cup \{\tilde{x}\}. \quad Z_{Eff} \leftarrow Z_{Eff} \cup \{\tilde{z}\}. \quad \eta \leftarrow \eta + 1.$

A fixed recourse matrix W is given by:

$$W = \begin{pmatrix} -2 & -1 & 2 & 1 \\ 3 & 2 & -5 & -6 \end{pmatrix}.$$

Given the probabilities of the two scenarios, the expected values of the objectives are:

$$\mathbb{E}[c^1(\xi)] = \mathscr{P}(\xi^{(1)})c^1(\xi^{(1)}) + \mathscr{P}(\xi^{(2)})c^1(\xi^{(2)}) = -3x_1 + x_2,$$
$$\mathbb{E}[c^2(\xi)] = \mathscr{P}(\xi^{(1)})c^2(\xi^{(1)}) + \mathscr{P}(\xi^{(2)})c^2(\xi^{(2)}) = 5x_1 - 2x_2,$$
$$\mathbb{E}[c^3(\xi)] = \mathscr{P}(\xi^{(1)})c^3(\xi^{(1)}) + \mathscr{P}(\xi^{(2)})c^3(\xi^{(2)}) = 2x_1 - x_2.$$

We start by solving the relaxed program with a single criterion:

$$(P_0)\begin{cases} \max & \mathbb{E}[c^3(\xi)] + \varepsilon\mathbb{E}[c^1(\xi)] + \varepsilon\mathbb{E}[c^2(\xi)], \\ s.t., & \\ & 4x_1 - 2x_2 + x_3 = 8, \\ & x_1 + x_2 + x_4 = 5, \\ & x_1, x_2, x_3, x_4 \geq 0, \end{cases}$$

where x_3 and x_4 are slack variables, $Z_{Eff} = \varnothing$ and $\mathscr{X}_{Eff} = S = \varnothing$ represents, respectively, the sets of non-dominated stochastic integer solutions and the set of efficient stochastic integer solutions.

First Iteration In the first iteration, we found the following solution: $z = (-6, 10, 4)$, which defined the set $S^0 = \{(-6, 10, 4)\}$. The corresponding decision variable value is $x = (2, 0)$. This solution is identified as the optimal solution to the first-stage problem, denoted as (P_0).

Feasibility test To test the feasibility of the solution $x = (2, 0)$, we solve the problem (9) independently for each scenario.

$$\mathscr{F}_s^{(1)}(x)\begin{cases} \max & -\sigma_1^{(1)} + 3\sigma_2^{(1)}, \\ s.t., & \\ & -2\sigma_1^{(1)} + 3\sigma_2^{(1)} \leq 0, \\ & -1\sigma_1^{(1)} + 2\sigma_2^{(1)} \leq 0, \\ & 2\sigma_1^{(1)} - 5\sigma_2^{(1)} \leq 0, \\ & -5\sigma_1^{(1)} - 6\sigma_2^{(1)} \leq 0, \\ & \sigma_1^{(1)} + \sigma_2^{(1)} \leq 1. \end{cases} \qquad \mathscr{F}_s^{(2)}(x)\begin{cases} \max & 6\sigma_1^{(2)} - 7\sigma_2^{(2)}, \\ s.t., & \\ & -2\sigma_1^{(2)} + 3\sigma_2^{(2)} \leq 0, \\ & -1\sigma_1^{(2)} + 2\sigma_2^{(2)} \leq 0, \\ & 2\sigma_1^{(2)} - 5\sigma_2^{(2)} \leq 0, \\ & -5\sigma_1^{(2)} - 6\sigma_2^{(2)} \leq 0, \\ & \sigma_1^{(2)} + \sigma_2^{(2)} \leq 1. \end{cases}$$

The obtained solutions are: for $\mathscr{F}_s^1(x)$ is $\sigma^{(1)^t} = (0.6667, 0.3333)$, and for $\mathscr{F}_s^2(x)$ is $\sigma^{(2)^t} = (0.7143, 0.2857)$. Furthermore, we have

$$\sigma^{(1)^t}[h(\xi^{(1)}) - T(\xi^{(1)})] = 1.6667 > 0 \quad \text{and} \quad \sigma^{(2)^t}[h(\xi^{(2)}) - T(\xi^{(2)})] = 2.2857 > 0.$$

Hence, we add the feasibility cut either by using $\sigma^{(1)}$ or $\sigma^{(2)}$. This indicates that the solution is not feasible for both scenarios. To adress this, we introduce a feasibility cut in the form of (10). We find $\frac{5}{3}x_2 \geq \frac{11}{3}$, and we add this cut to $\mathscr{F}^{(1)}$ as $\frac{5}{3}x_2 \geq \frac{11}{3}$, as well as to (P_0) to resolve. Upon solving the problem with this new cut, we obtain the solution $\hat{x} = (2,3)$. To test the feasibility of this new solution, $\hat{x} = (2,3)$, we solve the problem (9) again, corresponding to each scenario.

$$\mathscr{F}_s^{(1)}(\hat{x}) \begin{cases} \max & -5\sigma_1^{(1)} + 6\sigma_2^{(1)}, \\ s.t., & \\ & -2\sigma_1^{(1)} + 3\sigma_2^{(1)} \leq 0, \\ & -1\sigma_1^{(1)} + 2\sigma_2^{(1)} \leq 0, \\ & 2\sigma_1^{(1)} - 5\sigma_2^{(1)} \leq 0, \\ & -5\sigma_1^{(1)} - 6\sigma_2^{(1)} \leq 0, \\ & \sigma_1^{(1)} + \sigma_2^{(1)} \leq 1, \end{cases} \qquad \mathscr{F}_s^{(2)}(\hat{x}) \begin{cases} \max & 4\sigma_1^{(2)} - 17\sigma_2^{(2)}, \\ s.t., & \\ & -2\sigma_1^{(2)} + 3\sigma_2^{(2)} \leq 0, \\ & -1\sigma_1^{(2)} + 2\sigma_2^{(2)} \leq 0, \\ & 2\sigma_1^{(2)} - 5\sigma_2^{(2)} \leq 0, \\ & -5\sigma_1^{(2)} - 6\sigma_2^{(2)} \leq 0, \\ & \sigma_1^{(2)} + \sigma_2^{(2)} \leq 1. \end{cases}$$

The obtained solution of $\mathscr{F}_s^{(1)}(\hat{x})$ is $\sigma^{(1)^t} = (0,0)$, and that of $\mathscr{F}_s^{(2)}(\hat{x})$ is $\sigma^{(2)^t} = (0,0)$. We observe that $\sigma^{(1)} \leq 0$ and $\sigma^{(2)} \leq 0$. This indicate that $\hat{x} = (2,3)$ is feasible for both scenarios $\xi^{(1)}$ and $\xi^{(2)}$.

Optimality cut

To test the optimality of the solution $\hat{x} = (2,3)$, we solve the problem (11) corresponding to each scenario.

$$\mathscr{O}_p^{(1)}(\hat{x}) \begin{cases} \max & -5\pi_1^{(1)} + 6\pi_2^{(1)}, \\ s.t., & \\ & -2\pi_1^{(1)} + 3\pi_2^{(1)} \leq 1, \\ & -1\pi_1^{(1)} + 2\pi_2^{(1)} \leq 0, \\ & 2\pi_1^{(1)} - 5\pi_2^{(1)} \leq 6, \\ & -5\pi_1^{(1)} - 6\pi_2^{(1)} \leq 2, \end{cases} \qquad \mathscr{O}_p^{(2)}(\hat{x}) \begin{cases} \max & 4\pi_1^{(2)} - 17\pi_2^{(2)}, \\ s.t., & \\ & -2\pi_1^{(2)} + 3\pi_1^{(2)} \leq 5, \\ & -1\pi_1^{(2)} + 2\pi_1^{(2)} \leq 3, \\ & 2\pi_1^{(2)} - 5\pi_1^{(2)} \leq 2, \\ & -5\pi_1^{(2)} - 6\pi_1^{(2)} \leq 1. \end{cases}$$

The obtained solution of $\mathscr{O}_p^{(1)}(\hat{x})$ is $\pi^{(1)^t} = (-1,-0.5)$, and of $\mathscr{O}_p^{(2)}(\hat{x})$ is $\pi^{(2)^t} = (1,0)$.
$Q(x,\xi^{(1)}) = \pi_1^t[h(\xi^{(1)}) - T(\xi^{(1)})] = 2$ and $Q(x,\xi^{(2)}) = \pi_2^t[h(\xi^{(2)}) - T(\xi^{(2)})] = 4$.

$$Q(x) = \mathscr{P}(\xi^{(1)}) \times Q(x,\xi^{(1)}) + \mathscr{P}(\xi^{(2)}) \times Q(x,\xi^{(2)}) = \frac{1}{2} \times 2 + \frac{1}{2} \times 4 = 3.$$

Since, $\theta = -\infty < Q$, thus, we introduce the optimality cut to $\mathscr{O}_p^{(1)}$ of the form (12), we have $\theta \leq \frac{1}{4} - \frac{1}{2}x_1 + \frac{5}{4}x_2$. We introduce the new variable θ along with this new constraint, and we re-optimize the problem. The new solution is $\bar{x} = (2,3,3)$, where $\bar{x}(3) = \theta$. Thus, $x^1 = (2,3)$ is an efficient integer solution. The criterion vector for this solution is $z^1 = (-3,4,1)$. Hence, The sets $\mathscr{X}_{Eff}$ and Z_{Eff} are initialized as follows: $\mathscr{X}_{Eff} = \{(2,3)\}$, $Z_{Eff} = \{(-3,4,1)\}$.

Next, we identify the bounds using all possible combinations of s defined in 15, such that η is the number of non-dominated solutions, where $s \in 0,\ldots,\eta$. For $\eta = 1$ and $s \in \{0,1\}$, we get

$$b_1^{(s,1)} = \begin{cases} b_1^{(0,1)} = -M, & s = 0, \\ b_1^{(1,1)} = z_{s_1 1} + 1 = -2, & j = 1, \end{cases} \text{ and } b_2^{(s,1)} = \begin{cases} b_2^{(0,1)} = \max\limits_{z^t \in S_1^0} = 4 + 1 = 5, & S_1^0 \neq \varnothing, \\ b_2^{(1,1)} = -M, & S_1^1 = \varnothing. \end{cases}$$

After finding the bounds for all k, we add these bounds as cuts to the initial problem separately. Before solving the bounds, we check if each bound has already been solved. Two cases are possible:

1. The bound is infeasible: we compare it with the infeasible bounds already found. If the bound is greater than at least one of them, then the bound is infeasible.

2. The bound has already been resolved: we compare it with the previously resolved bounds that have already been found. For each bound, test feasibility and optimality. We choose the best in terms of $z(p)$, which means $\max\limits_{s \in K} z(p)^{b^{s,\eta}}$.

In summary, the efficient set is $\mathscr{X}_{Eff} = \{(2,3),(1,3),(1,4),(0,4),(0,5)\}$, with the corresponding non-dominated set is $Z_{Eff} = \{(-3,4,1),(0,-1,-1),(1,-3,-2),(4-8,-4),$ $(5,-10,-5)\}$. Table 1 presents the obtained results at each iteration of the numerical example.

5 COMPUTATIONAL STUDY

To further evaluate the performance of the proposed algorithms, we compare them against the state-of-the-art algorithm proposed by Amrouche & Moulaï (2012). Our algorithms, as well as the Amrouche and Moulaï algorithm, have been implemented in MATLAB and IBM ILOG CPLEX 12.6 as the integer programming solver. All computational experiments have been carried out on a HP Intel(R) CORE(TM) i5-4210U CPU @1.70GHz 2.40 GHz RAM 4Go. We conducted an empirical study to assess and compare the computational efficiency.

Three sets of experiments are reported below: randomly generated instances of various configurations (i.e., sizes and number of scenarios), and two state-of-the-art structured instances. In all of them, a time limit of 1200 seconds is imposed on each individual run, its role differing between the two kinds of instances. On the benchmark instances, a configuration is a single instance: a run reaching the limit is interrupted. On the random instances, a configuration is a sample of 10 instances and is processed in two steps. The 10 runs are first executed under the limit and the configuration is marked unsuccessfull for an algorithm if none of them terminates. Otherwise, if at least one run terminates under the limit, all configurations are executed again without limit and the statistics of Table 2 are computed from this second runs, which is why some averages exceed 1200 seconds. The first step thus decides whether a configuration can be solved by an algorithm within the limit, while the second one measures its runtime distribution over the whole sample.

Table 1 – Table containing all the steps and results of the numerical example.

η	s	$b^{s,\eta}$	$Z^{b^{s,\eta}}$	Solve $P^{b^{s,\eta}}$	$x^{b^{s,\eta}}$	SLK feasible	SLK optimal	Z^*
0	0	$(-M,-M)$	(-6,10,4)	$\surd$	(2,0)	(2,3)	(2,3,3)	(-3,4,1)
1	0	$(-M,5)$	infeasible	identical to $P^{b^{0,0}}$	(2,0)	-	-	(0,-1,-1)
	1	$(-2,-M)$	(-2,3,1)	$\surd$	(1,1)	(1,3)	(1,3,3.5)	-
2	0	$(-M,5)$	infeasible	identical to $P^{b^{0,0}}$	(2,0)	-	-	(1,-3,-2)
	1	(-2,0)	infeasible	identical to $P^{b^{0,0}}$	(2,0)	-	-	-
	2	$(1,-M)$	(1,-2,-1)	$\surd$	(0,1)	(1,4)	(1,4,4.75)	-
3	0	$(-M,5)$	infeasible	identical to $P^{b^{0,0}}$	(2,0)	-	-	(4,-8,-4)
	1	(-2,0)	infeasible	identical to $P^{b^{0,0}}$	(2,0)	-	-	-
	2	(1,-2)	infeasible	identical to $P^{b^{2,2}}$	(0,1)	-	-	-
	3	$(2,-M)$	(2,-4,-2)	$\surd$	(0,2)	(0,3)	(0,4,5.25)	-
4	0	$(-M,5)$	infeasible	identical to $P^{b^{0,0}}$	(2,0)	-	-	(5,-10,-5)
	1	(-2,0)	infeasible	identical to $P^{b^{0,0}}$	(2,0)	-	-	-
	2	(1,-2)	infeasible	identical to $P^{b^{2,2}}$	(0,1)	-	-	-
	3	(2,-7)	infeasible	identical to $P^{b^{3,3}}$	(0,1)	-	-	-
	4	$(5,-M)$	(5,-10,-5)	$\surd$	(0,5)	(0,5)	(0,5,7)	-
5	0	$(-M,5)$	infeasible	identical to $P^{b^{0,0}}$	(2,0)	-	-	All found
	1	(-2,0)	infeasible	identical to $P^{b^{0,0}}$	(2,0)	-	-	-
	2	(1,-2)	infeasible	identical to $P^{b^{2,2}}$	(0,1)	-	-	-
	3	(2,-7)	infeasible	identical to $P^{b^{3,3}}$	(0,1)	-	-	-
	4	(5,-9)	infeasible	identical to $P^{b^{4,4}}$	(0,5)	-	-	-
	5	$(6,-M)$	infeasible	$\surd$	(0,5)	-	-	-

The symbol '-' indicates that this linear program has already been solved.

5.1 Evaluation on randomly generated instances

In this section, we present the results of computational experiments conducted to evaluate the performance of the proposed algorithms (i.e., SLK and SSC) and to compare them with the state-of-the-art algorithm proposed by Amrouche & Moulaï (2012), using random SMOILP instances generated according to the following strategy.

5.1.1 Random Instance Generation

The deterministic and stochastic components of each instance were randomly generated as follows.

- The first-stage objective coefficients $c = (c^1, c^2, \ldots, c^p)$, where $z_k(x) = c^k x$ for all $k \in \{1, \ldots, p\}$, are drawn from a discrete uniform distribution over $[-100, 100]$.
- The constraint matrix A and the right-hand side vector b are drawn independently from discrete uniform distributions over $[0, 30]$ and $[90, 130]$, respectively. Since all coefficients of A are non-negative, the feasible region $D = \{x \in \mathbb{N}^n : Ax \leq b\}$ is bounded.
- The technology matrix $T(\xi)$ and the stochastic right-hand side vector $h(\xi)$ are drawn independently from discrete uniform distributions over $[0, 30]$ and $[90, 130]$, respectively. The non-negative support of $T(\xi)$ ensures that the induced feasibility set $\{x \mid h(\xi) - T(\xi)x \in \operatorname{pos} W\}$ remains non-trivial Kall & Wallace (1994), while the support $[90, 130]$ for $h(\xi)$ is chosen consistently with the range of b, so that the stochastic right-hand side remains in a realistic range relative to the first-stage constraints Birge & Louveaux (1997).
- The recourse matrix W and the second-stage cost vector q are drawn from discrete uniform distributions over $[5, 10]$ and $[0, 10]$, respectively. The matrix W is fixed across all scenarios, yielding a fixed recourse structure Wets (1974); this means that only the right-hand side $h(\xi) - T(\xi)x$ varies with the scenario ξ, while the recourse technology remains constant. Since all entries of W are drawn from a strictly positive support and W has full row rank, the three definitions of the feasibility region coincide, ensuring that the expected recourse function $\mathscr{Q}(x)$ is well-defined and finite on the feasibility region Wets (1974). Moreover, since $q \geq 0$ and $y \geq 0$, the recourse value satisfies $Q(x, \xi) \geq 0 > -\infty$ for all realizations of ξ Wets (1974).

Accordingly, we generated 10 instances of 18 different sizes (i.e., 5 to 250 variables and 3 to 125 constraints) and 10 instances with 30 variables and 13 different number of scenarios (i.e., 2 to 4000 scenarios).

5.1.2 Experimental results

In this experimentation study, we assessed the performance of the suggested methods. First, we aim to compare the efficiency of our algorithms, against the state-of-the-art method, in solving a SMOILP, using different instance sizes. On the other hand, the second part of this study focused on varying the number of scenarios to compare its effect on the CPU time required for each algorithm. The obtained results are resumed in Table 2, where

- N is the average number of solutions.

- For each method, we estimate the maximum (max), minimum (min), mean (Average), and standard deviation (std) runtimes for every instance.
- For SLK method, we included: the number of bounds solved (RIS), the number of infeasible bounds (INF), and the number of bounds already solved that we avoid their resolution for a second time (RES).
- As a performance measure, we use average runtime.

As it is mentioned above, for each parameter (i.e., row), we generated 10 instances to ensure the consistency of our evaluation.

5.1.3 Results Discussion

One can observe in Table 2 that our proposed algorithms are significantly faster than Amrouche and Moulaï's method Amrouche & Moulaï (2012). One can also notice when comparing SSC method and SLK method, the latter proved to be more efficient in terms of execution time. This is due to visiting more solutions and adding more binary variables and constraints, making the SSC method slower at each iteration. In other words, more sub-problems are solved, but each one keeps the same number of constraints and binary variables while yielding the same efficient set. For larger instances this observation stands. For example, when solving an instance with 50 variables, Amrouche and Moulaï's method is the largest solved instance where it required around 1200 seconds, against, 12.05 seconds required for SLK method and 92.43 seconds for SSC method. Furthermore, the suggested methods completed in a significantly shorter duration, i.e., 100 variables for SSC method and at least 250 for SLK method. Hence, we can conclude that the competing methods' order by efficiency is : SLK method, SSC method, and then Amrouche and Moulaï's method.

Table 2 – Performance comparison of the proposed algorithms against state-of-the-art Amrouche and Moulaï's algorithm on random instances.

				Stochastic Lokman and Kökslan							Stochastic Sylva and Crema				Amrouche and Moulaï			
Instances				Bound			Time in seconds				Time in seconds				Time in seconds			
K	R	$n \times m$	N	RIS	INF	RES	min	max	average	std	min	max	average	std	min	max	average	std
3	2	5×3	13.45	30.64	70.27	66.55	0.03	1.89	0.71	0.64	**0.02**	2.10	0.65	0.68	**0.02**	**1.36**	**0.53**	0.46
		10×5	27.27	66.09	161.82	519.09	0.47	**7.67**	**2.15**	2.17	**0.26**	26.94	4.42	8.00	0.37	16.28	4.56	4.73
		15×5	53	108,00	184.50	1524.00	**3.00**	**6.29**	**4.65**	2.33	6.14	18.80	12.47	8.95	16.04	32.70	24.37	11.78
		15×7	33.09	80.09	453.27	579.00	0.41	**8.30**	**3.11**	3.09	**0.15**	29.04	7.38	10.01	2.08	57.21	15.86	16.82
		20×7	32.50	67.00	256.50	257.50	**2.28**	**3.91**	**3.10**	1.15	4.00	5.66	4.83	1.17	128.20	230.98	179.59	72.68
		20×10	40.82	122.36	268.09	1211.36	1.47	**21.88**	**6.14**	5.76	**0.78**	96.27	22.99	47.64	3.11	67.84	25.08	21.25
		30×15	50	96	416.71	782.14	**2.71**	**29.60**	**9.41**	9.96	5.22	104.55	24.77	36.13	47.40	261.09	130.90	84.38
		40×20	48.20	141.36	360.91	1026.00	5.26	**31.01**	**12.18**	6.96	**3.26**	412.21	72.17	125.71	78.85	495.14	242.60	149.61
		50×25	48.00	88.00	418.60	891.60	1.93	**27.08**	**12.05**	12.66	**1.64**	368.52	92.43	158.05	687.26	2888.94	1284.63	1072.79
		60×30	41	115.55	340.09	718.09	**3.74**	**57.90**	**21.67**	16.11	4.74	239.95	76.98	91.47	-	-	-	-
		70×35	119.80	313.60	1622.00	5836.20	**43.44**	**137.90**	**84.24**	45.65	289.65	1643.19	769.32	525.40	-	-	-	-
		80×40	54.45	138.73	739.55	1563.91	14.38	**125.02**	**43.35**	30.09	**6.40**	2494.19	366.15	725.08	-	-	-	-
		90×45	63.18	202.27	713.18	2126.27	8.43	**215.34**	**67.25**	56.12	**6.87**	3557.10	641.62	1044.83	-	-	-	-
		100×50	73.25	208.22	712.78	2486.56	**38.05**	**284.07**	**98.11**	77.58	167.03	4625.77	1480.75	1529.39	-	-	-	-
		150×75	77.83	222.67	926.17	2305.17	**107.79**	**294.69**	**205.86**	81.29	-	-	-	-	-	-	-	-
		180×90	117.50	280.33	1768.33	7322.83	**103.29**	**824.07**	**481.95**	302.81	-	-	-	-	-	-	-	-
		200×100	74.17	272.33	467.33	3934.33	**156.28**	**496.46**	**377.19**	132.79	-	-	-	-	-	-	-	-
		250×125	148.50	299.00	2677.50	8907.67	**436.17**	**3560.51**	**1440.39**	1122.88	-	-	-	-	-	-	-	-
3	2	30×15	81.17	202.67	1196.83	2837.33	**5.61**	**29.46**	**15.87**	9.85	17.87	1170.42	262.07	455.42	46.35	454.10	300.73	157.18
	5	30×15	22.50	156.17	148.67	267.67	1.14	**36.11**	**12.91**	14.72	**0.50**	63.62	17.16	24.23	28.60	407.13	192.56	141.30
	10	30×15	55.33	123.67	673.83	2219.50	0.86	**34.21**	**13.48**	14.13	**0.46**	3788.69	699.74	1516.93	34.15	1560.12	507.26	628.57
	20	30×15	57.50	143.67	552.67	2025.67	**4.41**	**81.27**	**21.40**	29.52	4.65	822.41	150.87	329.22	204.23	2449.84	832.75	840.92
	50	30×15	41.83	88.67	442.50	680.50	1.29	**24.63**	**14.13**	9.75	**0.67**	70.73	29.38	26.25	12.38	2332.52	1342.92	793.25
	100	30×15	72.67	237.33	951.50	2712.67	18.26	**354.73**	**100.28**	130.02	**11.92**	379.72	123.93	162.25	-	-	-	-
	200	30×15	39.00	95.50	386.00	961.00	2.10	95.69	38.70	32.15	**0.87**	**38.47**	**20.62**	15.64	-	-	-	-
	500	30×15	90.33	182.67	1007.50	4225.17	27.04	490.13	236.74	152.80	**15.01**	**169.47**	**96.51**	50.05	-	-	-	-
	750	30×15	42.50	130.33	340.17	700.50	31.78	252.13	155.23	85.75	**17.75**	**111.00**	**65.09**	31.53	-	-	-	-
	1000	30×15	52.83	137.50	746.50	1430.50	65.54	1862.27	592.63	666.16	**64.29**	**385.54**	**165.18**	112.73	-	-	-	-
	2000	30×15	48.83	166.17	321.67	1422.83	117.42	**1488.99**	812.84	591.12	**74.69**	2960.85	**672.29**	1127.22	-	-	-	-
	3000	30×15	53.67	169.33	922.67	1110.17	693.43	4100.11	2153.47	1614.83	**291.92**	**723.19**	**530.18**	204.45	-	-	-	-
	4000	30×15	51.8	-	-	-	-	-	-	-	**689.00**	**4101.20**	**1770.20**	1377.50	-	-	-	-

Bold values indicate best results obtained in the correspondent row.

The symbol '-' indicates that none of the 10 runs of the configuration terminated within the per-run time limit of 1200 seconds.

We can also observe the performance of proposed algorithms for SMOILP by comparing them to method Amrouche & Moulaï (2012), as it is shown in Figure 1, which is illustrates using a tri-objectif with two scenario instances.

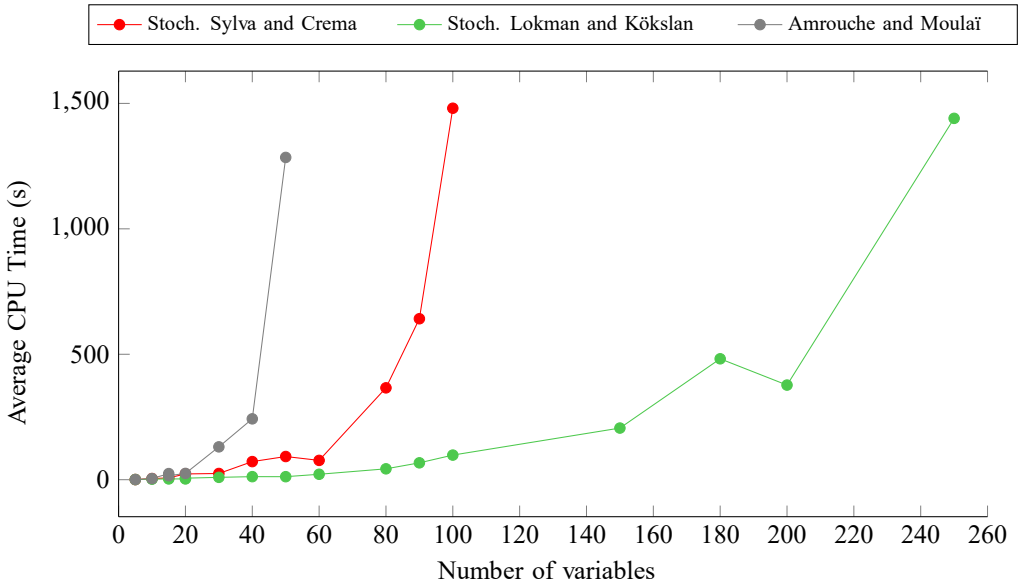


Figure 1 – Illustration of the proposed algorithms performance against the Amrouche and Moulaï algorithm regarding the instance size.

One can observe that the curve representing the Amrouche and Moulaï's method shows a steep increase, reaching 1200 seconds for just 50 variables. In contrast, our first algorithm, represented in red, processes 100 variables within the same execution time. Similarly, the blue curve progresses further, handling up to 250 variables in the same duration.

As to assess the overall contribution of the suggested algorithms, compared to the state-of-the-art approach, in terms of efficiency for solving SMOILPs, we adopt the speedup factor as the metric for comparison. This latter is defined as the ratio of Amrouche and Moulaï algorithm runtime over that of a suggested algorithm. Figure 2 illustrates the obtained results regarding the speedup factors. One can observe that, across all instance sizes, the obtained speedup factors are greater than 1 and attaining more than a factor of 10 for SSC method and 100 for SLK method. Furthermore, the speedup factor tends to increase when increasing the instance size. Hence, these results indicates a clear superiority of the suggested methods over the state-of-the-art method. On the other hand, as it is mentioned above, the experimental study has shown that SLK method is superior, in terms of efficiency, when compared to SSC method. Thus, we calculated the speedup factor between these latter to assess the overall SLK contribution over SSC method. The overall average speedup factor is 5.38 and the obtained maximal value is 15.1.

Considering the second part of Table 2, the obtained results show that both of the proposed algorithms, SSC method and SLK method, consistently outperformed Amrouche and Moulaï's

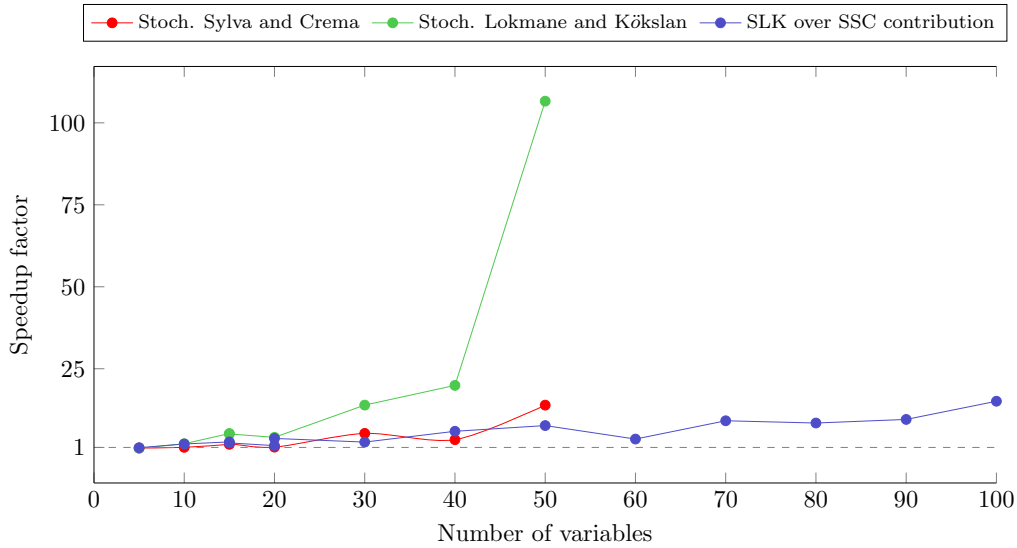


Figure 2 – Illustration of the speedup factors obtained for all competing algorithms.

method with respect to execution time, thereby reaffirming their computational advantages reported above. Notably, SLK method exhibited greater stability and superior efficiency compared to SSC method, as the number of scenarios increased. This indicates that SLK method is more effectively suited to adapt to problem instances with a small and medium number of scenarios. However, for large number of scenarios, exceeding 200 scenarios, SSC method is shown to be slightly more efficient than the SLK method.

Futhermore, one can notice in Table 2 that the number of scenarios has almost no impact on the performance of the suggested methods. This is in contrast with the state-of-the-art method, which surpasses the predefined time limit of 1200 seconds at only 50 scenarios. This observation can be clearly drawn from Figure 3. On the other hand, SLK method is shown to be comparable to SCC method for small and medium number of scenarios, and yet SCC method is slightly superior for large ones. Hence, the last drawn conclusion stands.

5.2 Evaluation on the GBD Benchmark Instance

To further assess the efficiency and the generality of the proposed algorithms on structured benchmark instances from the stochastic programming literature, we consider the George B. Dantzig (GBD) instance. GBD is a well-known benchmark derived from the aircraft allocation problem described by Dantzig (1963) and studied in Linderoth et al. (2006). In this problem, aircraft of different types must be allocated to routes in order to maximize profit under uncertain passenger demand. In addition to the cost of operating the aircraft, costs are incurred for bumping passengers when the demand for seats exceeds the available capacity.

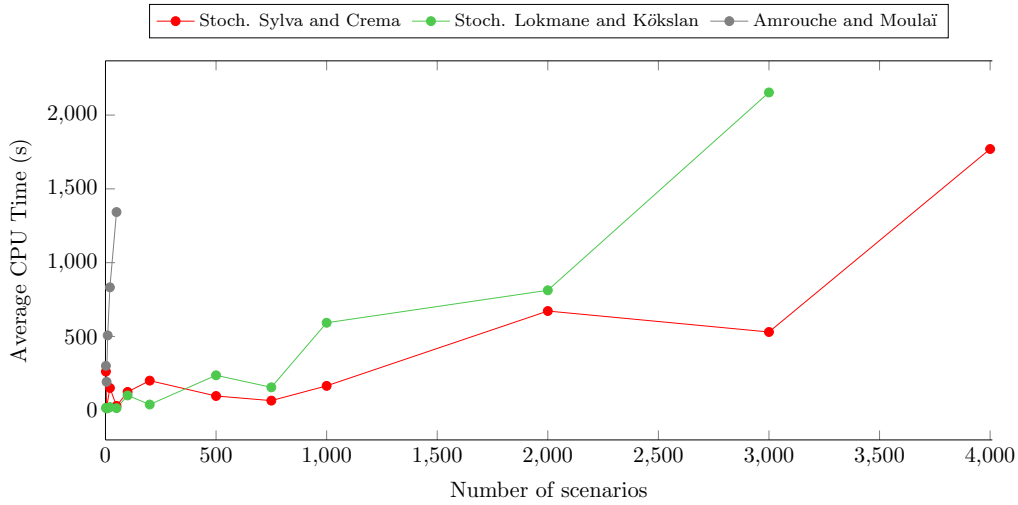


Figure 3 – Illustration of the proposed algorithms performance against the Amrouche and Moulaï algorithm regarding small and medium number of scenarios.

This instance features $n = 17$ integer first-stage variables, $m = 4$ deterministic constraints, $m_1 = 5$ equality recourse constraints, and $n_1 = 10$ continuous second-stage variables. The stochastic right-hand side values follow independent discrete distributions, generating a large Cartesian product of scenarios. The instance has fixed recourse, meaning that the technology matrix T, the recourse matrix W, and the penalty vector q do not depend on the scenario, and the recourse constraints are in equality form.

To obtain instances of varying difficulty, we use Sample Average Approximation (SAA) with sample sizes $R \in \{10, 40, 70, 100, 130\}$ drawn from the full scenario space Linderoth et al. (2006). The SMPS data files used in our experiments are taken from Wright (2005). We note that the recourse structure of the GBD instance differs from that of our randomly generated instances, as the recourse penalty vector contains both zero and non-zero entries with a specific pattern characteristic of real-world decomposition-based formulations. Furthermore, to create a bi-objective setting compatible with our SMOILP formulation, a second objective was generated using independent uniform random coefficients $c_j^2 \sim \mathscr{U}(10, 100)$.

Table 3 reports the results obtained with a time limit of 1200 seconds, where $|Y_N|$ denotes the number of efficient solutions found by each method for each scenario size R. Both SSC and SLK consistently identify the complete efficient set (i.e., 25 to 62 efficient solutions), whereas the Amrouche and Moulaï method finds at most three solutions within the 1200-second time limit. SLK achieves the fastest times across all scenario counts, confirming the scalability advantages observed on the random instances.

Table 3 – Performance comparison on the GBD benchmark instance with varying the number of scenarios.

Instances		Stochastic Sylva and Crema		Stochastic Lokman and Kökslan		Amrouche and Moulaï	
R	$\lvert\mathscr{X}_{eff}\rvert$	$\lvert Y_N\rvert$	Time (s)	$\lvert Y_N\rvert$	Time (s)	$\lvert Y_N\rvert$	Time (s)
10	62	62	125.65	62	**36.00**	1	-
40	28	28	111.74	28	**69.63**	1	-
70	28	28	160.91	28	**118.11**	3	-
100	25	25	213.90	25	**156.57**	2	-
130	25	25	240.83	25	**198.85**	1	-

Bold values indicate best results obtained in the correspondent row.

The symbol '-' indicates that the algorithm surpassed the predefined time limit of 1200 seconds.

5.3 Evaluation on the Stochastic Bi-Objective Knapsack Problem

Here, we evaluate the proposed methods on a classic combinatorial optimization problem under uncertainty. We introduce a structured experimental study centered on the Stochastic Bi-Objective Knapsack Problem (SBKP). This problem can be seen as a capacity-planning problem inspired by the pre-disaster pre-positioning of emergency supplies (Rawls & Turnquist, 2010): an agency must decide which essential items to procure before a disaster season, balancing population coverage against economic efficiency under uncertain post-disaster demands.

The SBKP is formulated as a two-stage stochastic bi-objective binary program, following the general SMOILP framework introduced in Section 2.2:

$$(SBKP)\begin{cases} \text{"}\max_{x}\text{"} & \tilde{z}_k = \tilde{c}^k x - Q(x), \quad k \in \{1,2\}, \\ s.t., & \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad \forall i \in \{1,\ldots,m\}, \\ & x_j \in \{0,1\}, \quad \forall j \in \{1,\ldots,n\}. \end{cases} \tag{17}$$

where, following Equation (7), $Q(x) = \sum_{r=1}^{R} \mathscr{P}^{(r)} Q(\xi^{(r)}, x)$ is the expected recourse penalty for unmet stochastic demand, and the recourse problem for each scenario $\xi^{(r)}$ is defined by:

$$Q(\xi^{(r)}, x) = \min\left\{ q(\xi^{(r)}) y \mid Wy = h(\xi^{(r)}) - T(\xi^{(r)}) x,\ y \geq 0 \right\}. \tag{18}$$

Here, $W = I_n$ is the identity matrix, $T(\xi^{(r)})$ is a diagonal technology matrix representing item effectiveness, $h(\xi^{(r)})$ is the stochastic demand vector, and $q(\xi^{(r)})$ is the emergency procurement cost vector for scenario $\xi^{(r)}$.

These instances were generated following established knapsack benchmarking principles (Pisinger, 2005). Anti-correlated objectives ensure a diverse and non-trivial Pareto front: $c_j^1 \sim \text{Uniform}(50, 150)$ and $c_j^2 = 200 - c_j^1$, so that items with high utility necessarily have low cost-effectiveness, and vice versa. Stochastic parameters are generated using stratified Latin Hypercube Sampling (McKay et al., 2000) with Poisson-distributed demands and exponentially scaled recourse penalties. All instances are provided in standard SMPS format (i.e., .cor, .tim, and .sto files). The complete instance set, including the generation methodology, SMPS-format data

files, and application context, is publicly available at https://github.com/kentachesarra/SBKP-Benchmark-instances.git.

We compared three methods on these benchmark instances: SSC, SLK, and the Amrouche and Moulaï method. Problem dimensions scale across $n \in \{10, 20, 30, 40\}$ variables and $R \in \{10, 30, 50, 70, 100\}$ scenarios.

Table 4 reports the obtained results for the runtime and the number of efficient solutions found $|Y_N|$ within the 1200 seconds time limit. For all instances, both SSC and SLK successfully enumerate the complete Pareto front within the 1200 seconds time limit, with SLK consistently achieving the fastest times. Amrouche and Moulaï method times out for all instances with $n \geq 20$, finding only partial frontiers. These findings confirm the clear superiority of the proposed methods over the Amrouche and Moulaï method, with SLK exhibiting the best overall scalability.

Table 4 – Performance comparison of the proposed algorithms against state-of-the-art Amrouche and Moulaï's algorithm on SBKP instances.

Instances			Stochastic Sylva and Crema		Stochastic Lokman and Kökslan		Amrouche and Moulaï	
n	R	$\lvert\mathscr{X}_{eff}\rvert$	$\lvert Y_N\rvert$	Time (s)	$\lvert Y_N\rvert$	Time (s)	$\lvert Y_N\rvert$	Time (s)
10	10	9	9	5.01	9	**3.84**	9	101.58
	30	10	10	32.01	10	**12.85**	10	148.19
	50	9	9	25.07	9	**19.93**	9	253.48
	70	11	11	34.41	11	**6.59**	11	333.06
	100	12	12	48.03	12	**42.05**	12	771.80
20	10	14	14	**37.98**	14	54.35	13	-
	30	12	12	69.10	12	**68.70**	11	-
	50	14	14	52.24	14	**41.57**	10	-
	70	13	13	60.01	13	**57.76**	10	-
	100	12	12	87.84	12	**86.76**	9	-
30	10	18	18	56.72	18	**47.96**	8	-
	30	31	31	250.19	31	**240.14**	5	-
	50	28	28	192.21	28	**176.31**	5	-
	70	31	31	276.13	31	**253.08**	5	-
	100	24	24	327.22	24	**296.34**	4	-
40	10	43	43	177.07	43	**115.79**	4	-
	30	44	44	499.51	44	**395.55**	4	-
	50	42	42	553.75	42	**458.62**	4	-
	70	41	41	869.78	41	**759.60**	4	-
	100	39	39	1094.17	39	**1060.15**	4	-

Bold values indicate best results obtained in the correspondent row.

The symbol '-' indicates that the algorithm surpassed the predefined time limit of 1200 seconds.

6 CONCLUSION

In this paper, we have presented two new exact algorithms for solving a discrete multi-objective stochastic linear program, designed to generate the entire efficient set of solutions. The primary

contribution of this work is the development of two novel methods that improve execution time by combining the classical L-shaped method with efficient state-of-the-art techniques adapted for discrete optimization. Moreover, the proposed approach extends practical efficacy in finding the entire efficient set of solutions for stochastic programs with fixed recourse and integer variables. Computational experiments demonstrate that our algorithms significantly outperform the recent state-of-the-art algorithm on randomly generated instances. Furthermore, an extended computational study on structured benchmark instances, including the GBD aircraft allocation instance and the newly introduced Stochastic Bi-Objective Knapsack Problem (SBKP), confirms the efficiency of both proposed methods on problems with real-world structural characteristics beyond randomly generated instances. Finally, as short-term perspectives, one can further extend the practicality of this approach by adapting it to non-linear problems and/or incorporating heuristic approaches.

References

ABDELAZIZ FB. 2012. Solution approaches for the multiobjective stochastic programming. *European Journal of Operational Research*, **216**(1): 1–16.

AHMED S, TAWARMALANI M & SAHINIDIS NV. 2004. A finite branch-and-bound algorithm for two-stage stochastic integer programs. *Mathematical Programming*, **100**(2): 355–377.

AMROUCHE S & MOULAÏ M. 2012. Multi-objective stochastic integer linear programming with fixed recourse. *International Journal of Multicriteria Decision Making 9*, **2**(4): 355–378.

BADAOUI I, MOULAÏ M, CHAIBLAINE Y & CHAABANE D. 2024. Biobjective integer stochastic optimization over the integer stochastic efficient set. *Pesquisa Operacional*, **44**: e281853.

BAKIR MA & BYRNE MD. 1998. Stochastic linear optimisation of an MPMP production planning model. *International Journal of Production Economics*, **55**(1): 87–96.

BENDERS JF. 1962. Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, **4**(1): 238–252.

BEREANU B. 1967. On stochastic linear programming distribution problems, stochastic technology matrix. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, **8**(2): 148–152.

BIRGE JR & LOUVEAUX F. 1997. *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. New York: Springer-Verlag.

BRANKE J. 2008. *Multiobjective optimization: Interactive and evolutionary approaches*. vol. 5252. Berlin: Springer Science & Business Media.

CHAABANE D & MEBREK F. 2013. Optimization over stochastic integer efficient set. In: *Optimization Theory, Decision Making, and Operations Research Applications*. pp. 103–115. Springer.

CHANKONG V & HAIMES YY. 2008. *Multiobjective decision making: theory and methodology*. New york: Courier Dover Publications.

CHARNES A, COOPER WW & SYMONDS GH. 1958. Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management science*, **4**(3): 235–263.

DANTZIG GB. 1955. Linear programming under uncertainty. *Management science*, **1**(3-4): 197–206.

DANTZIG GB. 1963. *Linear Programming and Extensions*. Princeton, NJ: Princeton University Press.

GOICOECHEA A. 1977. A multi-objective, stochastic programming model in watershed management. *The University of Arizona*, .

HELFRICH S, PERINI T, HALFFMANN P, BOLAND N & RUZIKA S. 2023. Analysis of the weighted Tchebycheff weight set decomposition for multiobjective discrete optimization problems. *Journal of Global Optimization*, **86**(2): 417–440.

KALL P & WALLACE SW. 1994. *Stochastic programming*. vol. 6. Berlin: Springer.

KING AJ. 1986. *Asymptotic behaviour of solutions in stochastic optimization: non smooth analysis and derivation of non-normal limit distribution (least squares)*. Ph.D. thesis. University of Washington.

KLEIN D & HANNAN E. 1982. An algorithm for the multiple objective integer linear programming problem. *European journal of operational research*, **9**(4): 378–385.

KONG N, SCHAEFER AJ & HUNSAKER B. 2006. Two-stage integer programs with stochastic right-hand sides: a superadditive dual approach. *Mathematical Programming*, **108**(2): 275–296.

LAPORTE G & LOUVEAUX FV. 1993. The integer L-shaped method for stochastic integer programs with complete recourse. *Operations research letters*, **13**(3): 133–142.

LEÓN J, PUERTO J & VITORIANO B. 2020. A risk-aversion approach for the multiobjective stochastic programming problem. *Mathematics*, **8**(11): 2026.

LINDEROTH J, SHAPIRO A & WRIGHT S. 2006. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research*, **142**(1): 215–241.

LOKMAN B & KÖKSALAN M. 2013. Finding all nondominated points of multi-objective integer programs. *Journal of Global Optimization*, **57**(2): 347–365.

MADANSKY A. 1960. Inequalities for stochastic linear programming problems. *Management science*, **6**(2): 197–204.

MARCOTTE O & SOLAND RM. 1986. An interactive branch-and-bound algorithm for multiple criteria optimization. *Management Science*, **32**(1): 61–75.

MCKAY MD, BECKMAN RJ & CONOVER WJ. 2000. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, **42**(1): 55–61.

PISINGER D. 2005. Where are the hard knapsack problems? *Computers & Operations Research*, **32**(9): 2271–2284.

RAWLS CG & TURNQUIST MA. 2010. Pre-positioning of emergency supplies for disaster response. *Transportation Research Part B: Methodological*, **44**(4): 521–534.

SCHULTZ R, STOUGIE L & VAN DER VLERK MH. 1998. Solving stochastic programs with integer recourse by enumeration: A framework using Gröbner basis. *Mathematical Programming*, **83**(1): 229–252.

SLOWINSKI R & TEGHEM JR J. 1988. Fuzzy versus stochastic approaches to multicriteria linear programming under uncertainty. *Naval Research Logistics (NRL)*, **35**(6): 673–695.

STANCU-MINASIAN I. 1984. *Stochastic programming with multiple objective functions*. vol. 13. California: Springer.

STEUER RE. 1986. Multiple criteria optimization. *Theory, computation and applications*, .

SYLVA J & CREMA A. 2004. A method for finding the set of non-dominated vectors for multiple objective integer linear programs. *European Journal of Operational Research*, **158**(1): 46–55.

SYLVA J & CREMA A. 2007. A method for finding well-dispersed subsets of non-dominated vectors for multiple objective mixed integer linear programs. *European journal of operational research*, **180**(3): 1011–1027.

TEGHEM J. 1983. Multiobjective and stochastic linear programming. *Foundations of Control Engineering*, **8**(3-4): 225–232.

TEGHEM JR J, DUFRANE D, THAUVOYE M & KUNSCH P. 1986. STRANGE: an interactive method for multi-objective linear programming under uncertainty. *European Journal of Operational Research*, **26**(1): 65–82.

URLI B & NADEAU R. 1990. Multiobjective stochastic linear programming with incomplete information: A general methodology. In: *Stochastic Versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty*. pp. 131–161. Springer.

VAN SLYKE RM & WETS R. 1969. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM journal on applied mathematics*, **17**(4): 638–663.

WALKUP DW & WETS RJB. 1967. Stochastic programs with recourse. *SIAM Journal on Applied Mathematics*, **15**(5): 1299–1314.

WALKUP DW & WETS RJB. 1969. Stochastic programs with recourse II: On the continuity of the objective. *SIAM Journal on Applied Mathematics*, **17**(1): 98–103.

WETS RJ. 2002. Stochastic programming models: wait-and-see versus here-and-now. In: *Decision Making Under Uncertainty*. pp. 1–15. Springer.

WETS RJB. 1966. Programming under uncertainty: the equivalent convex program. *SIAM Journal on Applied Mathematics*, **14**(1): 89–105.

WETS RJB. 1974. Stochastic programs with fixed recourse: the equivalent deterministic program. *SIAM Review*, **16**(3): 309–339.

WRIGHT SJ. 2005. Sampling in Stochastic Optimization. Available at: https://pages.cs.wisc.edu/~swright/stochastic/sampling/. Accessed: May 1, 2026.

ZAIDI A, CHAABANE D, MOURAD A, MATOUB S & IDIR L. 2025. A new method for optimizing a function over the efficient set. *Pesquisa Operacional*, **45**: e289592.

ZIONTS S. 1977. *Integer linear programming with multiple objectives*. vol. 1. New york: Elsevier. 551-562 pp.

How to cite

KENTACHE S & CHAABANE D. 2026. Exact algorithms for generating the efficient set of multi-objective stochastic integer linear programs. *Pesquisa Operacional*, **46**: e301123. doi: 10.1590/0101-7438.2026.046.00301123.

Author Contributions

Pesquisa Operacional adopts the CRediT – Contributor Roles Taxonomy to identify the contribution of each author to the publication. The contributions of the authors are as follows: Sarra Kentache: Conceptualization, Formal Analysis, Investigation, Resources, Software, Writing - Original Draft Preparation, Writing - Review and Editing. Djamal Chaabane: Conceptualization, Formal Analysis, Methodology, Project Administration, Resources, Supervision, Validation, Writing - Original Draft Preparation, Writing - Review and Editing.

Funding

The authors received no financial support for the research and the publication of this article.

Data Availability

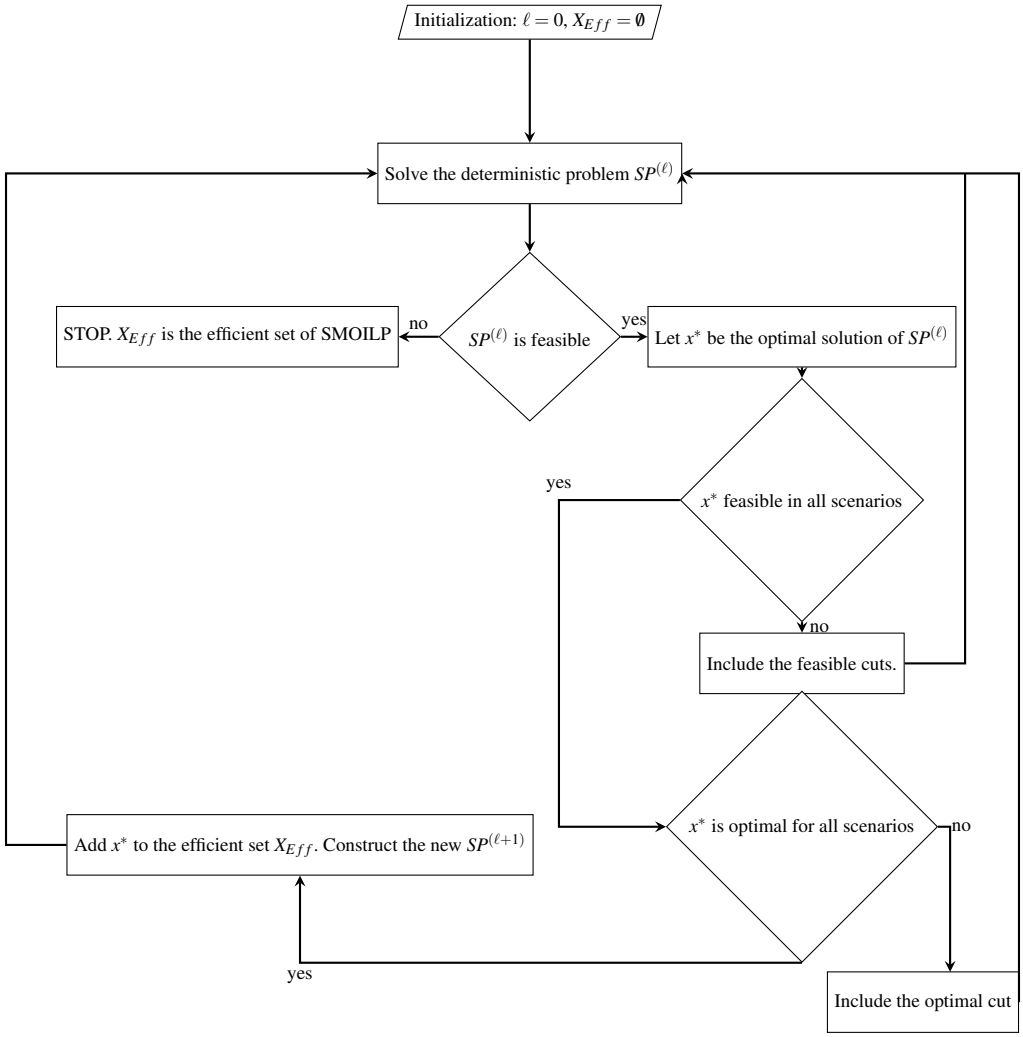
The SBKP benchmark instances generated and analyzed during this study are publicly available in the GitHub repository: https://github.com/kentachesarra/SBKP-Benchmark-instances.git.

Editor responsible for the review

Editor-in-Chief: Antônio Augusto Chaves.

APPENDIX A FLOWCHART OF SSC METHOD

In this appendix, we present the flowchart illustrating the process of the SSC suggested method.



APPENDIX B FLOWCHART OF SLK METHOD

In this appendix, we present the flowchart illustrating the process of the SLK suggested method.

