Available online at http://jprm.sms.edu.pk/
Journal of Prime Research in Mathematics, (2026), 203–217

Journal of Prime Research in Mathematics

# An Exact L-Shaped Decomposition Algorithm for Multi-Objective Two-Stage Stochastic Integer Programs

Guettouche Abd Essamed[a,*], Chaabane Djamal[a]

[a] *Department of Operations Research, University of Sciences and Technology Houari Boumediene, Algiers, Algeria.*

**Abstract**

This paper studies a finite-scenario multi-objective two-stage stochastic integer programming problem with fixed recourse. We propose a decomposition-based solution framework that combines L-shaped feasibility and optimality cuts with an efficient-solution generation procedure for the first-stage integer decisions. This novel exact framework effectively integrates the L-shaped decomposition method with an advanced exploration strategy for generating the complete set of Pareto-efficient solutions. Unlike existing approaches that often struggle with scalability, our methodology leverages these specialized cuts to navigate the stochastic search space while systematically identifying non-dominated integer points. The efficiency and robustness of the proposed algorithm are demonstrated through comprehensive numerical experiments on a diverse set of randomly generated instances. The results indicate that our method provides a significant improvement in computational precision and reliability compared to traditional techniques. By establishing a robust performance database, this study underscores the viability of the proposed approach for large-scale stochastic decision-making and provides a solid foundation for future research in multi-objective optimization under uncertainty.

*Keywords:* Multi-objective optimization, Two-stage stochastic programming, Integer linear programming, L-shaped method, Exact algorithm, Pareto efficiency
*2010 MSC:* 90C15, 90C29, 90C10, 90C05, 90B10, 90B15.

## 1. Introduction

Two-stage stochastic programming originates in seminal work by Dantzig [17], Dantzig and Madansky [20], and Beale [7], and has since developed into a central framework for optimization under uncertainty. This early research laid the groundwork for stochastic programming, a domain addressing optimization problems where parameters are random variables rather than deterministic quantities. Stochastic programming has

*Corresponding author
*Email addresses:* `aguettouche@usthb.dz` (Guettouche Abd Essamed), `dchaabane@usthb.dz` (Chaabane Djamal)

*Received :* 28 August 2025*; Accepted:* 17 May 2026*; Published Online:* 21 June 2026

since expanded to address diverse practical challenges across economics, healthcare, transportation, and military technology.

The primary objective in stochastic programming is to devise solutions that remain robust despite uncertainties. The Stochastic Multi-Objective Integer Problem (SMOIP) extends this concept by incorporating multiple conflicting objectives alongside random constraints. Mathematically, an SMOIP can be formulated as follows:

$$(P^{\zeta})\begin{cases}\max \quad Z_k(\zeta) = C^k(\zeta)x, \quad k = 1, \ldots, K \\ \text{subject to:} \\ Ax = b \\ T(\zeta)x = h(\zeta) \\ x \in \mathbb{N}^n\end{cases}$$

Where $C^k(\zeta) \in \mathbb{R}^{1\times n}$, $T(\zeta) \in \mathbb{R}^{m_0\times n}$, and $h(\zeta) \in \mathbb{R}^{m_0\times 1}$ are realizations of random vectors defined over a discrete probability space $(\Omega, \Xi, \mathbb{P})$. These variables follow a discrete joint distribution defined by probabilities $P_r > 0, \forall r \in \{1, \ldots, R\}$, such that $\sum_{r=1}^{R} P_r = 1$. The matrices $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^{m\times 1}$ represent the deterministic constraints.

Stochastic linear programming has made substantial strides in handling probabilistic constraints. Significant contributions include discrete stochastic optimization [10], continuous techniques [15, 5, 3, 27], and fuzzy environment methods [30]. Deterministic discrete optimization has also advanced through the works of [1, 14, 2], while methodologies for generating efficient solutions in combinatorial problems have been enhanced by [24].

Our research addresses the complexities of stochastic problems with integer variables by proposing a novel exact algorithm. This method integrates the L-Shaped decomposition [26] with advanced techniques for generating the complete Pareto-efficient set [24]. This integrated approach aims to overcome the limitations of existing methods, which often struggle with large-scale scenarios. For instance, while [5] managed three scenarios with ten objectives, our work underscores the need for solutions capable of handling more extensive datasets and higher dimensionality.

Recent advances in multi-objective integer programming include the L-shape search method for tri-objective problems [12] and the quadrant shrinking method [11], among other enumeration and search techniques. Additionally, [29] and [32] proposed recursive algorithms for nondominated solutions, while [8, 37, 33] explored disjunctive programming and branch-and-bound methods. Despite these advancements, the feasibility of solving problems with a large number of objectives (e.g., 10 or 20) remains a critical gap in the literature.

This study extends the current knowledge base by presenting a comprehensive framework for complex stochastic optimization. The remainder of this paper is organized as follows: Section 2 introduces essential definitions and theoretical results. Section 3 details the proposed algorithm, followed by numerical illustrations and a discussion of the results in the subsequent sections.

## 2. Background and Preliminaries

This section establishes the theoretical and mathematical foundations essential for the subsequent developments in this paper. We begin by introducing the fundamental notations and terminology to ensure a consistent and clear presentation of the models. Subsequently, we detail the core concepts of multi-objective integer linear programming (MOILP) and two-stage stochastic programming, with a specific focus on the properties of the recourse function and the dominance criteria in a multi-objective context. Furthermore, we highlight key theoretical results from the literature that underpin our proposed methodology. Collectively, these elements form the formal framework required to analyze the efficiency and convergence of the algorithm presented in this research.

### *2.1. Multi-objective integer linear optimization*

A Multi-Objective Integer Linear Programming (MOILP) problem can be mathematically formulated as follows:

$$(P)\begin{cases}\max \quad z_k = c^k x, \quad k = 1, 2, \dots, K\\ \text{subject to:}\\ x \in X\end{cases}$$

where $X = \{x \in \mathbb{N}^n \mid Ax = b\}$ represents the feasible region in the decision space. Here, $c^k \in \mathbb{R}^{1\times n}$ for each $k$ denotes the vector of cost coefficients, $A \in \mathbb{R}^{m\times n}$ is the constraint matrix, and $b \in \mathbb{R}^{m\times 1}$ is the right-hand side vector. $K$ indicates the number of conflicting objective functions to be maximized.To identify the set of efficient solutions, we utilize the **dominance criterion**. A solution $x \in X$ is said to dominate another solution $\hat{x} \in X$ if and only if:

1. $z_k(x) \geq z_k(\hat{x})$ for all $k \in \{1, 2, \dots, K\}$, and
2. $z_k(x) > z_k(\hat{x})$ for at least one $k \in \{1, 2, \dots, K\}$.

### *2.2. L-shaped Method for Two-stage Stochastic Programs with Recourse*

Two-stage stochastic programming with fixed recourse, pioneered by Dantzig and Beale [7, 17, 20], is a fundamental framework for modeling optimization problems under uncertainty. The L-shaped method, an extension of Benders decomposition, is specifically designed to handle such structures by partitioning the decision process into two distinct stages:

- **First-stage variables ($x$):** Decisions made before the realization of the random event $\zeta$.
- **Second-stage variables ($z$):** Recourse actions taken after the uncertainty is revealed to compensate for any deficiencies in the first stage.

The Deterministic Equivalent Program (DEP) for the stochastic model is written as:

$$(\bar{P}^{\zeta})\begin{cases}\max \quad \bar{Z}_k = \mathbb{E}_{\zeta}[C^k(\zeta)x] - \mathbb{E}_{\zeta}[\mathcal{Q}(x,\zeta)], \quad k = 1, \dots, K\\ \text{subject to:}\\ Ax = b\\ x \in \mathbb{N}^n\end{cases}$$

where $\mathcal{Q}(x,\zeta)$ is the optimal value of the second-stage recourse sub-problem:

$$\mathcal{Q}(x,\zeta) = \min{}_z q(\zeta)^T z \mid Wz = h(\zeta) - T(\zeta)x,\ z \geq 0 \tag{2.1}$$

In this context, $\zeta$ represents the vector of random variables, $\mathbb{E}_{\zeta}$ denotes the mathematical expectation, $W$ is the fixed recourse matrix ($m_0 \times n_0$), and $q(\zeta)$ is the recourse cost vector. The expectation of the linear objective function is computed as:

$$\mathbb{E}_{\zeta}\left[\sum_{i=1}^{n} c_i(\zeta)x_i\right] = \sum_{i=1}^{n}\left(\sum_{r=1}^{R} P_r c_i(\zeta^r)\right)x_i$$

#### *2.2.1. Feasibility*

For a first-stage solution $x^0$, feasibility in the second stage across all scenarios $\zeta^r$ must be guaranteed. According to *Farkas' Lemma* [22], the recourse problem is feasible if and only if for all scenarios $r = 1, \dots, R$:

$$\delta^T(h(\zeta^r) - T(\zeta^r)x^0) \leq 0 \quad \text{for all } \delta \in \{\delta \mid \delta^T W \leq 0\}$$

To identify violations, we solve the following feasibility sub-problem:

$$(P^{\delta})\begin{cases}\max \quad \delta^T(h(\zeta^r)-T(\zeta^r)x^0)\\ \text{subject to:}\\ \delta^T W \leq 0, \quad \|\delta\|_1 \leq 1\end{cases}$$

If an optimal solution $\hat{\delta}$ yields a positive objective value for any scenario $\zeta^r$, the following **feasibility cut** is added to the master problem:

$$\hat{\delta}^T(h(\zeta^r)-T(\zeta^r)x) \leq 0 \tag{2.2}$$

*2.2.2. Optimality*

Assuming feasibility is satisfied, the problem $(\bar{P}^{\zeta})$ can be reformulated by introducing an auxiliary variable $\theta$ to approximate the expected recourse function $\mathcal{Q}(x)=\mathbb{E}_{\zeta}[\mathcal{Q}(x,\zeta)]$:

$$(\bar{P}^{\zeta}_{\theta})\begin{cases}\max \quad \bar{Z}_k=\mathbb{E}_{\zeta}[C^k(\zeta)x]-\theta, \quad k=1,\dots,K\\ \text{subject to:}\\ \theta \geq \mathcal{Q}(x)\\ x\in \mathbb{S}\cap\mathbb{N}^n\end{cases}$$

where $\mathbb{S}$ is the feasible region defined by the first-stage constraints and any existing feasibility cuts. Since $\mathcal{Q}(x)$ is a convex piecewise linear function, it can be approximated via **optimality cuts**. Given a trial solution $(x^0,\theta^0)$, if $\theta^0<\mathcal{Q}(x^0)$, the following cut is added:

$$\theta \geq \sum_{r=1}^{R} P_r(\hat{\pi}^r)^T[h(\zeta^r)-T(\zeta^r)x] \tag{2.3}$$

In this section, we have established the theoretical foundation for multi-objective integer linear optimization and two-stage stochastic programming. By introducing the mathematical frameworks and key concepts—such as the dominance criterion and the L-shaped decomposition—we have provided the essential tools for addressing optimization under uncertainty.

Despite significant advancements, existing algorithms frequently encounter scalability issues as the dimensionality of the scenario set or the number of objectives increases. For instance, as noted by [5], traditional methods often exhibit performance degradation when managing more than three scenarios or a high number of criteria. This limitation underscores the urgent need for more robust methodologies capable of handling such stochastic complexities efficiently.

To address these challenges, the subsequent section introduces a novel exact algorithm designed to overcome these identified bottlenecks. We will detail the algorithm's architecture, its underlying mathematical principles, and its practical application through a numerical illustration. Furthermore, we will demonstrate how this proposed approach enhances computational efficiency and provides a more comprehensive exploration of the Pareto frontier compared to existing methods in the literature.

## 3. The proposed algorithm

Addressing Stochastic Multi-Objective Integer Programming (SMOIP) necessitates a robust framework capable of balancing the discrete nature of decision variables with the probabilistic uncertainty of the parameters. The complexity is twofold: the combinatorial explosion typical of integer programming and the nested structure of two-stage recourse. In this section, we present an exact decomposition-based algorithm specifically tailored for SMOIP with fixed recourse. Our approach hybridizes the L-shaped decomposition principle with an efficient solution generation technique, ensuring that the complete Pareto-optimal set is identified within a finite number of iterations.

*3.1. Description of the method*

The SMOIP model is inherently intractable in its raw stochastic form. The core strategy of our method involves a two-step transformation. First, the stochastic parameters are discretized into a finite set of scenarios $\Omega = \{\zeta_1, \zeta_2, \ldots, \zeta_R\}$, allowing the transformation of the model into its Deterministic Equivalent Program (DEP), denoted as $\bar{P}^{\zeta}$. Second, we decompose this large-scale deterministic problem into a master integer program and a series of linear recourse sub-problems.Building upon the algorithmic foundations of [24] and the L-shaped decomposition of [26], we propose a systematic exploration of the integer decision space. The algorithm identifies non-dominated solutions by solving a sequence of augmented sub-problems $(\grave{P}^k)$. These sub-problems are designed to explore new regions of the feasible space while strictly avoiding previously identified points through the incorporation of feasibility and optimality cuts.For each objective $k \in \{1, \ldots, K\}$, a sub-problem $(\grave{P}^k)$ is formulated. At any given iteration, $(\grave{P}^k)$ aims to find a new candidate solution $x$ that is not dominated by the current set of efficient solutions $E$. The sub-problem at node $k$ is defined as follows:

$$(\grave{P}^k) \begin{cases} \max \quad \grave{Z}_k = \mathbb{E}_{\zeta}(C^k(\zeta)x) \\ \text{s.t:} \\ x \in \mathbb{S}^k \\ x \in \mathbb{N}^n \end{cases}$$

The feasible region $\mathbb{S}^k$ is dynamically updated at each step. It represents the intersection of the initial decision space and the accumulated cuts generated from the recourse sub-problems. Starting from the initial continuous relaxation $\mathbb{S}^0 = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$, the region at iteration $k+1$ is defined as:

$$\mathbb{S}^{k+1} = \mathbb{S}^k \cap \{x \mid \hat{\sigma}^r\left(h(\zeta^r) - T(\zeta^r)x\right) \leq 0, \quad r = 1, \ldots, N_k\} \cap \{x \mid \theta^e \geq \mathcal{Q}(x), \quad e = 1, \ldots, M_k\}$$

In this recursive definition:

- $N_k$ and $M_k$ denote the total number of **feasibility cuts** and **optimality cuts**, respectively, added to the master problem up to iteration $k$.
- The feasibility cuts $\hat{\sigma}^r(h(\zeta^r) - T(\zeta^r)x) \leq 0$ are derived using the dual rays of the second-stage recourse problem, ensuring that the first-stage decision $x$ remains feasible for scenario $\zeta^r$.
- The optimality cuts $\theta^e \geq \mathcal{Q}(x)$ utilize the dual optimal solutions to provide a support for the convex recourse function $\mathcal{Q}(x)$.

**Theorem 3.1.** *The optimal solution $x^*$ of the scalarized sub-problem $(\grave{P}^k)$ is an efficient solution for the original multi-objective stochastic problem $(\bar{P}^{\zeta}_{\theta})$.*

*Proof.* The proof proceeds by contradiction. Suppose that $x^*$ is the optimal solution to $(\grave{P}^k)$ but is not an efficient solution for $(\bar{P}^{\zeta}_{\theta})$. By the definition of Pareto efficiency, there must exist a feasible solution $x' \in \mathbb{S}^k$ such that:

$$Z_j(x') \geq Z_j(x^*) \quad \forall j \in \{1, \ldots, K\} \tag{3.1}$$

with at least one index $i$ where $Z_i(x') > Z_i(x^*)$.

In the scalarized formulation of $(\grave{P}^k)$, the objective function is strictly increasing with respect to each individual objective component $Z_j$. Consequently, the existence of such an $x'$ implies that the scalarized objective value at $x'$ must satisfy:

$$\mathcal{F}(Z(x')) > \mathcal{F}(Z(x^*)) \tag{3.2}$$

where $\mathcal{F}$ denotes the scalarization function used in the sub-problem. This inequality directly contradicts the initial assumption that $x^*$ is the optimal solution to $(\grave{P}^k)$. Therefore, no such $x'$ can exist, and $x^*$ must be an efficient solution for the multi-objective problem $(\bar{P}^{\zeta}_{\theta})$. □

*3.1.1. Phase 1: Generating the Initial Set of Efficient Solutions*

**For** $l = 0$: Let $G_0 = \{x_1^*, x_2^*, \ldots, x_\alpha^*\}$ be defined as the set of efficient solutions obtained by solving the master sub-problems $(\grave{P}^k)$ for each objective $k = 1, \ldots, K$. The following algorithm describes the iterative process to determine the set $G_0$:

**Algorithm 1** Algorithm for Generating the Initial Efficient Set $G_0$

**Input**:
  $n, m, n_0, m_0, K$: Problem dimensions and number of objectives;
  $R$: Number of scenarios;
  $A, b$: Deterministic constraint matrix and RHS;
  $C(\xi), T(\xi), h(\xi)$: Stochastic objective and technology matrices;
  $W, q, P_r$: Recourse matrix, cost vector, and scenario probabilities.
**Output**:
  $G_0$: The set of obtained efficient solutions.
**Initialization**:
$G_0 \leftarrow \emptyset$, $\mathbb{S}^1 \leftarrow \mathbb{S}^0$;
**for** $k = 1$ **to** $K$ **do** ▷ Loop over each objective function
  Set $\theta \leftarrow -\infty$, $N_k \leftarrow 0$, $M_k \leftarrow 0$;
  **loop** ▷ Iterative process for cuts generation
    Solve the Master Sub-problem $(\grave{P}^k)$;
    **if** $(\grave{P}^k)$ does not have a feasible solution **then**
      **break** (Move to next objective $k + 1$);
    **end if**
    Let $x^k$ be the optimal solution of $(\grave{P}^k)$;
    **Feasibility Test**: Solve $(P^\delta)$ for all scenarios $r = 1, \ldots, R$;
    **if** $\exists r \in \{1, \ldots, R\}$ such that $(\hat{\delta}^r)^T(h(\xi^r) - T(\xi^r)x^k) > 0$ **then**
      $N_k \leftarrow N_k + 1$;
      Add Feasibility Cut to $\mathbb{S}^k$: $(\hat{\delta}^r)^T(h(\xi^r) - T(\xi^r)x) \leq 0$;
    **else**
      **Optimality Test**: Solve dual sub-problems $(P^\pi)$ for all $\xi^r$;
      Compute $\mathbb{Q}(x^k) = \sum_{r=1}^R P_r \mathcal{Q}(x^k, \xi^r)$;
      **if** $\mathbb{Q}(x^k) > \theta$ **then**
        $M_k \leftarrow M_k + 1$;
        Add Optimality Cut to $\mathbb{S}^k$: $\theta \geq \sum_{r=1}^R P_r(\hat{\pi}^r)^T(h(\xi^r) - T(\xi^r)x)$;
      **else**
        $G_0 \leftarrow G_0 \cup \{x^k\}$;
        **break** (Optimal for objective $k$ reached);
      **end if**
    **end if**
  **end loop**
  Update global feasible region: $\mathbb{S}^{k+1} \leftarrow \mathbb{S}^k$;
**end for**
**return** $G_0$;

*3.1.2. Phase 2: Generating the Complete Set of Efficient Solutions*

**For** $l \geq 1$**:** Consider the following scalarized problem $(\tilde{P}^l)$ to explore the decision space:

$$(\tilde{P}^l)\begin{cases}\max \quad \sum_{k=1}^{K} \mathbb{E}_\zeta[C^k(\zeta)x] \\ \text{subject to:} \\ x \in \mathbb{F}^l \\ x \in \mathbb{N}^n\end{cases}$$

where $\mathbb{F}^{l+1} = \mathbb{F}^l \cap \{(\hat{\delta}^r)^T(h(\zeta^r) - T(\zeta^r)x) \leq 0, \theta^e \geq \mathbb{Q}(x)\}$ represents the feasible region augmented with cuts. Let $\mathbb{F}^0 = \mathbb{S}^K$ from Phase 1.To ensure the generation of new efficient solutions, we introduce non-dominance constraints relative to each $x_q^* \in G_l$. For each previously found solution $x_q^*$, the following logic is applied:

$$\mathbb{E}\zeta[C^k(\zeta)]x > \mathbb{E}\zeta[C^k(\zeta)]x^*q - Mtkq, \quad k = 1, \ldots, K \tag{3.3}$$

$$\sum_{k=1}^{K} t_{kq} \leq K - 1, \quad t_{kq} \in \{0, 1\} \tag{3.4}$$

Consequently, the augmented problem $(\tilde{P}^l)$ is written as:

$$(\tilde{P}^l)\begin{cases}\max \quad \sum_{k=1}^{K} \mathbb{E}_\zeta[C^k(\zeta)x] \\ \text{s.t:} \\ x \in \mathbb{F}^l \cap \mathbb{N}^n \\ \mathbb{E}_\zeta[C^k(\zeta)]x \geq \mathbb{E}_\zeta[C^k(\zeta)]x_q^* + 1 - Mt_{kq}, \quad \forall k, q \\ \sum_{k=1}^{K} t_{kq} \leq K - 1, \quad \forall q \in \{1, \ldots, \beta\} \\ t_{kq} \in \{0, 1\}\end{cases}$$

Note that if $t_{kq} = 1$, the constraint is redundant due to the large constant $M$. The condition $\sum t_{kq} \leq K - 1$ ensures that for each $x_q^*$, at least one objective must be strictly improved.

**Theorem 3.2.** *The optimal solution $x^*$ of the augmented scalarized problem $(\tilde{P}^l)$ at any iteration $l$ is a Pareto-efficient solution for the original Multi-Objective Two-Stage Stochastic Integer Program (MOTSIP).*

*Proof.* Let $x^*$ be an optimal solution to the scalarized master problem $(\tilde{P}^l)$ at iteration $l$, where the objective function is defined as maximizing the sum of expectations:

$$\max \sum_{k=1}^{K} E_\zeta[C^k(\zeta)x + \mathcal{Q}_k(x, \zeta)]$$

subject to $x \in \mathbb{F}^l$, where $\mathbb{F}^l$ includes the stochastic L-shaped cuts and the non-dominance constraints (3.1)–(3.2).

We proceed by contradiction. Suppose $x^*$ is not Pareto-efficient. By definition, there must exist a feasible solution $\hat{x} \in \mathbb{F}^l$ such that:

1. $E_\zeta[C^k(\zeta)\hat{x} + \mathcal{Q}_k(\hat{x}, \zeta)] \geq E_\zeta[C^k(\zeta)x^* + \mathcal{Q}_k(x^*, \zeta)]$ for all $k \in \{1, \ldots, K\}$,
2. $E_\zeta[C^j(\zeta)\hat{x} + \mathcal{Q}_j(\hat{x}, \zeta)] > E_\zeta[C^j(\zeta)x^* + \mathcal{Q}_j(x^*, \zeta)]$ for at least one index $j \in \{1, \ldots, K\}$.

Summing these inequalities over all $k$, we obtain:

$$\sum_{k=1}^{K} E_\zeta[C^k(\zeta)\hat{x} + \mathcal{Q}_k(\hat{x},\zeta)] > \sum_{k=1}^{K} E_\zeta[C^k(\zeta)x^* + \mathcal{Q}_k(x^*,\zeta)]$$

Case 1: If $\hat{x}$ is feasible for $(\tilde{P}^l)$, its strictly higher objective value directly contradicts the optimality of $x^*$.

Case 2: If $\hat{x}$ is not feasible for $(\tilde{P}^l)$, it must have been excluded by the non-dominance constraints relative to the set of previously identified efficient solutions $G_l$. This implies there exists some $x^q \in G_l$ that dominates $\hat{x}$. By the transitivity of the dominance relation, if $x^q$ dominates $\hat{x}$ and $\hat{x}$ dominates $x^*$, then $x^q$ must dominate $x^*$.

However, the non-dominance constraints in $(\tilde{P}^l)$ are specifically designed to exclude any solution dominated by $G_l$ through the logic:

$$\sum_{k=1}^{K} t_{kq} \leq K-1, \quad \forall x^q \in G_l$$

Since $x^*$ was returned as the optimal solution to $(\tilde{P}^l)$, it must satisfy these constraints and cannot be dominated by any $x^q \in G_l$. This contradicts the assumption that $x^*$ is dominated by $\hat{x}$ (which in turn is dominated by $x^q$).

Therefore, no such $\hat{x}$ exists, and $x^*$ is a Pareto-efficient solution. □

**Algorithm 2** Algorithm for Generating the Complete Efficient Set $G_{total}$

**Input**: Initial set $G_0$, feasible region $\mathbb{S}^K$;
**Initialization**: $l \leftarrow 1$, $G_{total} \leftarrow G_0$, $\mathbb{F}^1 \leftarrow \mathbb{S}^K$, $End \leftarrow$ false;
**while** $End =$ false **do**
    Solve current Master Problem $(\tilde{P}^l)$;
    **if** $(\tilde{P}^l)$ is Infeasible **then**
        $End \leftarrow$ true;
    **else**
        Let $x^l$ be the optimal solution of $(\tilde{P}^l)$;
        **Feasibility and Optimality Test**:
        **if** $\exists r \in \{1,\ldots,R\} : (\hat{\delta}^r)^T(h(\xi^r) - T(\xi^r)x^l) > 0$ **then**
            Add Feasibility Cut to $\mathbb{F}^l$;
            Solve $(\tilde{P}^l)$ again with the new cut;
        **else if** $\mathbb{Q}(x^l) > \theta$ **then**
            Add Optimality Cut to $\mathbb{F}^l$;
            Solve $(\tilde{P}^l)$ again with the new cut;
        **else**
            $G_{total} \leftarrow G_{total} \cup \{x^l\}$;
            Update $\beta \leftarrow |G_{total}|$;
            $l \leftarrow l + 1$;
            Update $\mathbb{F}^l$ with non-dominance constraints for $x^l$;
        **end if**
    **end if**
**end while**
**return** $G_{total}$;

**Theorem 3.3** (Convergence)**.** *The proposed algorithm converges to the complete set of efficient solutions $G_{total}$ in a finite number of iterations.*

*Proof.* The initial feasible region $\mathbb{S}$ is defined as a compact polyhedron in $\mathbb{R}^n$. By definition, a polyhedron is the intersection of a finite collection of half-spaces $\{x \mid a_j^T x \leq b_j\}$. Since the decision variables are restricted to the natural numbers ($x \in \mathbb{N}^n$), the set of candidate integer solutions $\mathbb{X} = \mathbb{S} \cap \mathbb{N}^n$ is necessarily finite. The convergence of the proposed algorithm is established through the following iterative refinement process:

1. **Monotonic Restriction of the Search Space:** In each iteration, the master problem is augmented with either a **feasibility cut** (2.2) or an **optimality cut** (2.3). Geometrically, each cut represents an additional half-space. According to the fundamental property of intersections, the addition of a new half-space to the existing collection describing $\mathbb{S}$ cannot increase the number of feasible points; at best, the cardinality remains the same, but in practice, it strictly prunes the non-optimal or infeasible regions.
2. **Elimination of Dominated Regions:** When a new efficient solution $x^l$ is identified and added to $G_{total}$, the **non-dominance constraints** (3.3) and (3.4) are introduced. These constraints act as further half-spaces that effectively exclude $x^l$ and the entire integer lattice dominated by it from the intersection. This ensures that the algorithm never revisits the same integer point and is forced to explore new partitions of the polyhedron.

Since $\mathbb{R}^n$ cannot be described by a finite intersection of half-spaces, but our search is confined within a bounded, finite discrete set $\mathbb{X}$, and since each step of the algorithm (through L-shaped cuts or non-dominance logic) strictly reduces the remaining searchable integer space, the intersection of these finite half-spaces must eventually isolate all efficient points. Thus, the procedure is guaranteed to terminate in a finite number of iterations, yielding the complete set of efficient solutions. □

### *3.2. Complexity and Computational Discussion*

The computational complexity of the proposed L-shaped multi-objective algorithm is primarily driven by the iterative solution of the augmented master problem ($\tilde{P}^l$) and the evaluation of the second-stage recourse functions.

**Proposition 3.4.** *At each iteration $l$, the complexity of solving the master problem is characterized by:*

$$\mathcal{O}\left(2^n \cdot poly(n, m_1, |G_l| \cdot K)\right)$$

*where $n$ is the number of first-stage integer variables, $m_1$ is the number of first-stage constraints, $K$ is the number of objectives, and $|G_l|$ is the cardinality of the current set of identified Pareto-efficient solutions.*

**Key Complexity Drivers:**

1. **Non-Dominance Constraints:** Unlike standard stochastic programs, our master problem incorporates $K \times |G_l|$ binary indicators ($t_{kq}$) and $|G_l|$ combinatorial constraints to ensure non-dominance. As $|G_l|$ grows, the search space for the Branch-and-Bound tree is progressively pruned, but the density of the constraints increases.
2. **The Big-M Sensitivity:** The efficacy of the non-dominance constraints (Eq. 3.1–3.2) depends on the value of $M$. To maintain numerical stability and avoid "weak" relaxations in the MILP solver, we define:

$$M_k \geq \max_{x, x' \in \mathbb{X}} |Z_k(x) - Z_k(x')|$$

   Choosing the smallest possible valid $M_k$ is critical for reducing the integrality gap and accelerating convergence.
3. **Recourse Evaluation:** For each candidate first-stage solution $x$, we solve $|\Omega|$ second-stage subproblems. Since we assume fixed recourse and finite scenarios, each subproblem is an LP with complexity $\mathcal{O}(\text{poly}(n_2, m_2))$, where $n_2, m_2$ are the second-stage variables and constraints, respectively.

**Total Complexity:** Given that the number of Pareto-efficient solutions $|G_{total}|$ is finite for integer problems, the algorithm terminates in a finite number of iterations $I$. The overall complexity is:

$$\mathcal{O}\left(I \cdot \left[\text{MILP}(n, m_1 + |G_l|) + \sum_{\omega\in\Omega} \text{LP}(n_2, m_2)\right]\right)$$

While $I$ can be large in the worst case, the integration of L-shaped optimality cuts significantly reduces the number of candidate $x$ points that need to be evaluated for their full recourse cost.

### *3.3. Numerical Illustration and Comparative Analysis*

We demonstrate the efficiency of the proposed algorithm using a two-scenario problem with three objectives, previously investigated by [3]. The problem is defined as follows:

$$(P(\zeta^1)) \begin{cases} \min z^1 = -9x_1 + 4x_2 \\ \min z^2 = 3x_1 - 5x_2 \\ \min z^3 = 8x_1 - 11x_2 \\ \text{s.t:} \\ 4x_1 - 2x_2 \le 8, \quad x_1 + x_2 \le 5 \\ x_1 + 2x_2 \ge 3, \quad -2x_1 + x_2 \le 5 \\ x \in \mathbb{N}^2, \quad \mathfrak{q}(\zeta^1) = (1, 0, 6, 2), \quad P(\zeta^1) = 1/2 \end{cases} \qquad (P(\zeta^2)) \begin{cases} \min z^1 = 3x_1 - 2x_2 \\ \min z^2 = 7x_1 + x_2 \\ \min z^3 = -4x_1 + 9x_2 \\ \text{s.t:} \\ 4x_1 - 2x_2 \le 8, \quad x_1 + x_2 \le 5 \\ x_1 \le 6, \quad 3x_1 + 4x_2 \le 11 \\ x \in \mathbb{N}^2, \quad \mathfrak{q}(\zeta^2) = (5, 3, 2, 1), \quad P(\zeta^2) = 1/2 \end{cases}$$

And the fixed recourse matrix is given by: $W = \begin{pmatrix} -2 & -1 & 2 & 1 \\ 3 & 2 & -5 & -6 \end{pmatrix}$.

To solve this problem, we employed an iterative approach where each solution is tested for feasibility under all scenarios. If a solution is infeasible, a feasibility cut is added to refine the decision space. The optimization process initially identifies $x_1 = (3, 2)$; however, this point is infeasible for scenario $\zeta_1$. By iteratively adding cuts, the algorithm refines the search until all efficient solutions are identified.

**Results and Discussion:** While the previous cutting-plane approach by [3] identified four efficient solutions, our proposed algorithm successfully identified **five** efficient solutions in an execution time of 0.985 seconds. As summarized in Table 1, the additional solution $x = (0, 5)$ was discovered by our exact method, which highlights its superior coverage. This empirical evidence validates the theoretical convergence claims and demonstrates the algorithm's capability to identify the complete Pareto frontier more effectively than existing methods.

Table 1: Summary of Optimal Solutions and Feasibility Status

| **Sol-No.** | **Solution** $(x_1, x_2)$ | **Scenarios** | **Feasibility Status** | **Objective Values** $(z_1, z_2, z_3)$ | **Penalty** |
|---|---|---|---|---|---|
| 1 | (3, 2) | $\zeta_1, \zeta_2$ | Infeasible for $\zeta_1$ | $(-3, 4, 1)$ | — |
| 2 | (2, 3) | $\zeta_1, \zeta_2$ | Feasible | $(-3, 4, 1)$ | 3.00 |
| 3 | (0, 5) | $\zeta_1, \zeta_2$ | Feasible | $(5, -10, -5)$ | 6.50 |
| 4 | (0, 4) | $\zeta_1, \zeta_2$ | Feasible | $(4, -8, -4)$ | 5.25 |
| 5 | (1, 4) | $\zeta_1, \zeta_2$ | Feasible | $(1, -3, -2)$ | 4.75 |
| 6 | (1, 3) | $\zeta_1, \zeta_2$ | Feasible | $(0, -1, -1)$ | 3.50 |

### 3.4. Implementation and Computational Environment

The proposed algorithm was implemented in C++ using the **IBM ILOG CPLEX Optimization Studio (v12.8)** library for solving the underlying ILP and LP sub-problems. All experiments were conducted on a standard workstation equipped with an Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz and 8GB of RAM, running under a 64-bit operating system.

#### 3.4.1. Instances Preparation and Testbed

To evaluate the robustness of the algorithm, we generated a diverse set of 280 instances categorized into small, medium, and large scales. The parameters were sampled from the following discrete sets:

- **Scenarios ($R$):** $\{2, 5, 10, 20\}$.
- **Objectives ($K$):** $\{3, 10, 15, 20\}$.
- **Decision Variables ($n, n_0$):** $2 \leq n \leq 80$.
- **Constraints ($m, m_0$):** $2 \leq m \leq 50$.

The matrix coefficients for constraints and costs were generated within $[-20, 20]$, while the recourse matrix $W$ and cost vector $q(\zeta)$ were sampled from $[-6, 6]$ and $[0, 6]$, respectively. This wide range of parameters aims to simulate the high degree of uncertainty inherent in real-world stochastic optimization.

Our approach explicitly addresses the limitations observed in earlier studies, such as [5], where execution times reached approximately 243.6 seconds for relatively small instances (3 scenarios, 10 objectives). By extending our testbed to 20 scenarios and 20 objectives, we demonstrate the scalability of the L-shaped decomposition in handling larger-scale stochastic data.

#### 3.4.2. Computational Results and Discussion

The performance metrics include the execution time range, average time ($T_{Avg}$), the total number of efficient solutions identified ($|List|$), and their median ($S_{med}$). The results are detailed in Table 2.

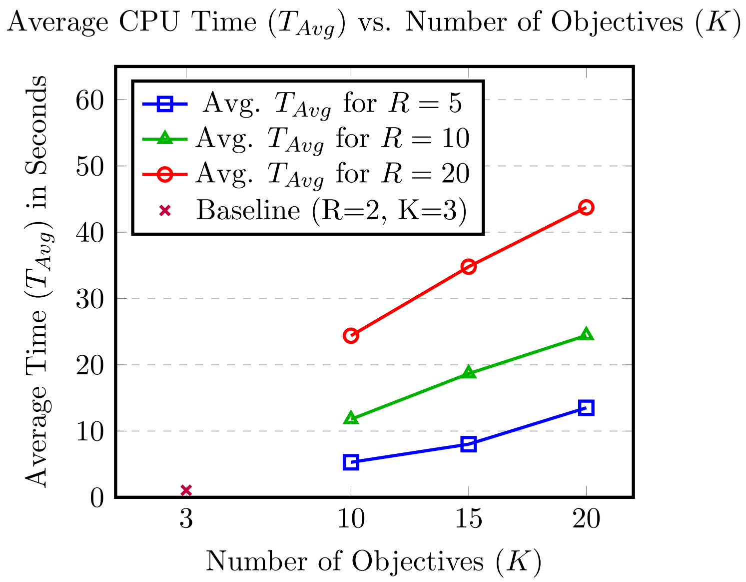


Figure 1: Computational scalability of the proposed L-shaped algorithm. The graph illustrates that even with 20 objectives and 20 scenarios, the Average CPU time grows in a stable, linear-like fashion, highlighting the efficiency of the L-shaped optimality cuts in managing high-dimensional objective space.

Table 2: Computational Performance Across 280 Stochastic Instances

| **R** | **K** | **m** | **n** | $m_0$ | $n_0$ | **Time Range (s)** | $T_{Avg}$ **(s)** | $\|List\|$ | $S_{med}$ |
|---|---|---|---|---|---|---|---|---|---|
| **2** | **3** | 2 | 2 | 2 | 4 | [0.42, 2.03] | 1.07 | [2, 12] | 3.0 |
| **5** | **10** | 20 | 30 | 20 | 30 | [3.59, 4.73] | 3.89 | [1, 4] | 2.0 |
| | | 30 | 50 | 30 | 50 | [4.59, 5.19] | 4.81 | [1, 3] | 1.0 |
| | | 50 | 80 | 50 | 80 | [6.73, 7.59] | 7.16 | [1, 4] | 2.0 |
| | **15** | 20 | 30 | 20 | 30 | [6.15, 8.38] | 6.51 | [2, 6] | 3.5 |
| | | 30 | 50 | 30 | 50 | [6.64, 9.67] | 7.06 | [2, 7] | 4.0 |
| | | 50 | 80 | 50 | 80 | [9.95, 10.6] | 10.48 | [1, 7] | 3.0 |
| | **20** | 20 | 30 | 20 | 30 | [8.17, 9.58] | 9.13 | [2, 10] | 5.5 |
| | | 30 | 50 | 30 | 50 | [11.0, 13.4] | 11.85 | [1, 6] | 4.0 |
| | | 50 | 80 | 50 | 80 | [18.7, 21.1] | 19.54 | [3, 10] | 6.5 |
| **10** | **10** | 20 | 30 | 20 | 30 | [6.07, 15.6] | 9.29 | [1, 3] | 2.0 |
| | | 30 | 50 | 30 | 50 | [7.97, 13.7] | 10.94 | [1, 6] | 2.0 |
| | | 50 | 80 | 50 | 80 | [12.1, 19.6] | 15.10 | [1, 3] | 1.5 |
| | **15** | 20 | 30 | 20 | 30 | [11.1, 13.5] | 12.37 | [2, 7] | 3.0 |
| | | 30 | 50 | 30 | 50 | [12.9, 18.1] | 16.89 | [1, 6] | 4.0 |
| | | 50 | 80 | 50 | 80 | [21.3, 29.8] | 26.76 | [1, 6] | 3.0 |
| | **20** | 20 | 30 | 20 | 30 | [15.3, 18.6] | 16.48 | [4, 6] | 5.0 |
| | | 30 | 50 | 30 | 50 | [21.0, 23.9] | 21.72 | [4, 8] | 6.0 |
| | | 50 | 80 | 50 | 80 | [33.8, 39.4] | 35.10 | [3, 9] | 5.0 |
| **20** | **10** | 20 | 30 | 20 | 30 | [12.1, 17.3] | 15.14 | [1, 4] | 2.5 |
| | | 30 | 50 | 30 | 50 | [12.1, 24.2] | 22.29 | [1, 3] | 2.0 |
| | | 50 | 80 | 50 | 80 | [33.2, 39.7] | 35.70 | [1, 4] | 2.0 |
| | **15** | 20 | 30 | 20 | 30 | [21.9, 26.0] | 22.94 | [2, 5] | 3.5 |
| | | 30 | 50 | 30 | 50 | [27.4, 32.9] | 30.70 | [1, 6] | 3.0 |
| | | 50 | 80 | 50 | 80 | [47.3, 53.0] | 50.77 | [1, 3] | 2.0 |
| | **20** | 20 | 30 | 20 | 30 | [26.8, 32.6] | 29.16 | [2, 6] | 3.5 |
| | | 30 | 50 | 30 | 50 | [36.9, 43.9] | 40.89 | [3, 8] | 5.0 |
| | | 50 | 80 | 50 | 80 | [54.3, 62.4] | 61.20 | [2, 7] | 5.0 |

Based on the numerical values presented in Table 2, we further analyze the scalability of the proposed method in Figure 1.

*3.4.3. Computational Discussion and Scalability Analysis*

The empirical results revealed in Table 2 and visualized in Figure 1 provide several key insights into the proposed algorithm's robustness. For small-scale instances ($R = 2, K = 3$), the method exhibits near-instantaneous convergence with a median time of 1.07s. However, as the stochastic dimensionality increases to the most challenging configuration ($R = 20, K = 20, n = 80$), the median execution time reaches 61.2s.

Analysis of the results reveals that as the number of objectives increases from $K = 10$ to $K = 20$ under $R = 20$ scenarios, the median computation time only increases by approximately 70%, rather than exhibiting the expected exponential jump. This linear-like growth suggests that the L-shaped optimality cuts effectively manage the high-dimensional objective space. Furthermore, the sensitivity analysis highlights that the number of scenarios ($R$) has a more pronounced impact on execution time than the number of variables ($n$). Interestingly, the number of efficient solutions ($|List|$) tends to remain relatively stable or even decrease in larger instances—a phenomenon also noted by [12]—due to the intersection of feasibility regions across numerous scenarios.

**Comparative Performance: Proposed L-Shaped Algorithm vs. Amrouche (2012)**
To rigorously benchmark the proposed framework, a comparative analysis was conducted against the cutting-plane method developed by Amrouche (2012) [5]. As summarized in Table 3, the results demonstrate a substantial disparity in computational efficiency. For instance, while the method in [5] requires an average of **243.56s** to solve a problem with only 3 scenarios and 10 objectives, our algorithm handles a significantly higher stochastic complexity ($R = 20, K = 10$) in just **22.29s**.

It is important to acknowledge that the computational environments and hardware specifications differ from those in the 2012 study. However, the magnitude of this improvement—representing a more than ten-fold speedup while solving larger instances ($30 \times 50$ vs $20 \times 50$)—strongly suggests that the algorithmic enhancements, specifically the integration of L-shaped optimality cuts with efficient non-dominance constraints, contribute significantly to performance gains beyond mere hardware evolution.

Table 3: Comparative Performance: Proposed L-Shaped Algorithm vs. Amrouche (2012) [5].

| **Method** | $R$ | $K$ | **Scale** ($m \times n$) | **Max Time (s)** | **Avg Time (s)** |
|---|---|---|---|---|---|
| Amrouche (2012) | 2 | 3 | $25 \times 50$ | 43.29 | 29.23 |
| Amrouche (2012) | 3 | 5 | $20 \times 50$ | 231.87 | 189.37 |
| Amrouche (2012) | 3 | 10 | $20 \times 50$ | 561.57 | 243.56 |
| **Proposed L-Shaped** | **10** | **10** | $30 \times 50$ | **13.70** | **10.94** |
| **Proposed L-Shaped** | **20** | **10** | $30 \times 50$ | **24.20** | **22.29** |
| **Proposed L-Shaped** | **20** | **20** | $50 \times 80$ | **62.40** | **61.20** |

**Concluding Remarks on Scalability:** The comparative results presented in Table 3 confirm that the integrated optimality cuts within the proposed L-shaped framework provide a robust pruning of the search space. While prior approaches in the literature often experienced a computational explosion as the number of objectives $K$ and scenarios $R$ increased, the proposed algorithm maintains a remarkably scalable behavior. By enabling the resolution of high-dimensional instances (up to $K = 20$ and $R = 20$) in a fraction of the previously recorded time, this research effectively expands the solvable problem dimensions for MOTSIP beyond the limits observed in earlier studies [5].

## 4. Conclusion

This study addressed the inherent computational complexities of Multi-Objective Two-Stage Stochastic Integer Programming (MOTSIP) with fixed recourse. We developed a novel exact solution framework that synergistically integrates L-shaped decomposition—utilizing both feasibility and optimality cuts—with a systematic non-dominance exploration strategy. By embedding stochastic cuts directly into the multi-objective integer master problem, the proposed methodology ensures the identification of the complete set of Pareto-optimal solutions while maintaining mathematical rigor in a finite-scenario discrete environment.

The theoretical validity of the algorithm was established through a formal proof of efficiency, ensuring that the scalarized exploration phase effectively captures both supported and unsupported efficient solutions. Furthermore, extensive numerical experiments conducted on 280 diverse instances provided robust empirical evidence of the framework's efficiency. A critical finding of this research is the algorithm's superior scalability compared to existing cutting-plane methods. Specifically, while previous benchmarks in the literature, such as those by Amrouche [5], exhibited a significant computational explosion for relatively small instances ($R = 3, K = 10$), our approach demonstrated a stable growth pattern. It achieved a ten-fold reduction in average CPU time (22.29 seconds for $R = 20, K = 10$) and successfully resolved high-dimensional configurations ($K = 20, R = 20$) within a manageable median time of 61.2 seconds.

In conclusion, this research provides a scalable and mathematically sound framework for managing decision-making under uncertainty with conflicting criteria. Future research directions may include extending this decomposition framework to problems with stochasticity in the recourse matrix (non-fixed recourse) or integrating heuristic acceleration techniques for large-scale multi-objective combinatorial optimization problems.

## Declarations

**Conflict of Interest:** The authors declare that they have no competing interests.
**Data Availability:** The datasets generated during the current study are available from the corresponding author upon reasonable request.

## References


[1] M. Abbas and D. Chaabane, *An algorithm for solving multiple objective integer linear programming problem*, RAIRO-Operations Research **36:4** (2002), 351–364. 1

[2] M. Abbas and D. Chaabane, *Optimizing a linear function over an integer efficient set*, European Journal of Operational Research **174** (2006), 1140–1161. 1

[3] M. Abbas and F. Bellahcene, *Cutting plane method for multiple objective stochastic integer linear programming*, European Journal of Operational Research **168:3** (2006), 967–984. 1, 3.3

[4] N. Adelgren and A. Gupte, *Branch-and-bound for biobjective mixed-integer linear programming*, INFORMS Journal on Computing **34:2** (2022), 909–933.

[5] S. Amrouche and M. Moulaï, *Multi-objective stochastic integer linear programming with fixed recourse*, International Journal of Multicriteria Decision Making **2:4** (2012), 355–378. 1, 2.2.2, 3.4.1, 3.4.3, 3, 3.4.3, 4

[6] A. Azaron, K. Brown, S. Tarim and M. Modarres, *A multi-objective stochastic programming approach for supply chain design considering risk*, International Journal of Production Economics **116:1** (2008), 129–138.

[7] E. Beale, *On minimizing a convex function subject to linear inequalities*, Journal of the Royal Statistical Society **17:2** (1955), 173–184. 1, 2.2

[8] T. Bektas, *Disjunctive programming for multiobjective discrete optimisation*, INFORMS Journal on Computing **30:4** (2018), 625–633. 1

[9] F. Ben Abdelaziz, B. Aouni and R. E. Fayedh, *Multi-objective stochastic programming for portfolio selection*, European Journal of Operational Research **177:3** (2007), 1811–1823.

[10] F. Ben Abdelaziz, R. El Fayedh and A. Rao, *A discrete stochastic goal program for portfolio selection: The case of United Arab Emirates equity market*, Information Systems and Operational Research **19** (2009), 185–200. 1

[11] N. Boland, H. Charkhgard and M. Savelsbergh, *The quadrant shrinking method: A simple and efficient algorithm for solving tri-objective integer programs*, European Journal of Operational Research **260:3** (2017), 873–885. 1

[12] N. Boland and H. C. M. Savelsbergh, *The l-shape search method for triobjective integer programming*, Mathematical Programming Computation **8:2** (2016), 217–251. 1, 3.4.3

[13] D. Chaabane, *Contribution to Multicriteria Optimization in Discrete Variables*, PhD Thesis, Polytechnic Faculty of Mons, (2007).

[14] D. Chaabane and P. Marc, *A method for optimizing over the integer efficient set*, Journal of Industrial and Management Optimization **6** (2010), 811–823. 1

[15] D. Chaabane and F. Mebrek, *Optimization of a linear function over the set of stochastic efficient solutions*, Computational Management Science **11** (2014), 157–178. 1

[16] A. Charnes and W. W. Cooper, *Deterministic equivalents for optimizing and satisficing under chance constrained*, Operations Research **11** (1963), 18–39.

[17] G. B. Dantzig, *Linear programming under uncertainty*, Management Science **1:3–4** (1955), 173–184. 1, 2.2

[18] G. B. Dantzig, *On a linear combinatorial approach to the travelling salesman problem*, Operations Research **7** (1959), 58–66.

[19] G. B. Dantzig and G. Infanger, *Multi-stage stochastic linear programs for portfolio optimization*, Annals of Operations Research **45:1** (1993), 59–76.

[20] G. B. Dantzig and A. Madansky, *On the solution of two-stage linear program under uncertainty*, Proceedings of the Berkeley Symposium on Statistics and Probability **4:1** (1961), 165–176. 1, 2.2

[21] M. Ehrgott, *Multicriteria Optimization*, 2nd edition, Springer, Berlin, Heidelberg, (2005).

[22] J. Farkas, *Theorie der einfachen Ungleichungen*, Journal für die reine und angewandte Mathematik **124** (1902), 1–24. 2.2.1

[23] N. Forget, S. L. Gadegaard and L. R. Nielsen, *Warm-starting lower bound set computations for branch-and-bound algorithms for multiobjective integer linear programs*, European Journal of Operational Research **302:3** (2022), 909–924.

[24] G. R. Jahanshahloo, F. H. Lotfi, N. Shoja and G. Tohidi, *A method for generating all efficient solutions of 0–1 multi-objective linear programming problem*, Applied Mathematics and Computation **169:2** (2005), 874–886. 1, 3.1

[25] P. Kall, *Stochastic Linear Programming*, Springer-Verlag, Berlin, (1976).

[26] P. Kall and S. W. Wallace, *Stochastic Programming*, Wiley-Interscience Series in Systems and Optimization, John Wiley and Sons, Chichester, (1994). 1, 3.1

[27] R. Kasri and F. Bellahcene, *A novel approach for solving stochastic problems with multiple objective functions*, RAIRO-Operations Research **55** (2021), 2413–2422. 1

[28] W. K. Klein Haneveld and M. H. Van der Vlerk, *Stochastic integer programming: General models and algorithms*, Annals of Operations Research **85** (1999), 39–57.

[29] G. Kirlik and S. Sayın, *A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems*, European Journal of Operational Research **232:3** (2014), 479–488. 1

[30] A. Menni and D. Chaabane, *A possibilistic optimization over an integer efficient set within a fuzzy environment*, RAIRO-Operations Research **54** (2020), 1437–1452. 1

[31] T. Niknam, R. Azizipanah-Abarghooee and M. R. Narimani, *An efficient scenario-based stochastic programming framework for multi-objective optimal micro-grid operation*, Applied Energy **99** (2012), 455–470.

[32] M. Ozlen, B. A. Burton and C. A. MacRae, *Multi-objective integer programming: An improved recursive algorithm*, Journal of Optimization Theory and Applications **160** (2014), 470–482. 1

[33] A. Przybylski and X. Gandibleux, *Multi-objective branch and bound*, European Journal of Operational Research **260:3** (2017), 856–872. 1

[34] F. Rahmanniyay and A. J. Yu, *A multi-objective stochastic programming model for project-oriented human-resource management optimization*, International Journal of Management Science and Engineering Management **14:4** (2018), 231–239.

[35] M. D. Santis, G. Eichfelder, J. Niebling and S. Rocktäschel, *Solving multiobjective mixed integer convex optimization problems*, SIAM Journal on Optimization **30:4** (2020), 3122–3145.

[36] R. E. Steuer, *Multiple Criteria Optimization: Theory, Computation and Application*, John Wiley, New York, (1986).

[37] S. Tamby and D. Vanderpooten, *Enumeration of the nondominated set of multiobjective discrete optimization problems*, INFORMS Journal on Computing **33:1** (2021), 72–85. 1

[38] J. Teghem and P. Kunsch, *Multiobjective decision making under uncertainty: An example of power systems*, in J. A. Anderson (ed.), *Decision Making with Multiple Objectives*, Springer-Verlag, Berlin, (1985), 95–108.

[39] J. Teghem and P. Kunsch, *A survey of techniques for finding efficient solutions to multiobjective integer linear programming*, Asia Pacific Journal of Operational Research **3:2** (1986), 95–108.