N° d'ordre: 267|2025 – *C|MT*

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Sciences and Technology Houari Boumediene USTHB

Faculty of Mathematics

Doctoral Thesis

Submitted to obtain the degree of doctor
In Applied Mathematics
Specialty: Stochastic Optimization
Presented by:

Ilias BADAOUI

Title :

Bi-objective Stochastic Optimization over the Efficient Set of a Multi-Objective Stochastic Integer Linear Problem

Defended on February 27^{th}, 2025

Jury members:

Mr. AÏDER Méziane,	Professor at USTHB	President
Mr. MOULAÏ Mustapha,	Professor at USTHB	Supervisor
Mr. BOUZID Mouaouia Cherif,	Associate Professor/A at ENSTA	Reviewer
Mr. OUANES Mohand,	Professor at UMMTO	Reviewer
Mrs. DAHMANI Isma,	Associate Professor/A at USTHB	Reviewer
Mr. CHAABANE Djamal,	Professor at USTHB	Invited

Many thanks

First of all, I would like to thank the Almighty and Clement Allah for the blessings and strength He has given me throughout my academic career. In Him, I have found the perseverance and wisdom necessary to overcome the challenges encountered during this research.

I would like to express my deep gratitude to my thesis supervisor, MOULAÏ Mustapha, for his invaluable support, patience and expert knowledge which have guided this work.

I would like to express my deep gratitude to Professor, Mahdi BASHIRI, for his invaluable support, and help during my stay at the CBiS center, at Coventry University.

I would like to express my deep gratitude to Dr. Oussama Gacem, Dr Nedjmeddine Kantour, and Dr Yacine Chaibelainne for their invaluable collaboration.

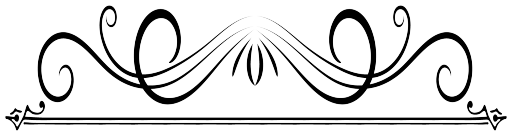
I would like to thank Heather Parker, and Sami Mehdaoui for their help during my stay at Coventry.

I want to thank my fellow PhD students for their camaraderie and support along the way, particularly Ali Benchikh, Ahmed Tobal, Bachir Mohamed, Belkacem, Charef, Salim, Sofiane, Hakim, Sami for their friendship.

Finally, I would like to thank my family and friends for their unconditional love, encouragement, and faith in me, which have been a constant source of strength and inspiration.

This thesis is the fruit of a collective effort, and I am honored to share its achievements with all those who contributed to its realization.

Dedication



To my dear parents, whose unwavering support, caring upbringing and prayers have been the beacon of my educational journey. Your patience and love have paved the way for my success.

To my brothers, Abd Elkarim, Rabeh, Amine Radoun, Ahmed, Ayoub Belkacem for their constant encouragement and faith in my abilities.

To my dear friends Gacem Oussama, Ali Benchikh, Yacine Chaibelainne, Ahmed Tobal, Sadak Nasri, Salah Bouagada, Ouadani Mohamed, Boumaad Oussama, Hadbi Youcef, Lamraoui Boualem, Djafri Mourad, Bouchile Ramzi, Mehaoui Sami, Abd Elkader Bouchaalla, Islem Benmoussa, Omar, Ouled Sidieamer, and Amer Rahmouni for their bits of help during my university time.

This thesis is dedicated to all of you, for all that you have sacrificed, and for the richness that you have brought to my life.

Résumé

La programmation multi-objectifs joue un rôle très important dans la résolution des problèmes de la vie réelle, car les décideurs ont souvent plusieurs fonctions objectives à optimiser, et le but est de trouver des solutions dites efficaces. Cependant, le nombre de ces solutions peut être considérable, et les trouver toutes peut impliquer des coûts de computation élevés. De plus, le décideur peut éprouver des difficultés à choisir une solution parmi elles. Pour contourner ce problème, nous pouvons considérer une nouvelle fonction à optimiser sur les solutions efficaces. Cette approche est appelée optimisation sur l'ensemble efficace. Plusieurs méthodes ont été proposées dans la littérature pour résoudre ce type de problème, essentiellement dans le cas linéaire, dans un environnement déterministe. Néanmoins, de nombreux problèmes du monde réel sont modélisés par des fonctions linéaires dans un environnement stochastique. La variante du problème de l'optimisation sur l'ensemble efficace dans l'environnement stochastique n'a pas reçu beaucoup d'attention. La présente thèse aborde le défi de l'optimisation sur l'ensemble efficace dans l'environnement stochastique, où nous contribuons à la résolution d'un problème d'optimisation d'une fonction linéaire stochastique sur l'ensemble efficace d'un problème de programmation linéaire stochastique en nombres entiers multi-objectif (*MOSILP*). Nous proposons également de résoudre un problème où deux décideurs ont chacun une fonction linéaire stochastique à optimiser sur l'ensemble efficace d'un problème de *MOSILP*.

Mots-clés: Programmation multi-objectifs, Optimisation sur un ensemble efficace, Optimisation stochastique, Optimisation stochastique en deux étapes, L-shaped méthode.

Abstract

Multi-objective programming plays a very important role in real-life problem solving, as decision-makers often have multiple objective functions to optimize, and the aim is to find solutions that are said to be efficient. However, the number of such solutions can be considerable, and finding them all can entail high computational costs. Moreover, the decision-maker may find it difficult to choose a solution from among them. To overcome this problem, we can consider a new function to be optimized over the efficient solutions. This approach is called optimization over the efficient set. Various methods have been proposed in the literature to solve this type of problem, essentially in the linear case, in a deterministic environment. Nevertheless, many real-world problems are modeled by linear functions in the stochastic environment. The variant of the problem of optimization over the efficient set in the stochastic environment has not received much attention. The present thesis addresses the challenge of optimization over the efficient set in the stochastic environment, where we contribute to resolving the problem of optimizing a stochastic linear function over the efficient set of a multi-objective stochastic integer linear programming (*MOSILP*) problem. We also propose to solve a problem where two decision-makers each have a stochastic linear function to optimize over the efficient set of a *MOSILP* problem.

Keywords: Multi-objective programming, Optimization over efficient set, Stochastic optimization, Two-stage stochastic optimization, L-shaped method.

مُلَخَّص

تلعب البرمجة متعددة الأهداف دورًا مهمًا جدًا في حل المشكلات في الحياة الواقعية، حيث أن صانعي القرار غالبًا ما يكون لديهم وظائف أهداف متعددة لتحسينها، والهدف هو إيجاد حلول يقال إنها فعالة. ومع ذلك، قد يكون عدد هذه الحلول كبيراً، وقد يستلزم إيجادها جميعاً تكاليف حسابية عالية. علاوة على ذلك، قد يجد صانع القرار صعوبة في اختيار حل من بينها. للتغلب على هذه المشكلة، يمكننا التفكير في دالة جديدة يتم تحسينها على الحلول الفعالة. يُطلق على هذا النهج اسم التحسين على المجموعة الفعالة. وقد تم اقتراح طرق مختلفة في الأدبيات لحل هذا النوع من المشاكل، بشكل أساسي في الحالة الخطية، في بيئة حتمية. ومع ذلك، يتم تمثيل العديد من المشاكل في العالم الحقيقي بواسطة دوال خطية في بيئة عشوائية. لم يحظَ المتغير الخاص بمشكلة التحسين على المجموعة الفعالة في البيئة العشوائية باهتمام كبير. تتناول هذه الأطروحة الحالية تحدي التحسين على المجموعة الفعالة في البيئة العشوائية، حيث نساهم في حل مشكلة تحسين خطية خطية على المجموعة الفعالة للبرمجة الخطية الصحيحة متعددة الأهداف العشوائية متعددة الأهداف *MOSILP* . نقترح أيضًا حل مشكلة حيث يكون لكل من صانعي القرار دالة خطية عشوائية عشوائية لتحسينها على المجموعة الفعالة لمشكلة برمجة خطية متعددة الأهداف متعددة الأهداف *MOSILP* .

الكلمات الرئيسية : البرمجة متعددة الأهداف، التحسين على مجموعة فعالة، التحسين العشوائي، التحسين العشوائي متعدد الأهداف، التحسين العشوائي على مرحلتين، طريقة على شكل حرف L .

Contents

List of Figures

List of Tables

List of Algorithms

List of abbreviations

LP	Linear Programming.
ILP	Integre Linear Programming.
CLP	Continuous Linear Programming.
MILP	Mixte Integer Linear Programming.
MOP	Multi-Objective programmin.
MOIP	Multi-Objective Integer Programming.
MOLP	Multi-Objective Linear Programming.
MOILP	Multi-Objective Integer Linear Programming.
MOMILP	Multi-Objective Mixte Integer Linear Programming.
BOILP	Bi-objective Integer Linear Programming.
BOMILP	Bi-objective Mixed Integer Linear Programs.
MOLFP	Multi-objective Linear Fractional Programming.
MOILFP	Multi-objective Integer Linear Fractional Programming.
BOLF/MOILP	Bi-objective optimization over a Multi-Objective Integer Linear Programming.
SP/SLP	Stochastic Programming / Stochastic Linear Programming.
MOSP	Multi-Objective Stochastic Programming.
MOSLP	Multi-Objective Stochastic Linear Programming.
MOSILP	Multi-Objective Stochastic Integer Linear Programming.
LF/MOSILP	Linear optimization over a Multi-Objective Stochastic Integer Linear Programming.
BSLF/MOSILP	Bi-objective Stochastic Linear optimization over a Multi-Objective Stochastic Integer Linear Programming.
B & B	Branch and Bound.
B & C	Branch and Cut.
WS	Weighte Sum Method.
EC	Epsilon Constraint Method.
CC	Chance Constraints Model.
CCM	Chaabane and Mabrek Method.
BCM	A Branch and Cut based Method.

PART

I

BACKGROUNDS AND PRELIMINARIES

Introduction

> *"The seeker after the truth does not study the writings of the ancients and, following his natural disposition, puts his trust in them, but rather the one who suspects his faith in them and questions what he gathers from them, the one who submits to argument and demonstration, and not to the sayings of a human being whose nature is fraught with all kinds of imperfection and deficiency. Thus the duty of the man who investigates the writings of scientists, if learning the truth is his goal, is to make himself an enemy of all that he reads, and, apply his mind to the core and margins of its content."* **Ibn al-Haytham (Alhazen).**

Life, in all its complexity, holds both challenges and unique beauty. This complexity is reflected in the daily choices individuals, groups, and institutions face, as they strive to make the best decisions when dealing with multiple, often conflicting, objectives. Most everyday decisions are made based on intuition, common sense, chance, or a mix of these. It's natural to want these choices to be as good as possible—or optimal. However, in fields like engineering and economics, mathematical modeling and programming are often required. This is where "multi-objective optimization" comes in, aiming to find a set of solutions that meet these different goals. Many scientists and experts have studied these challenges, developing mathematical models and proposing various methods and algorithms to solve them, leading to what we call "efficient solutions. Finding efficient solutions is relatively straightforward when objectives are modeled by linear functions and decision variables are continuous. However, many real-world applications in engineering, operations research, and the sciences involve discrete or mixed decision variables (both discrete and continuous), as well as deterministic or stochastic variables. This adds significant complexity to the problem-solving process. Additionally, the large number of potential solutions can be overwhelming for decision-makers, who must choose the one that best aligns with their preferences. To address this challenge, researchers have developed approaches that help model decision-maker preferences and select the optimal solution from the set of efficient ones. This approach is commonly referred to in the literature as "optimization over the efficient set."

It is in this perspective that our contribution to the research falls, tackling the problem of 'optimization over the efficient set of a stochastic multi-objective problem'. In the course of this thesis, we propose and develop algorithms dedicated to solving this type of problem through 2 publications, one of which is currently being revised (at the time of writing).

Organization of the thesis

This thesis is divided into two main parts. The first part establishes the theoretical framework and fundamental concepts needed to understand the research work described in the second part.

First part

- **Chapter 1: Multi-objective Integer Linear Programming**
  This chapter introduces some fundamental notations and definitions of linear programming (LP). In addition, it provides a general introduction to multi-objective integer linear programming (MOILP). The chapter then presents methods for solving different classes of MOILP problems, as well as optimization techniques over the efficient set of these problems.

- **Chapter 2: Multi-objective Stochastic Integer Linear Programming**
  This chapter provides a general overview of stochastic linear programming (SLP) along with multi-objective stochastic integer programming (MOSILP). In addition, It presents solution methods for different classes of MOSILP problems, as well as optimization techniques over the efficient set of these problems.

Second part

The second part forms the main core of this thesis and presents our contributions and research to the optimization over the efficient set of a MOSILP problem.

- **chapter 3: Optimizing a linear function over the stochastic efficient set**
  An original algorithm is presented in this chapter for optimizing a linear function over the efficient set of a MOSILP problem (LF/MOSILP). Moreover, this chapter presents the theoretical foundations of the algorithm, an illustrative application, and the results of numerical experiments.

- **chapter 4: Biobjective integer stochastic optimization over the integer stochastic efficient set [Badaoui et al., 2024]**
  This chapter extends the work of the previous chapter by considering the simultaneous optimization of two linear stochastic functions over the efficient set of a MOSILP problem BSLF/MOSILP.

Chapter

1 Multi-objective Linear Programming

"Life is an optimization problem."
"La vie est un problème d'optimisation."
Anonyme

Vangelis Th. PASCHOS.

1.1 Introduction

Multi-objective programming (MOP) is rooted in Edgeworth and Pareto's economic work [Ehrgott, 2012]. It was first used in economics and management sciences, and then later in engineering sciences. Despite the importance of multi-objective modeling of problems encountered in industry, telecommunications, mechanics, etc., very few studies have been carried out on MOP. However, over the last 20 years, there has been a strong interest in multi-objective decision support. In this chapter, we introduce some basic definitions and concepts of LP [Ignizio and Cavalier, 1994, Dantzig, 2002, Karloff, 2008] and multi-objective linear programming (MOLP) [Zeleny, 2012, Cohon, 2013]. We start with an overview of LP, then give some general notions on MOLP, and finally, we present some solving methods for optimizing over an efficient set of a MOILP problem.

1.2 Linear Programming Overview

1.2.1 Formulation problem

Linear programming is a special case of mathematical programming in which the objective function and the constraints of the problem are linear. The general form of a LP problem can be given by:

$$(LP)\begin{cases} "Opt" \quad f = Cx, \\ s.t., \\ \qquad\quad Ax \leq\geq b, \\ \qquad\quad x \geq 0, \end{cases} \tag{1.1}$$

where, $A \in \mathbb{R}^{m\times n}$, $C \in \mathbb{R}^{1\times n}$, $x \in \mathbb{R}^{n\times 1}$et b $\in \mathbb{R}^{m\times 1}$. The standard form of an optimisation problem is:

$$(SLP)\begin{cases} "Opt" \quad f = Cx, \\ s.t., \\ \qquad\quad Ax = b, \\ \qquad\quad x \geq 0. \end{cases} \tag{1.2}$$

In general, the form of a uni-criteria LP problem can be given by different formulas such as:

Continuous linear programming (CLP):

The general form of a CLP problem where all variabels are continuous is given by :

$$(CLP)\begin{cases} \min \quad f = Cx, \\ s.t., \\ \qquad\quad Ax \leq b, \\ \qquad\quad x \geq 0, \end{cases} \tag{1.3}$$

where, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^{m\times 1}$ and $C \in \mathbb{R}^{1\times n}$.

Mixte Integer linear programming (MILP):

If some of the variables are integers, then we have a MILP problem which is written as follows:

$$(MILP)\begin{cases} \min \quad Cx + hy, \\ s.t., \\ \qquad\quad Ax + Gy \leq b, \\ \qquad\quad x \geq 0, \\ \qquad\quad y \in \mathbb{Z}, \end{cases} \tag{1.4}$$

where, $A \in \mathbb{R}^{m\times n1}$, $G \in \mathbb{R}^{m\times n2}$, $b \in \mathbb{R}^{m\times 1}$, $C \in \mathbb{R}^{1\times n1}$ and $h \in \mathbb{R}^{1\times n2}$.

Integer linear programming (ILP):

If all the variables are integers, then the ILP problem is given by:

$$(ILP)\begin{cases} \min \quad f = Cx, \\ s.t., \\ \qquad Ax \leq b, \\ \qquad x \geq 0, \\ \qquad x \in \mathbb{Z}. \end{cases} \tag{1.5}$$

where, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^{m\times 1}$ and $C \in \mathbb{R}^{1\times n}$.

1.2.2 Principal tools

Basic concepts of linear programming

Assume that the matrix A has rank m ($m \leq n$). Let B be a square submatrix consisting of m independent columns of A. We associate with this basis the decompositions $A = [B|N]$, $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$. I is the set of basic indices and J the set of non-basic indices.

Definition 1.2.1. *A solution $x^\star$ is said to be feasible if it verifies*

$$\begin{cases} Ax^\star = b, \\ x^\star \geq 0. \end{cases} \tag{1.6}$$

Then $x^\star$ is a feasible solution.

Definition 1.2.2. *We call* **basic variables**, *the m variables of $\mathbb{R}^n$ corresponding to the system of m independent vectors defined by the linear constraints. The remaining $(n-m)$ variables are called* **non-basic variables**. *A solution is said to be basic if it consists of basic variables.*

Note, x_B the basic variables corresponding to m positive or null variables and x_N the non-basic variables, corresponding to n null variables.

Definition 1.2.3. *A basic solution $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$ is degenerate if x_B has at least one null component.*

Decomposing the matrix A and the vector x according to the basic and non-basic variables, the problem (1.2) can be rewritten in the form

$$(P_B)\begin{cases} \min \quad f = C_B x_B + C_N x_N, \\ s.t., \qquad Bx_B + Nx_N = b, \\ \qquad\qquad x_B \;, \quad x_N \geq 0. \end{cases} \tag{1.7}$$

According to (P_B) we have :

$$Bx_B + Nx_N = b \Rightarrow x_B = B^{-1}b - B^{-1}Nx_N.$$

By substituting this expression into the objective function, we obtain:

$$f = C_B(B^{-1}b - B^{-1}Nx_N) + C_N x_N.$$

and consequently:

$$f - C_B B^{-1}b = (C_N - C_B B^{-1}N)x_N.$$

Since x_B is admissible, we obtain:

$$\begin{cases} \min \quad f - C_B B^{-1}b = (c_N - C_B B^{-1}N)x_N, \\ s.t., \\ \quad x_B = B^{-1}b - B^{-1}Nx_N \geq 0, \\ \quad x_N \geq 0, \end{cases} \tag{1.8}$$

Definition 1.2.4. *A basic solution associated with the base B is defined by:*

$$x_B = (B)^{-1}b, \; x_N = 0$$

If $x_B \geq 0$ then the basic solution is feasible.

A necessary and sufficient condition, in the absence of degeneracy, for x_B to be an optimal basic solution is that $(c_N - C_B B^{-1}N) \geq 0$.

Convexity & Polyhedron

Definition 1.2.5 (Convex Set)**.** *A set $\mathcal{S}$ is said to be convex when for all x and y of $\mathcal{S}$, the segment $[x, y]$ is entirely contained in $\mathcal{S}$, that is to say*

$$\forall\, x, y \in \mathcal{S} : \lambda x + (1 - \lambda)y \in \mathcal{S}; \forall\, \lambda \in [0, 1].$$

. if $\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_p$ are convex then their intersection is convex.

Definition 1.2.6 (Convex function)**.** *Given a convex set $\mathcal{S} \subseteq R^n$, a function $f : \mathcal{S} \longrightarrow \mathbb{R}^n$ is convex if*

$$\forall\, x, y \in \mathcal{S} : f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y); \; \forall\, \lambda \in [0, 1].$$

- *f is strictly convex if this inequality is strict.*
- *f is concave if $(-f)$ is convex.*

Definition 1.2.7 (Polyhedron)**.** *We call a polyhedron P a connected interior part (convex or concave) of real space which is the union of finite intersections of half-spaces bounded by hyperplanes H_i.*

Definition 1.2.8 (Polyhedron set). *We call a polyhedral set a set of convex or concave parts of $\mathbb{R}^d$, which is the union of several polyhedra with empty or non-empty intersections. A bounded polyhedron is called a polytope.*

The nature of LP problems in continuous variables *CLP* and LP problems in discrete variables *ILP* is different. Unlike the *CLP* problem where we are only interested in the extreme vertex solutions of the convex polyhedron, the optimal solutions of the *ILP* problem can be found inside the polyhedron. Therefore, the fundamental theorem of *LP* is not applicable.

1.2.3 Simplex Algorithm

In this part, we describe the different steps in the simplex algorithm [Nabli, 2009]. Indeed, from the following initial simplex table: (see Table.1.1).

X	RHS
A	b
C	f

Table 1.1: Simplex table

The transition to the canonical form relative to the realizable basis B.

1	0	$\cdots$	0	$\overline{N} = B^{-1}N$	$\overline{b} = B^{-1}b$
0	$\ddots$		$\vdots$		
$\vdots$		$\ddots$	0		
0	$\cdots$	0	1		
0	$\cdots$	0	0	$\overline{C}_N = C_N - C_B B^{-1}N$	$f(x) - C_B B^{-1}b$

Table 1.2: Optimal simplex table associated to the basis B

All the different steps of the simplex are present in the following algorithm 1.

1.2.4 Duality

Each linear problem is associated with another linear problem known as the dual of the initial problem. The latter being called by opposition primal. Table (1.4) gives the general rules for primal and dual problem:

Algorithm 1: Simplex algorithm

Inputs: A, b, C : Parameters: constraints and objective function.

Outputs: X_{opt}: Optimal solution of the Linear Program and Optimal Table (see 1.3).

Initial phase Start

The initial table is established by switching to the canonical form with regard to the base B (see 1.3). The constraints are written as : $\overline{A}X = \overline{b}$; $\overline{A} = B^{-1}A$, $\overline{b} = B^{-1}b$. $\overline{C}_N \leftarrow C_N - C_B B^{-1}N$.

End

IF $\overline{C}_N \geq 0$ **Then**

The solution is optimal and the algorithm ends.

Else

Consider the index j such that: $\left(\overline{C}_N\right)_j = \min_{k \in J}\left(\overline{C}_N\right)_k \mid \left(\overline{C}_N\right)_k < 0$.

IF $a_{ij} < 0, \forall i \in \{1, \cdots, m\}$ **Then**

There is no finite optimal solution, the value of the function is unbounded. $F \rightarrow \infty$;

Else

The variable x_j enters the basis, the index ℓ of the leaving variable is given by:

$\frac{\overline{b}_\ell}{\overline{a}_{\ell k}} = \min_{i \in I}\{\frac{\overline{b}_i}{\overline{a}_{ik}}\}, \overline{a}_{ik} > 0$. **Go to**, $\overline{I} \leftarrow I \backslash \{\ell\} \cup \{j\}$.

A new simplex table is created by applying the following formulae:

$a^\star_{ik} \leftarrow a_{ik} - \frac{a_{ij}a_{\ell k}}{a_{\ell k}},\ i \neq \ell,\ k \neq j. - a^\star_{jk} \leftarrow \frac{a_{\ell k}}{a_{\ell j}},\ k \neq \ell. - a^\star_{i\ell} \leftarrow -\frac{a_{ij}}{a_{\ell j}},\ i \neq j.$

$a^\star_{lk} \leftarrow \frac{1}{a_{\ell j}}. - c_j = c_j - c_\ell \times \frac{a_{\ell j}}{a_{\ell \ell}},\ \ j \in \{1, \cdots, n\}.$

End If

Stop: Back to. The algorithm ends when the solution is optimal or the pivot operation is impossible. See the optimal table (1.3) at iteration k.

End If

	value of $\mathbf{x_{B_k}}$	x_1	x_2	$\cdots$	x_{n_k}
$\mathbf{x_{B_k}}$	x_1^0	t_{11}	t_{12}	$\cdots$	t_{1n_k}
	x_2^0	t_{21}	t_{22}	$\cdots$	t_{2n_k}
	x_m^0	t_{m1}	t_{m2}	$\cdots$	t_{mn_k}
$(\overline{C_N})_j = (C_N - C_B B^{-1}N)_j$	$(\overline{C_N})_0$	$(\overline{C_N})_1$	$(\overline{C_N})_2$	$\cdots$	$(\overline{C_N})_{n_k}$

Table 1.3: Optimal simplex table obtained after k iterations

Problem Max f	Problem Min w
Constraints	Variables
$\leq$	≥ 0
$=$	unrestricted
Variables	Constraints
≥ 0	$\leq$

Notion of duality

Proposition 1.2.1 (Theorem of **Complementary Slackness**). *Let $\overline{x}$ and $\overline{y}$ be admissible solutions of the primal and dual, respectively. A necessary and sufficient condition for $\overline{x}$ and $\overline{y}$ to be optimum is that they satisfy the relations:*

$$\begin{cases} \left(b_i - \sum_{j=1}^{n} a_{ij}\overline{x}_j\right)\overline{y}_i = 0 & \forall\, i \in \{1, \cdots, m\}, \\ \left(c_j - \sum_{i=1}^{m} a_{ij}\overline{y}_i\right)\overline{x}_j = 0 & \forall\, j \in \{1, \cdots, n\}. \end{cases} \tag{1.9}$$

Proposition 1.2.2 (Weak duality). *If $\overline{x}$ and $\overline{y}$ are admissible solutions of the primal and dual respectively, they verify: $c\overline{x} \geq \overline{y}b$*

Corollary 1.2.1 (Sufficient optimality condition). *Let $\overline{x}$ and $\overline{y}$ admissible solutions of the primal and dual respectively. If $c\overline{x} = \overline{y}b$, then $\overline{x}$ and $\overline{y}$ are optimal solutions.*

Proposition 1.2.3 (Strong duality). *If the primal (dual) problem has a finite optimal solution, then so does the dual (primal) problem. $\widetilde{f} = \widetilde{w}$*

Theorem 1.2.1 (Duality theorem of Gale, Kuhn and Tucker). *The primal problem has an optimal solution $\overline{x}$ if and only if the dual problem has an optimal solution $\overline{y}$. In this case, we necessarily have $\widetilde{f} = \widetilde{w}$.*

1.2.5 Solving Methods for ILP Problems

In the development of ILP problem, there are three successive steps: cutting methods were the first to be proposed, mainly as a result of R.E. Gomory's work [Gomory, 2002], the early 60s, the Branch and Bound (B&B) procedure was then widely developed and began the path of combinatorial optimization. Twenty years later, cutting plans approach was given a second lease of life in the context of polyhedral theory, with the notion of valid inequalities, which may also need to be integrated into a Branch and Cut approach.

Branch & Bound Method

The "Branch & Bound" (B& B) method has been specially developed for integer linear programming problems (ILP) [Morrison et al., 2016]. The principle of this method consists in subdividing the set $\mathcal{S}$ (the set of admissible solutions) $\mathcal{S} = \{x \in \mathbb{R}^n, Ax \leq b, x \geq 0\}$, into a finite number of subsets $\mathcal{S}_i$, generally we take:

$$\mathcal{S} = \bigcup_i \mathcal{S}_i \;\; with \;\; \mathcal{S}_i \cap \mathcal{S}_j = \emptyset \;\; \forall i \neq j.$$

Definition 1.2.9 (Active node). *The set of nodes created but not fathomed, are called active nodes.*

The B&B method consists mainly of three different steps:

Branching: This step involves dividing the problem into a number of sub-problems, each with its own set of feasible solutions. We create a problem tree with the initial problem as the root. Let the node i be non-fathomed . There is therefore at least one non-integer variable in the optimal solution $x_i^\star$ of the relaxed problem $LP^{(i)}$, we choose the first non-integer $x_l = \alpha_l \notin \mathbb{Z}$.
The node i is separated into two sub-nodes, by imposing respectively the additional constraint:

$$\begin{cases} (i) & x_l & \leq & \lfloor \alpha_l \rfloor, \\ (ii) & x_l & \geq & \lfloor \alpha_l \rfloor + 1. \end{cases}$$

The $x_i^\star$ solution satisfies neither of these constraints. Constraints (i) and (ii) respectively define two disjoint areas of $\mathcal{S}_i$ containing all solutions of $\mathcal{S}_i$. It is easy (using the dual algorithm) to reoptimize the relaxed problems corresponding to these two subnodes.

$$\begin{cases} \mathcal{S}_{i+1} = & \mathcal{S}_i & \cap & \{x |\ x_l \leq \lfloor \alpha_l \rfloor\}, \\ \mathcal{S}_{i+2} = & \mathcal{S}_i & \cap & \{x |\ x_l \geq \lfloor \alpha_l \rfloor + 1\}. \end{cases}$$

Bounding: The purpose of bounding a node in the tree structure is to determine the optimum (an upper or lower bound for a minimization problem) of the set of feasible solutions associated with the node in question, or on the contrary, to prove mathematically that this set contains no solution of interest for solving the initial problem. The optimal solution of the sub-problem associated with a given node is called a partial solution.

Root Node Sterilization: The purpose of this step is to avoid examining all the nodes in the tree. If the upper bound of the optimal solution of the sub-problem under consideration is lower than the global upper bound, certainly, any feasible solution of this sub-problem is not better than the current global optimum, so there is no need to branch its solution set. We can also stop the search at a node when the associated sub-problem is infeasible.
The B&B method used to solve integer optimization problems can be summarized in the following algorithm 2:

Algorithm 2: Branch & Bound Method

Result: The integer optimal solution x_{opt}.

Initialization: R={L_o}, où L_o The relaxed main program (without integrity constraints). $x_{opt} = \emptyset$ et $f_{opt} = -\infty$

while *R is non-empty* **do**
  Take L a node of R. Solve L.
  if *L is unfeasible* **then**
    Fathom the node L.
  end
  if *L has an optimal integer solution* x_l **then**
    if $f_l < f_{opt}$ **then**
      $x_{opt} = x_l$ et $f_{opt} = f_l$.
    end
  end
  if *L has a continuous solution* **then**
    Choose an index l such that x_l is not an integer. Next, add two nodes to the tree R, adding to each node, respectively, the constraints $x_l \leq \lfloor x_l \rfloor$ et $x_l \geq \lfloor x_l \rfloor + 1$.
  end
end

Cutting Methods

Cutting methods aim to obtain the convex envelope of feasible integer solutions, i.e. the smallest polyhedron containing all feasible integer solutions of the ILP problem. If we can identify this convex envelope, the ILP relaxation resolution reduced to this set yields an integer optimal solution. The difficulty with these methods remains in generating efficient cuts. Cutting methods used alone show simple performance. On the other hand, they are effective when combined with tree search methods such as branch and cut [Lucena and Beasley, 1996] (a method combining the B& B algorithm and the polyhedral cut method).

1.3 Generalities of the Multi-Objective Linear Programming.

1.3.1 Problem formulation & definitions

MOP consists in simultaneously optimizing several conflicting functions on a discrete space [Gunantara, 2018]. Consequently, there is no single integer solution to be considered optimal. However, the challenge is to find a set of integer compromise solutions that contains the solutions that achieve a compromise between these conflicting objectives, known as efficient solutions. A MOP problem can be formulated

as follows:

$$(MOP)\begin{cases} \text{"}Opt\text{"} \quad F = f_1(x), f_2(x), \ldots, f_K(x)), \\ s.t., \\ \qquad\qquad x \in \mathcal{S} = \{Ax = b, x \geq 0\}, \end{cases} \tag{1.10}$$

where, $K \geq 2$ is the number of objectives and $\mathcal{S}$ the set of admissible solutions in the decision space. The image of $\mathcal{S}$ by the vector function $F = (f_1(x), f_2(x), \ldots, f_K(x))$ representing the image of the feasible set in criterion space is denoted by $\mathcal{Y}$.
This formulation is general, but multi-objective problems can be classified according to the nature of the objective functions and/or the admissible region. For example:

- If the objective functions and the admissible region are convex, then we say it's a convex MOP problem.
- If the objective functions and the functions forming the admissible region are linear, then it's called a MOLP problem.
- If the admissible region contains only integer solutions, then it's said to be a MOILP problem.

A multi-objective integer linear programming problem defined by:

$$(MOILP)\begin{cases} \min \quad f_i = C_i x, & i \in \{1, 2, \cdots, K\}, \\ s.t., \\ \qquad\qquad x \in \mathcal{S} = \{Ax = b, x \geq 0\}, \end{cases}$$

where, $\mathcal{S} \cap \mathbb{Z}^n$, $\mathbb{Z}$ is the set of relative integers number.

Definitions

Solving a MOILP problem often consists of finding the set of Pareto optimal solutions, i.e. those solutions where there is no solution that is at least as good on all criteria, and clearly better on at least one criterion.

Definition 1.3.1 (Dominance). *Let be two criteria vectors $f(x), f(y) \in \mathcal{Y}$. We say that $f(x)$ **dominates** $f(y)$ if and only if $f(x) \leq f(y)$ and $f(x) \neq f(y)$*
i.e.

$$f_i(x) \leq f_i(y),\ \forall\, i \in \{1, 2, \cdots, K\} \text{ and } \exists\, i \in \{1, 2, \cdots, K\} \text{ such that } f_i(x) < f_i(y).$$

If $f(x)$ **dominates** $f(y)$, then $f(x)$ is preferred to $f(y)$ since it is at least as good on all criteria and better on at least one.

Definition 1.3.2 (Efficiency). *A solution $\hat{x} \in \mathcal{X}$ is called a **efficient** (or Pareto optimal) solution, if and only if, $\nexists\, x \in \mathcal{S}$ such that $f_i(x)$ dominates $f_i(\hat{x})$, $\forall i \in \{1, 2, \cdots, K\}$.*

A point is efficient if and only if its image by f is an undominated criterion vector.

Definition 1.3.3. *A solution $\hat{x} \in \mathcal{S}$ is called an efficient solution if and only if there is no solution $x \in \mathcal{X}$ such that*

$$f_i(x) \leq f_i(\hat{x}),\ \forall\ i \in \{1, 2, \cdots, K\}\ et\ \exists\ i \in \{1, 2, \cdots, K\}\ with\ f_j(x) < f_j(\hat{x}).$$

From an efficient point, it is impossible to decrease the value of one criterion without increasing that of at least one other.

Definition 1.3.4. *A solution $\hat{x} \in \mathcal{S}$ is called a weakly efficient solution if and only if there is no solution $x \in \mathcal{S}$ such that:*

$$f_i(x) < f_i(\hat{x}),\ \forall i \in \{1, 2, \cdots, K\}.$$

Any efficient solution is necessarily weakly efficient, but the reverse is not true.

Definition 1.3.5 (Pareto optimal set)**.** *The Pareto optimal set of $\mathcal{S}$ (or the efficient set), is defined by the set $\mathcal{X}_{Eff}$:*

$$\mathcal{X}_{Eff} = \{x \in \mathcal{X} |\ \nexists\ x' \in \mathcal{S},\ f(x')\ dominates\ f(x)\}.$$

Definition 1.3.6. *A feasible solution $x \in \mathcal{S}$ is called supported if there is a linear combination of criterion objective functions for which x is optimal.*

Consider the following P_λ program:

$$(P_\lambda) \begin{cases} \min & \lambda_1 f_1(x) + \lambda_2 f_2(x) + \cdots + \lambda_K f_K(x), \\ \text{s.t.}, & \\ & x \in \mathcal{S}, \end{cases} \tag{1.11}$$

where, $\lambda_i \geq 0, \forall i \in \{1, 2, \ldots, K\}$.

Theorem 1.3.1. *Any optimal solution of P_λ is an efficient solution.*

Definition 1.3.7 (Pareto frontier of $\mathcal{Y}$)**.** *Let $\mathcal{Y}$ the image in criterion space of the feasible set $\mathcal{S}$. The Pareto frontier $\mathcal{Y}_{ND}$ of $\mathcal{Y}$ is defined by:*

$$\mathcal{Y}_{ND} = \{y \in \mathcal{Y} |\ \nexists y' \in \mathcal{Y},\ y' < y\}.$$

We provide here definitions of three particular points in the criterion space. The anti-ideal, ideal and nadir points are defined as follows:

Definition 1.3.8. *The ideal point and the anti-ideal point are defined respectively by:*

$$y_i^I = \min_{x \in \mathcal{S}} f_i(x), \quad i \in \{1, 2, \cdots, K\}.$$

$$y_i^A = \max_{x \in \mathcal{S}} f_i(x), \quad i \in \{1, 2, \cdots, K\}.$$

The nadir point is a refinement of the anti-ideal point, defined by:

$$y_i^N = \max_{x \in \mathcal{X}_{Eff}} f_i(x), \quad i \in \{1, 2, \cdots, K\}.$$

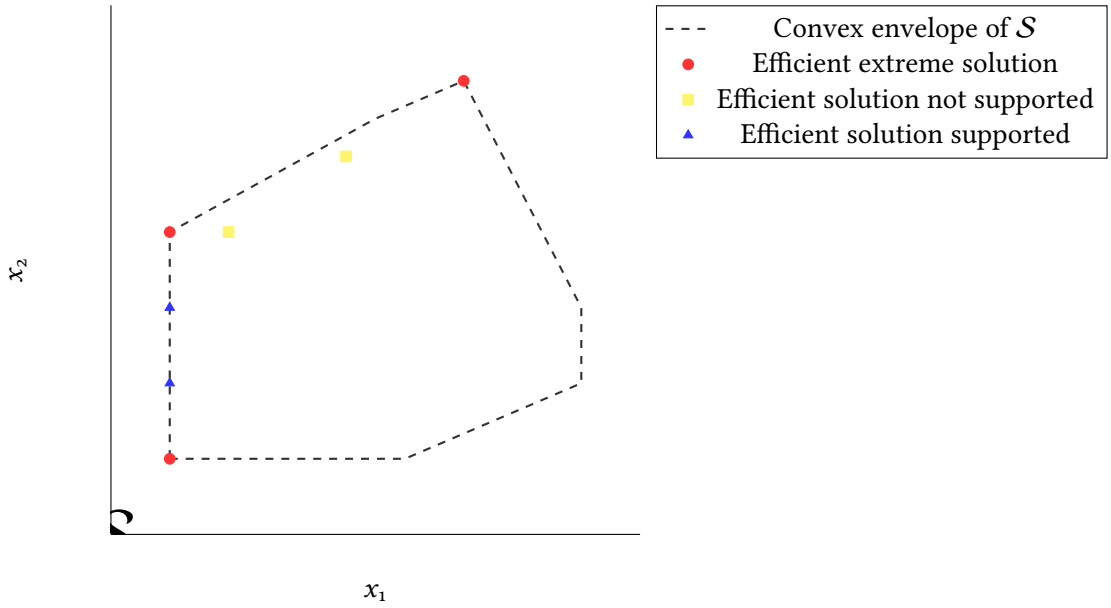


Figure 1.1: A representation of different types of solution in the decision space.

There are efficient solutions that are not supported in MOILP, unlike in the continuous case, and it is these solutions that make the problem difficult. A representation of $\mathcal{S}$ and $\mathcal{Y}$ with two decision variables and two objectives are shown in figures (1.1) and (1.2) respectively.

Efficiency test

In the realm of MOP, the existence of conflicting objectives leads to the generation of multiple efficient solutions. Additionally, efficiency can be defined in this context as follows:

Definition 1.3.9. *A point $\overline{x} \in \mathcal{S}$ is an efficient solution for a problem (1.10) if and only if there is no $x \in \mathcal{S}$ such that $\widetilde{f_i}(x) \leq \widetilde{f_i}(\overline{x})$ for all $i \in \{1, 2, \ldots, K\}$ and $\widetilde{f_i}(x) < \widetilde{f_i}(\overline{x})$ for at least one $i \in \{1, 2, \ldots, K\}$. Otherwise, x is not efficient and the corresponding vector $(\widetilde{f_1}(\overline{x}), \widetilde{f_2}(\overline{x}), \ldots, \widetilde{f_K}(\overline{x}))$ is said to be dominated.*

The following theorem (1.3.2) provides another characterization of an efficient solution, which is used as a test procedure in our study.

Theorem 1.3.2. *$x^l \in \mathcal{X}_E$ if and only if the optimal value of the objective function $\Psi(\psi, x)$ is null in the following integer linear programming problem:*

$$(EK(x^l)) \begin{cases} max \ \Psi = \sum_{i=1}^{K} \psi_i, \\ s.t., \begin{cases} C_i x + \psi_i = C_i x^l, \ i \in \{1, 2, \ldots, K\}, \\ x \in \mathcal{S}, \\ \psi_i \geq 0, \quad i \in \{1, 2, \ldots, K\}. \end{cases} \end{cases} \tag{1.12}$$

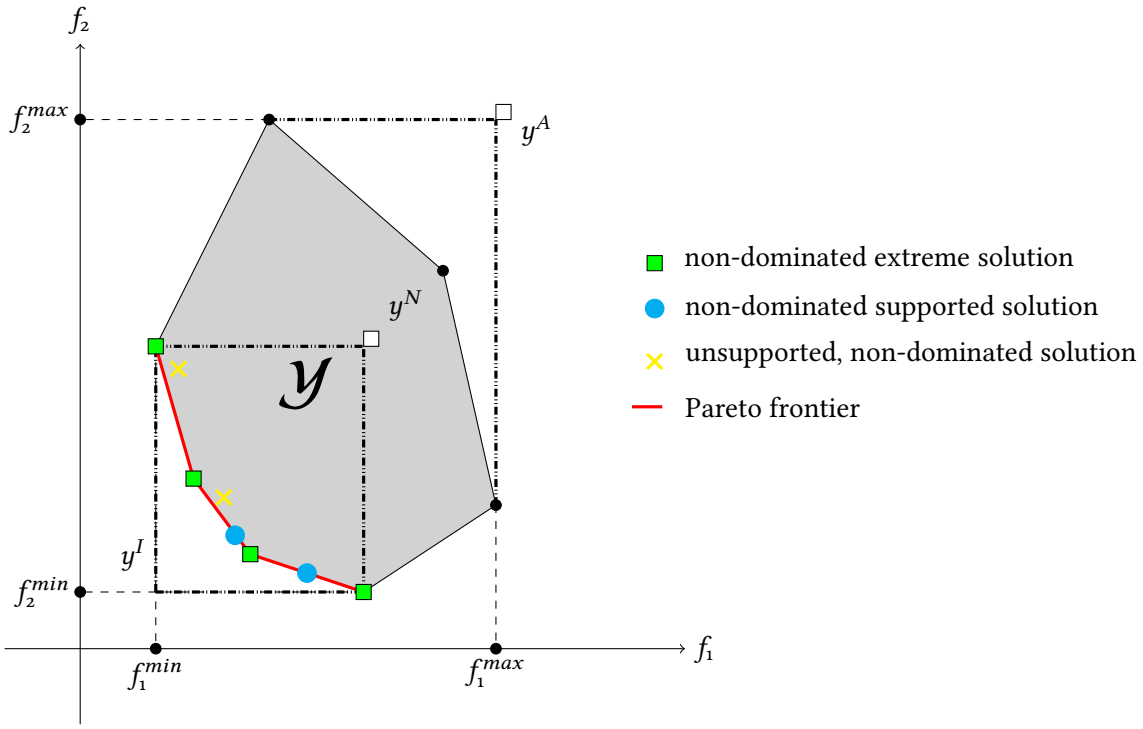


Figure 1.2: A representation of the different types of solution in the criteria space.

The efficiency of the solution x^l is verified through the resolution of the problem (1.12). In particular, x^l is efficient if and only if the optimal value of the objective function Ψ equals zero in this problem (1.12). In cases where the optimal value is not zero, the optimal solution of this problem (1.12) is considered an efficient solution to the problem (2.24). It is noteworthy that Theorem 1.3.2 is specifically applicable to the linear case and was originally proposed by [Ecker and Kouada, 1978]. For the general case, we mention the scalarization introduced by [Benson, 1984].

1.3.2 Solving Methods For MOLP Problems

weight sum method

The weighted sum (WS) method is a classic technique used to solve MOLP problems. This method converts multiple objectives into a single objective by assigning a weight to each objective function. The purpose is to find a solution that optimizes this weighted sum, thus balancing the trade-offs between the objectives.

The Weighted Sum Method can be formulated as follows:

$$(P_{WS}) \begin{cases} \min & F(x) = \sum_{i=1}^{K} w_i f_i(x), \\ s.t., & \\ & x \in \mathcal{S}, \end{cases} \tag{1.13}$$

where w_i are the non-negative weights assigned to each objective function, and $\sum_{i=1}^{K} w_i = 1$. By varying the weights, different trade-off solutions can be obtained, generating the Pareto front of the MOLP problem.

This method is widely used due to its simplicity and ease of implementation. However, it may not capture all Pareto optimal solutions, especially in non-convex problems. For further reading, see [Miettinen, 1999, Kim and de Weck, 2006, Marler and Arora, 2010].

The following algorithm 3 presents the main principles of the WS method.

Algorithm 3: Weighted Sum Method

Inputs: Objective functions $f_1(x), f_2(x), \ldots, f_K(x)$, Weights $w_1, w_2, \ldots, w_K$, $\mathcal{S}$.

Outputs: x_{opt}: Optimal solution Obtained.

Initial phase Start

> ***WeightedSumMethod*** $(f_1, f_2, \ldots, f_K, X, w_1, w_2, \ldots, w_K)$.
>
> *Define the weighted sum objective function:* $F(x) = \sum_{i=1}^{K} w_i f_i(x)$.
>
> *Solve the single objective linear programming problem 1.13*

End

Stop: The algorithm ends with the optimal solution x_{opt}

Epsilon Constraint Method

The Epsilon-Constraint (EC) Method is a widely used technique to solve MOLP problems. Instead of combining multiple objective functions into a single one, as in the WS Method.

The EC Method optimizes one objective function while treating the other objectives as constraints with specified bounds (epsilon constraints). The Epsilon-Constraint Method can be formulated as follows:

$$(P_{EC}) \begin{cases} \min & f_1(x), \\ s.t., & \\ & f_i(x) \leq \epsilon_i, i \in \{2, \ldots, K\}, \\ & x \in \mathcal{S}, \end{cases} \tag{1.14}$$

where, ϵ_i are the upper bounds for the objective functions $f_i(x)$, $i \in \{2, \ldots, K\}$. By varying the epsilon values ϵ_i, different trade-off solutions can be obtained, generating the Pareto front of the MOLP problem. The Epsilon-Constraint Method is advantageous because it directly handles the objectives as constraints, allowing more control over each objective function's satisfaction level. However, choosing appropriate epsilon values can be challenging and may require problem-specific knowledge. For further reading, see [Miettinen, 1999, Mavrotas, 2009].

The principle of the method is present in following algorithm 4

1.3.3 Solving Methods For MOILP Problems

Solving principles for multi-objective problems: The MOLIP class is relevant in many real-world applications, such as scheduling, resource allocation and supply chain management, where solutions

Algorithm 4: Epsilon-Constraint Method

Inputs: Objective functions $f_1(x), f_2(x), \ldots, f_K(x)$, Epsilon values $\epsilon_2, \epsilon_3, \ldots, \epsilon_K$, $\mathcal{S}$.

Outputs: x_{opt}: Optimal solution Obtained.

Initial phase Start

EpsilonConstraintMethod *($f_1, f_2, \ldots, f_K, X, \epsilon_2, \epsilon_3, \ldots, \epsilon_K$).*

Define the primary objective function to optimize: $f_1(x)$.

Define the epsilon constraints: $f_i(x) \leq \epsilon_i$ for $i \in \{2, \ldots, K\}$.

Solve the single objective linear programming problem 1.14

End

Stop: The algorithm ends with the optimal solution x_{opt}

must be discrete. Several methods have been developed to solve MOLIP problems. Such as, in the article [Halffmann et al., 2022], the authors provide a comprehensive overview of exact algorithms used to solve MOILP and multi-objective mixed integer linear programming (MOMILP) problems. The choice of algorithm depends on the specific characteristics of the problem, the desired level of accuracy and the available computing resources. In the following table 1.5, we present some algorithms that solve the MOLP and MOILP problems.

Algorithm	Reference
Branch-and-Bound Algorithm	[Przybylski and Gandibleux, 2017, Parragh and Tricoire, 2019]
NSGA-II (Genetic Algorithms II)	[Ibrahim et al., 2018, Verma et al., 2021]
Branch-and-Cut Algorithm	[Gadegaard et al., 2019, Cherfaoui and Moulaï, 2021]
Benders Decomposition	[Irmansyah et al., 2022, Raith et al., 2024]
Pareto Local Search	[Antunes et al., 2016, Pal and Charkhgard, 2019]

Table 1.5: Algorithms for MOLP and MOILP problems

In this part, we present some exact methods for generating the efficient set of a MOILP problem which is defined as follows:

$$(MOILP) \begin{cases} \min & f_i = C_i x \quad i \in \{1, 2, \ldots, K\}, \\ s.t., & \\ & Ax \leq b, \\ & x \in \mathbb{N}. \end{cases} \tag{1.15}$$

where, $k \geq 2$ is the number of objectives, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $C_i \in \mathbb{R}^{1 \times n}$.

Ouaïl et Al method

The method proposed by [Chergui et al., 2008] is based on the simplex, *B&B* procedure and an efficient cut to solve a *MOILP*. At each iteration l, they solve a single-objective LP problem by minimizing one

of the objectives, the choice of objective has no influence on the final solution.

$$(LP^1)\begin{cases} \min & f_1 = C_1 x, \\ s.t., & \\ & Ax \leq b. \end{cases} \tag{1.16}$$

Let x_l^* be the integer solution obtained after solving the problem (1.16), let $\mathcal{B}_l$ be the set of indices of the basic variables, and let $\mathcal{N}_l$ be the set of indices of the non-basic variables of x_l^*.
Let $\bar{C}_i^j$ is the j^{th} component of reduced gradient vectors of the objective function $\widetilde{f_i}$.
The method define a new set of all decreasing directions of the criteria as follows:

- $\mathcal{H}_l = \left\{ j \in \mathcal{N}_l / \exists i \in \{1, 2, \ldots, K\}, \text{ with } \bar{C}_i^j < 0 \right\} \bigcup \left\{ j \in \mathcal{N}_l / \bar{C}_i^j = 0; \forall i \in \{1, 2 \ldots, K\} \right\}.$

Once this set has been identified, the method proceed to formulate the efficient cut (1.17) designed to eliminate all inefficient solutions [Chergui et al., 2008]. This cut plays an essential role in the optimization process, enabling the algorithm to reduce the search space, improve the overall efficiency of the optimization process, and ultimately, find optimal solutions.

$$\sum_{j \in \mathcal{H}_l} x_j \geq 1. \tag{1.17}$$

Algorithm of the Ouaïl et Al method: The algorithm for generating all efficient integer solutions of the program (1.15) is presented in the following algorithm 5:

Algorithm 5: Ouaïl et Al method for MOILP problem

Result: Efficient solutions set $\mathcal{X}_E$

Step (0) **: Initialization (**$l = 0$**)**

$R = \{\mathcal{S}_0 = \{x, Ax \leq b\}\}$ ($\mathcal{X}_E = \emptyset$ ($\mathcal{X}_E$ being the efficient set of the problem (1.15)).

Step (1) **: General step**

while *There is an unfathomed node* l *in the tree* R. **do**

  Solve the linear program corresponding to the node l.

  if *Program has no solution* **then**

    Node l is fathomed.

  else

    Let $\widetilde{x_l}$ the obtained optimal solution.

    if $\widetilde{x_l}$ *is not integer* **then**

      Apply the branch and bound process, adding two nodes to R.

    else

      if $f(\widetilde{x_l})$ *is not dominated by* $f(x)$ *for any solution* $x \in \mathcal{X}_E$ **then**

        $\mathcal{X}_E = \mathcal{X}_E \cup \{\widetilde{x_l}\}$.

      else

        $\mathcal{X}_E = \mathcal{X}_E \backslash \{x\} \cup \{\widetilde{x_l}\}$

      end

      Determining the sets $\mathcal{N}_l$ et $\mathcal{H}_l$. **if** $\mathcal{H}_l = \emptyset$ **then**

        the corresponding node is fathomed.

      else

        add the cut $\sum_{j \in \mathcal{H}_l} x_j \geq 1$ to program (1.16).

      end

    end

  end

end

Sylva & Crema method

The method of [Sylva and Crema, 2004] is based on solving a sequence of integer linear programs that optimize a linear combination of criteria at each step. Each time, a new set of constraints is applied, ensuring the discovery of a new efficient solution. Finally, the method returns the set of all non-dominated multiobjective discrete linear programming solutions. The method is summarized in the algorithm 6 :

Algorithm 6: Sylva & Crema Method

Result: $\mathcal{X}_E$ Efficient set of *MOILP*

Choose $\lambda > 0$ and the steps $\epsilon_i > 0$ and M^i the upper bounds of f_i, $i \in \{1, 2, \dots K\}$, $l = 0$,
$\mathcal{S}_l = \{x, Ax <= b, x \in \mathbb{N}\}$ $\mathcal{X}_E = \emptyset$.
Solve: P^l_λ : min $\{\lambda Cx : x \in \mathcal{S}_l\}$.
while P^l_λ *is feasible* **do**
  Let x_l be the optimal solution of P^l_λ. Add x_l to the set $\mathcal{X}_E$.
  Poser $l = l + 1$ and add the following constraints to $\mathcal{S}_l$:

$$C_i x \leq (C_i x_l - \epsilon_i) y^i_l - M_i(1 - y^i_l), \forall i \in \{1, 2, \dots, K\}.$$

$$\sum_{i=1}^{p} y^i_l = 1.$$

end

1.4 Optimization over the efficient set of a Multi-objective Integer Linear Programming problem.

An exhaustive search for all efficient solutions is impractical, as the number of such solutions is sometimes enormous, and the decision-maker will be faced with the additional problem of choosing applicable solutions from a set of theoretically equivalent solutions. To overcome this problem, several researchers have considered including a utility function to be optimized over the efficient set. This approach gave rise to another type of problem, called optimization over the efficient set. It was first considered by [Philip, 1972]. Later, several researchers became interested in the subject, including, [Abbas and Chaabane, 2006], where they introduced an exact method for solving the optimization problem of a linear function over the efficient set of MOILP problem (LF/MOILP). On the other hand, [Jorge, 2009] proposed another approach to solve this problem using the same principle used by [Sylva and Crema, 2004] to enumerate the efficient set of the MOILP problem. Several recent works tackle the same problem and propose more efficient methods, e.g., [Boland et al., 2017, Lokman, 2021, Sierra Altamiranda and Charkhgard, 2019]. Regarding the bi-objective optimization over an efficient set of MOILP problem (BOLF/MOILP), we mention the only work of [Cherfaoui and Moulaï, 2021]. Below, a few works are presented that address the problem of optimization over the efficient set of MOILP problem.

1.4.1 Optimizing a liner Fuction Over an efficient set

Optimizing a linear function on an efficient set of MOILP problems can be written as follows:

$$(LF/MOILP)\begin{cases} \min\ \phi(x), \\ s.t., \\ \qquad x \in \mathcal{X}_E, \end{cases}$$

where, ϕ is a function and $\mathcal{X}_E$ is the set of efficient solutions to a MOILP problem.

Jorge Method

The [Jorge, 2009] method solves the problem of optimizing a linear function over the efficient set of a MOILP problem. The method is iterative: at each step, they optimize the utility function over the set of admissible solutions of the MOILP problem. the method then test the efficiency of the solution. If it is not efficient, the efficiency test returns an efficient solution, which will be the lower bound of the problem. Using the cuts used by [Sylva and Crema, 2004] to eliminate this solution from the domain along with all solutions dominated by it. Jorge's method is presented in the algorithm 7.

Variants of Jorge's method:

Jorge's method can be used to solve an optimization problem for any convex function on the efficient set of a MOILP problem. The steps remain almost the same except that instead of optimizing a linear function at each step, the method optimizes the convex function using an appropriate method [Mahdi and Chaabane, 2015] have considered the case where the utility function is fractional. Furthermore, a number of recent methods use techniques to improve Jorge's method. [Lokman, 2021] proposes a method to avoid adding constraints and decision variables to the R_l program at each iteration, which slows down the method. The method consists in establishing bounds that encompass all possible scenarios for the improvements to be made (instead of using binary variables). Then, if these bounds have not already been solved, they are solved. When there are fewer than four objectives, the method is efficient; however, when there are more than four objectives, the number of bounds explodes, and the method slows down considerably. On the other hand, [Boland et al., 2017], proposes dividing the search space into rectangles in order to find the optimal efficient solution. Tests have shown that their method is fast and can solve problems of considerable size.

1.4.2 B.Lokman Method

The author has developed an algorithm for optimizing a linear function over the set of non-dominated points of MOILP problem.

$$(P_\psi)\begin{cases} \max \psi(f) = \sum_{i=1}^{K} v_i f_i(x), \\ s.t., \\ \qquad x \in \mathcal{S}, \\ \qquad f(x) \in \mathcal{Y}_{ND}, \end{cases}$$

Algorithm 7: Jorge Method

Result: The optimal solution x_{opt}

Initialization: Poser $\phi_{inf} = -\infty$, $x_{opt} = \emptyset$ $\mathcal{S}_0 = \{Ax <= b, x \in \mathbb{N}\}$, $\phi_{sup} = +\infty$, $l = 0$, $(R_0) : \{max_{x \in \mathcal{S}} \ \phi(x)\}$.

while R_l *has an optimal solution* x_l **do**

  if *(x^l) is efficient* **then**

    if $\phi(x^l) > \phi_{inf}$ **then**

      $\phi_{inf} = \phi(x_l)$

      $x_{opt} = x_l$

    end

    return x_{opt}

  else

    Solve $(T_l) : \max\{\phi(x) \backslash Cx = Cx_l, x \in \mathcal{S}\}$.

    Let $\hat{x}^l$ optimal solution of T_l.

    if $\phi(\hat{x}_l) > \phi_{inf}$ **then**

      Poser $\phi_{inf} = \phi(\hat{x}_l)$ and $x_{opt} = \hat{x}^l$.

    end

    if $\phi_{inf} = \phi_{sup}$, **then**

      Stop. x_{opt} is optimal solution.

    end

    Build $\mathcal{S}_{l+1} = \mathcal{S}_l \cup \{Cx > C\hat{x}_l\}$.

    Poser $l = l + 1$.

  end

end

where, $\psi(f)$ is not a strictly positive linear combination of the objectives, there is a $v_i \leq 0$ for a $i \in \{1, 2, \ldots, K\}$. The algorithm iteratively generates non-dominated points and converges on an optimal solution by progressively reducing the admissible set. It maximizes a criterion $f_m(x),\ m \in \{1, 2, \ldots, K\}$ at each iteration, and generates new points by imposing bounds on the value of the linear utility function. The problem to be solved at each iteration is given by:

$$(P_m^p) \begin{cases} \max f_m(x), \\ s.t., \\ \qquad f_i(x) \geq b_i^k, \quad i \in \{1, 2, \ldots, K\}, \quad i \neq m, \\ \qquad \sum_{i=1}^{K} v_i f_i(x) \geq v^l + g^* |v^l|, \\ \qquad \sum_{i=1}^{K} v_i f_i(x) \leq v^u, \\ \qquad x \in \mathcal{S}. \end{cases}$$

The algorithm's decomposition and search procedure is accompanied by problem-specific mechanisms for efficiently exploring the space of objective functions. The algorithm is designed to produce solutions meeting a predefined accuracy level g^*.

Method's algorithm:

Algorithm 8: B.Lokman's method

Resultat : The optimal solution of the utility function on all non-dominated points $\mathcal{Y}_{ND}$ of MOILP problem

• **Step 1: Initialization**

$m = \arg\max_{i\in\{1,2,\dots,K\}} v_i$.

$w_i = v_i - \min_{l\in\{1,2,\dots,K\}} v_l + 1$, $i \in \{1, 2, \dots, K\}$.

$f'^{o} = \max_{x\in\mathcal{X}} \psi(f)$ et $\psi^u = \psi(f'^{o})$.

Solve E_o^w and note the optimum point as f^o.

Define $f^{inc} = f^o$ and $\psi^l = \psi(f^o)$,

put $f^{inc} = f^o$ and $\psi^l = \psi(f^{inc})$.

$f'^{1} = \max_{x\in\mathcal{X}} f_m(x)$ and $n = 1$

• **Step 2: Non-dominance check**

Solve $E_n^w(x)$.

Note the optimum point as f^n.

• **Step 3: Updating of bounds**

If $\psi(f^n) > \psi(f^{inc})$, **Then**

Define $f^{inc} = f^n$ and $\psi^l = \psi(f^{inc})$.

IF $g = \frac{|\psi^w - \psi(f^{inc})|}{|\psi^w|} \leq g^*$.

Go to step 5.

Else.

Go to step 4.

• **Step 4: Generating new points**

Define $n = n + 1$ and update $\mathbf{K}_m^n$.

Solve P_m^k corresponding to each$k \in \mathbf{K}_m^n$.

If $P_n^k$is impossible for all **Then**

$k \in \mathbf{K}_m^n$, go to step 5.

Else

Find $k^* = \arg\max_{k\in\mathbf{K}_m^n, f_{P_m^k}\neq\emptyset} \psi(f_m^k)$, assign $f'^{n} = f^{k^*}$, and go to step 2.

• **Step 5: Stop.**

f^{inc} maximizes (approximates) the ψ function on the set of undominated points (with the level of precision, g^*).

The algorithm verifies whether a point $f_i^{'n}$ is dominated or not by solving the non-dominance test 1.18:

$$(E_n^w)\begin{cases} \max \sum_{j=1}^{K} w_i f_i(x), \\ s.t., \\ \qquad f_i(x) \geq f_i^{'n}, \quad i \in \{1, 2, \ldots, K\}, \quad i \neq m, \\ \qquad x \in \mathcal{S}. \end{cases} \tag{1.18}$$

1.4.3 Biobjective Optimzation over an efficient Set

Bi-objective optimizing over the efficient set of MOILP problems can be written as follows:
Let d^1, d^2 be two row-vectors in $\mathbb{R}^n$ representing preference functions of decision makers.

$$(BOLF/MOILP)\begin{cases} \min \ \phi^1 = d^1 x, \\ \min \phi^2 = d^2 x, \\ s.t., \\ \qquad x \in \mathcal{X}_E, \end{cases}$$

where, $\mathcal{X}_E$ is the set of efficient solutions to a MOILP problem.

Charfaoui and Moulaï Metohd

[Cherfaoui and Moulaï, 2021] propose an exact method to optimize two preference functions over the efficient set of a multiobjective integer linear program (MOILP). The problems arise whenever two associated decision-makers have to optimize their respective preference functions over many efficient solutions.
The proposed method is a branch-and-cut algorithm based on linear programming, for finding efficient solutions in terms of both preference functions and the MOILP problem, without explicitly enumerating all efficient solutions of the MOILP problem. The branch and bound process, strengthened by efficient cuts and tests, allows us to prune a large number of nodes in the tree to avoid many solutions.
Algorithm: The following algorithm 9 present the different steps of the proposed method:

Algorithm 9: Bi-objective optimization over an efficient set of MOILP problem.

Result: $\mathcal{X}_{Eff}$: the solution to $(BOLF/MOILP)$

• **Step 1: Initialization**: $\mathcal{X}_{Eff} = \emptyset$, $l = 0$, $\mathcal{S}_0 = \mathcal{S}$.

while *there is an unfathomed node* l **do**

• solve $(LP)_l$ using simplex or dual simplex method.

if $(LP)_l$ *has an optimal solution* $x^{*(l)}$ **then**

if $x^{*(l)}$ *is integer* **then**

Solve $(EK^1(x^l))$;

if *the optimal value of the objective function of* $(EK^1(x^l))$ *is* 0 **then**

Solve $(EK^2(x^l))$;

if *the optimal value of the objective function of* $(EK^2(x^l))$ *is* 0 **then**

$\mathcal{X}_{Eff} = \mathcal{X}_{Eff} \cup \{x^{*(l)}\}$

end

end

Construct the sets $\mathcal{H}_l$ and $\mathcal{H'}_l$;

if $\mathcal{H}_l = \emptyset$ *or* $\mathcal{H'}_l = \emptyset$ **then**

Fathom the node l

else

Add the two cuts relative to $\mathcal{H}_l$ and $\mathcal{H'}_l$ to the successors of l;

end

else

Choose an index k such that $x_k^{*(l)}$ is fractional. Then, split the program $(LP)_l$ into two subprograms by adding respectively the constraints $x_k = \lfloor x_k^{*(l)} \rfloor$ and $x_k = \lfloor x_k^{*(l)} \rfloor + 1$ to obtain $(LP)_{l_1}$ and $(LP)_{l_2}$, where $l_1 > l$, $l_2 > l$, and $l_1 \neq l_2$.

end

else

Fathom the node l.

end

end

Chapter

2 Multi-objective Stochastic Linear Programming

"The only way to do great work is to love what you do."
"La seule façon de faire un excellent travail est d'aimer ce que vous faites."
Steve Jobs

2.1 Introduction

The notion of randomness in optimization problems first appeared in the 50s in the work of [Bellman, 1958], [Beale et al., 1986], and ([Sengupta et al., 2012]) and since then, stochastic programming has seen very rapid development. Uncertainty in optimization problems affects many areas, such as market prices, delivery times, machine availability and many others. In stochastic linear programming, certain aspects of the random phenomenon may be known from historical data, or from probability laws, or from the moments of the random variable (e.g. the moment of order 1, the mathematical expectation or the variance), or from scenarios that can be created from historical data (e.g. the mortality rate among newborns, the temperatures of the last decade), or from experts who can predict the behavior of the random phenomenon.
In this chapter, we present various approaches used in stochastic programming and explore different models within stochastic programming. We will also provide an overview of MOSILP and discuss stochastic optimization over the efficient set of a MOSILP problem.

2.2 Stochastic Linear Programming Overview

2.2.1 Definitions & Approaches

The general model of stochastic programming is defined as follows:

$$(SLP)\begin{cases} \text{“ min ”} & f = C(\xi)x, \\ s.t., & \\ & T(\xi)x = h(\xi), \\ & x \in \mathcal{S}, \end{cases} \tag{2.1}$$

where, $x \in \mathbb{R}^n$, (T, C, h) of respective dimensions $(m_0 \times n)$, $(1 \times n)$ and $(m_0 \times 1)$.
The coefficients of $T(\xi)$, $C(\xi)$ as well as $h(\xi)$ are random variables defined on a probability space (Ω, Ξ, P).
$\mathcal{S} = \{x \mid Ax \leq b,\ x \geq 0\}$ is a deterministic convex polyhedron.

There are two approaches to stochastic linear programming [Kolbin, 1977, Birge and Louveaux, 2011, Haneveld and Van der Vlerk, 2020].

1. ***”Here and Now” Approach.***
   As its name suggests, it involves dealing with the real problem of making a decision in the face of an uncertain future. A solution must be found immediately, without prior knowledge of the outcome of the random $\xi \in \Omega$.

2. ***”Wait and See” Approach.***
   This approach designates the situation in which the decision-maker can wait for the random realization before taking the decision, which therefore corresponds to the optimal solution $\widetilde{x}(\overline{\xi})$ of the deterministic linear problem defined by the $\overline{\xi}$ realization obtained. This approach is interesting from a theoretical point of view because of the distribution problem it poses. Considering $\widetilde{x}(\overline{\xi})$ as the optimal solution of the problem depending on a random parameter $\overline{\xi} \in \Omega$, this problem consists in determining the distribution of the optimal solution $\widetilde{x}(\xi)$ for $\xi \in \Omega$.

However, the problem (2.1) is not well defined as the meaning of “ min ”, and the constraints are not if we want to make a decision on x before knowing . Consequently, a revision of the modeling process is necessary, leading to a transformation into an equivalent deterministic problem for (2.1) that can be established by different approaches.

2.2.2 Optimization criterion for the equivalent problem

Various ways of defining the objective function of the equivalent problem can be used:

Bayes or mathematical expectation criterion

This criterion is used to minimize the mathematical expectation $\overline{f} = E(C^T(\xi))x = C^T(\xi)x$ of objective $f(x, \xi)$.

$$\min_{x \in \mathcal{S}} \overline{f}(x) = C^T(\xi)x. \tag{2.2}$$

The disadvantage of this criterion is that, by unifying the importance of gains, whatever their value, it does not take into account the notion of risk.

Variance criterion

Another classic interpretation of the program (2.1)is to minimize the variance of the objective $f(x, \xi)$

$$\min_{x \in \mathcal{S}} \sigma^2 = x^t V x. \tag{2.3}$$

Expectation-Variance criterion

This criterion consists in minimizing the variance of $f(x, \xi)$ while achieving a minimum level of return f_o set beforehand by the decision-maker:

$$\begin{cases} \min\limits_{x \in \mathcal{S}} \quad \sigma^2, \\ s.t., \\ \qquad\qquad \overline{C}^t x \geq f_o. \end{cases} \tag{2.4}$$

The problem with this criterion is the choice of the right f_o.

Kataoka criterion

Kataoka-type problems are of the form:

$$(p) \begin{cases} \min \quad u, \\ s.t., \\ \qquad\qquad P\left(C^T(\xi)x \leq u\right) = \alpha. \\ \qquad x \in \mathcal{S}. \end{cases} \tag{2.5}$$

For this model to be used effectively, u must minimize the objective $F(x, \xi)$, in this case $C(\xi)$ should follow a normal distribution with mean $\overline{c}$ and variance $\sigma^2(x) = x^t V x$, such that:

$$P\left(C^T(\xi)x \leq u\right) = \alpha \;\Leftrightarrow\; \phi\left(\frac{u - \overline{C}^t x}{\sqrt{x^t V x}}\right) = \alpha \;\Leftrightarrow u = \overline{C}^t x + \phi^{-1}\sqrt{x^t V x}.$$

Minimize u reduces to solving the following problem:

$$\min_{x\in\mathcal{S}} g(x) = \overline{C}^t x + \phi^{-1}(\alpha)\sqrt{x^t V x}. \tag{2.6}$$

The matrix V is positive definite, hence $\sqrt{x^t V x}$ is convex, $\overline{C}^t x$ is affine. The function $g(x)$ is convex if and only if $\phi^{-1}(\alpha) \geq 0$, in other words, $\alpha \geq \frac{1}{2}$, see [Kataoka, 1963].
The term $\phi^{-1}(\alpha)\sqrt{x^t V x}$ can be interpreted as a penalty for accepting a risk, which is greater the higher the variance $\sigma^2(x)$ of $F(x,\xi)$.
The problem (2.6) is solved by an iterative method by Kataoka:
The term $\phi^{-1}(\alpha)\sqrt{x^t V x}$ is first neglected, which amounts to minimizing $\overline{C}^t x$ over the decision set $\mathcal{S}$.
Given x_0 as the optimal solution and $R_0 = x_0^t V x_0$, we solve the following quadratic problem:

$$\max_{x\in\mathcal{S}} \left(\overline{C}^t + \phi^{-1}(\alpha)(R_0^{\frac{1}{2}} x^t V x)\right). \tag{2.7}$$

Solving the program (2.7) gives a solution x_0^1 and $R_0^1 = x_0^{1^t} V x_0^1$, we solve the quadratic program by replacing R_0 by $R_0^{(1)}$.
Handling random constraints requires special attention.

2.2.3 Different types of SLP model

There are two ways of defining the constraints of the equivalent deterministic problem in stochastic programming:

- Model with probability threshold on constraints "Chance Constrainted Programming"
- Recourse model "Stochastic Programming with Recourse"

Chance Constraints (CC) Model

These models first appeared in the work of [Charnes and Cooper, 1962] under the name of probabilistic constraint programming. The modeling idea for CC model is to impose a probability for violation of random constraints.

- Assume a probability threshold α_i for each random constraint, $i \in \{1,\ldots,m\}$, with $0 \leq \alpha_i \leq 1$. We assume that the cost function does not depend on the random variable ξ

$$\begin{cases} \min \quad C^T x, \\ s.t., \\ \qquad P(T_i(\xi)x \geq h_i(\xi)) \geq \alpha_i, \qquad i \in \{1,\ldots,m\}, \\ \qquad Ax \leq b, \end{cases} \tag{2.8}$$

  where, α_i being a user-set probability that reflects an acceptable tolerance for random constraint violation.

- Assume a global probability threshold α, for all constraints. The problem is as follows:

$$\begin{cases} \min \quad C^T x, \\ s.t., \\ \qquad P(T(\xi)x \geq h(\xi)) \geq \alpha, \\ \qquad Ax \leq b. \end{cases} \tag{2.9}$$

In this situation:

$$\mathcal{S}(\alpha_i) = \{x \in \mathbb{R}^n | P[T_i(\xi) \geq h_i(\xi)]\}.$$
$$\mathcal{S}(\alpha) = \{x \in \mathbb{R}^n | P[T(\xi) \geq h(\xi)]\}.$$

The problem with this approach is that the solution set of problems (2.8)and (2.9) in general is not convex, in addition, this problem of convexity of the sets $\mathcal{S}(p_i)$ and $\mathcal{S}(p)$ depends on the random nature of T, h.and also probability thresholds α, α_i. To solve this problem, we cite some convexity conditions for the sets $\mathcal{S}(\alpha)$ or $\mathcal{S}(\alpha_i)$, see [Kall, 1976, Sengupta et al., 2012].

T and h are random and normally distributed

- **T and h are not independent**
  Assume that $(T_i,\ h_i)^{'}$ is a mean normally distributed random vector $\mu \in \mathbb{R}^{n+1}$ and variance-covariance matrix $\mathcal{S}$

$$\mathcal{S}(\alpha_i) = \{x | P(\zeta(x) \geq 0) \geq \alpha_i\},$$
$$\mathcal{S}(\alpha_i) = \{x | P(\frac{\zeta(x) - m_\zeta(x)}{\sigma_\zeta(x)} \geq \frac{-m_\zeta(x)}{\sigma_\zeta(x)}) \geq \alpha_i\},$$
$$= \{x | 1 - \phi(\frac{-m_\zeta(x)}{\sigma_\zeta(x)}) \geq \alpha_i\},$$
$$= \{x | \phi(\frac{-m_\zeta(x)}{\sigma_\zeta(x)}) \leq 1 - \alpha_i\},$$
$$= \{x | \frac{-m_\zeta(x)}{\sigma_\zeta(x)} \leq \phi^{-1}(1 - \alpha_i)\},$$
$$= \{x |\ m_\zeta(x)\ +\ \sigma_\zeta(x)\phi^{-1}(1 - \alpha_i) \geq 0\}.$$

  Since $m_\zeta(x)$ is affine in x and $\sigma_\zeta(x)$ is convex in x, the constraint $\{m_\zeta(x) + \sigma_\zeta(x)\phi^{-1}(1 - \alpha_i) \geq 0\}$ is convex if and only if $\{\phi^{-1}(1 - \alpha_i) \geq 0\}$ and this is the case if $\alpha_i \leq 0.5$.

- **T and h are independent**
  Let the distributions of variables T and h.

$$T_{ij} \rightsquigarrow N(\mu_{ij}, V_{ij}^2), h_i \rightsquigarrow N(m_i, \sigma_i^2).$$

  Let the variable $y_i = Tix - h_i$ has for distribution:

$$y_i \rightsquigarrow N(\sum_{j=1}^{n} \mu_{ij} - m_i,\ \sum_{j=1}^{n} V_{i,j}^2 x_j^2 + \sigma_i^2).$$

Hence

$$\mathcal{S}(\alpha_i) = \{x \in \mathbb{R}^n | \sum_{j=1}^{n} \mu_{ij} x_j - m_i + \phi^{-1}(\alpha_i) \sqrt{\sum_{j=1}^{n} V_{i,j}^2 x_j^2 + \sigma_i^2} \leq 0\}.$$

T is deterministic and h is random

In this case, the problem is simple. Let F_i be the distribution function of h_i, then:

$$\mathcal{S}(\alpha_i) = \{x | P(T_i(\xi)x \geq h_i(\xi) \geq \alpha_i\},$$

$$\mathcal{S}(\alpha_i) = \{x | F_i(T_i x) \geq \alpha_i\},$$

$$\mathcal{S}(\alpha_i) = \{x | T_i x \leq F_i^{-1}(\alpha_i)\},$$

where $\mathcal{S}(\alpha_i)$ is a set of linear constraints at x, hence it is convex.

Recourse model

Stochastic programming (SP) is essentially interested in problems where a decision has to be made about the field in the presence of uncertainty, without waiting for the realization of certain random variables; this type of problem is often referred to as "here and now" [Walkup and Wets, 1967, Hansotia, 1980]. The problem can be spread over several stages, depending on the purpose of the study, but in this chapter let's consider the case of two stages only. The variables in the first stage are considered to be variables that can no longer change in the second stage. The decision at the first stage must be taken within the constraints specific to that stage. The second stage comprises actions that can be taken to satisfy constraints involving variables from the first stage. All these steps are summarized in the following program:

$$\begin{cases} \min & C^T x + Q(x), \\ s.t., & \\ & Ax \leq b, \\ & x \geq 0, \\ with & Q(x) = E[Q(x,\xi)] = \sum_j p^j Q(x, \xi^j), \end{cases} \tag{2.10}$$

where,

$$Q(x,\xi) = \min\{q(\xi)^T y | W(\xi)y = h(\xi) - T(\xi)x, y \geq 0\}. \tag{2.11}$$

And, p^j is the probability of $\widetilde{\xi} = (\xi)^j$, the j^{eme} realization of $\widetilde{\xi}$, $W(\xi)$ is called a recourse matrix of dimension (m_0, n_0).
$h(\xi) = h_0 + H\xi = h_0 + \sum_i h_i \xi_i$, $T(\xi) = T_0 + \sum_i t_i \xi_i$ and $q(\xi) = q_0 + \sum_i q_i \xi_i$ and $Q(x,\xi)$ represents the optimal recourse value and $E[Q(x,\xi)]$ represents the mathematical expectation of the cost of recourse. Problem (2.10) is the first-level problem, to be solved without the random constraints, and problem (2.11) is the second-level problem for a given decision x and realization x^i.

Different types of recourse

In the literature, there are several types of recourse; in our thesis, we will mention the most frequently used:

Definition 2.2.1. ***Fixed recourse*** *A recourse is considered to be fixed or deterministic if the values q and W of a program with recourse (2.10) are not dependent on ξ, and their values are known a priori.*

Definition 2.2.2. ***Complete recourse*** *The fixed recourse is considered as complete if:*
$\forall x \in \mathbb{R}^n$ *there is* $y \in \mathcal{Y}$ *such as* $Q(x, \xi) < +\infty; \forall \xi \in \Xi$.

Definition 2.2.3. ***Relatively complete recourse*** *A fixed remedy is relatively complete if:*
$\forall x \in \mathcal{S} \subseteq \mathbb{R}^n$ *there is* $y \in \mathcal{Y}$ *such as* $Q(x, \xi) < +\infty \; \forall \xi \in \Xi$.

The difference between the two types of recourse concerns the set of x for which there is a solution at the second level. If this set is the domain of definition of x in the first step, the recourse is relatively complete, whereas if it is $\mathbb{R}^n$, then the recourse is complete.

Definition 2.2.4. ***Simple recourse*** *This is a special case of complete fixed recourse, corresponding to the case where the recourse matrix $W = (I, -I)$ where I represents the identity matrix of order m_0.*

2.2.4 L-Shaped Decomposition Method

In this section, we describe the L-shaped decomposition method of Van Slyke and Wets [Van Slyke and Wets, 1969]. The principle of this method is to break down the problem into two stages. The first-stage variables are those whose value must be fixed before the random variables are realized, and cannot be modified in any way following the realization of the random variable. On the other hand, the second-stage variables (the recourse variables) are allowed to depend on the random variables and the first-stgae variables. The principle of the L-Shaped method is that we take into account all available information in order to make decisions, but at the same time, it is unrealistic to take into account unknown events, which exist, at the time of the decision, only in a probabilistic space. It's crucial to discern between two methodologies, both labeled as the "L-shaped method." The more widely recognized version, employed in this paper, pertains to a SP approach. In contrast, [Boland et al., 2016] introduced their "L-shaped method" tailored for solving deterministic multi-objective problems through a search for non-dominated solutions.

Feasibility test

Another definition of feasibility is introduced. If we have a solution of the first level $x = x^0$ of the problem (2.10), how do we decide if this solution is feasible for the problem (2.11) and for all possible values of ξ, in the case where we can't know if the recourse is relatively complete.

Lemma 2.2.1 (Farkas's lemma [Kall and Kall, 1976])**.**

$$\{y|Wy = h, y \geq 0\} \neq \emptyset$$

, if and only if

$$W^T u \geq 0 \Rightarrow h^T u \geq 0.$$

By changing the sign of u, the second part of the equivalent can be rewritten as follows:

$$W^T u \leq 0 \Rightarrow h^T u \leq 0,$$

or the equivalent:

$$h^T u \geq 0 \quad when \quad t \in \{u|W^T u \geq 0\}.$$

However, this can be reformulated as:

$$\{u|W^T u \leq 0\} = \{u|u^T Wy \leq 0 \quad \forall y \leq 0\}.$$

$$\{u|W^T u \leq 0\} = \{u|u^T h \leq 0 \quad \forall h \in pos\ W\}. \tag{2.12}$$

The expression (2.12) defines the polar cone of $posW$:

$$pol\ pos\ W = \{u|u^T h \leq 0 \quad \forall h \in posW\}.$$

In the case where we do not know all the values of $pol\ posW$, and we are not aware of the relatively complete recourse, for a given x^0 and for all $\xi \in$ we need to check feasibility.
We would like to find σ such that:

$$\sigma^T t \leq 0 \quad \forall t \in posW.$$

This is equivalent to requiring that $\sigma^T W \leq 0$. In other words,σ must be in the $polposW$ cone, we should at the same time require that $\sigma^T [h(\xi) - T(\xi) x^0] \geq 0$, because if we can add the constraint $\sigma^T [h(\xi) - T(\xi) x] \leq 0$ to the problem (2.1), we must exclude $[h(\xi) - T(\xi) x^0]$ without excluding feasible solutions. So we solve the following problem:

$$\max_{\sigma} \{\sigma^T (h(\xi) - T(\xi)) x^0 | \sigma^T W \leq 0, \|\sigma\| \leq 1\}. \tag{2.13}$$

The last constraint is introduced to bound σ, otherwise the maximum value will be equal to $+\infty$, and we are not interested in this as we are looking for a direction well defined by σ.
If for a given ξ_i we have found that $[h_i - T_i x^0]\ \sigma > 0$ then for this ξ_i the first-stage solution $x = x^0$ does not generate the feasible second-stage problem, so we have to exclude this x^0 solution, thus creating the feasibility cut

$$\sigma^T [h_i - T_i x] \leq 0. \tag{2.14}$$

Optimality Test

Assuming we have a complete relative recourse or feasibility cuts, the second-stage problem:

$$\min\{q(\xi)^T y | Wy = h(\xi) - T(\xi)x, y \geq 0\}. \tag{2.15}$$

is feasible and its dual problem is given by:

$$\max_{\pi}\{\pi^T (h(\xi) - T(\xi)x) \mid \pi^T W \leq q(\xi)^T\}. \tag{2.16}$$

The problem (2.10) can be rewritten by introducing a new variable θ

$$\begin{cases} \min \quad C^T x + \theta, \\ s.t., \\ \qquad\qquad Ax \leq b, \\ \qquad\qquad \sigma^T T x \geq \sigma^T h, \\ \qquad\qquad \theta \geq Q(x), \\ \qquad\qquad x \geq 0. \end{cases} \tag{2.17}$$

x^0 a feasible solution of the first stage and θ^0 initially set to $-\infty$, the value of $Q(x^0)$ is calculated from the dual problem (2.16)

$$Q(x^0) = \sum_i^N p^i Q(x^0, \xi^i) = \sum_i^N p_i \pi_i^T (h_i - T_i x^0). \tag{2.18}$$

If $Q(x^0) < \theta^0$, then x^0 is an optimal solution, otherwise we exclude the x^0 solution and create the optimality cut

$$\theta \geq \sum_i^N p_i \pi^T (h_i - T_i x). \tag{2.19}$$

The following algorithm 10 resumes in detail the main steps of the L-shaped method:

2.3 General Aspects of Multi-Objective stochastic Linear Programming

2.3.1 Problem formulation & Definitions

Algorithm 10: L-shaped Method

Data: Initialization $\theta = -\infty$, $l = 0$.
Result: X_{sol}: the optimal solutions of the problem (2.17).
while *As long as a unfathomed node l exists in the tree* **do**
  solve the problem (2.17) using simplex or dual simplex method;
  if *the problem (2.17) has a feasible solution* x^l **then**
    if x^l *is integer* **then**
      Feasibility test
      Solve the program (2.13);
      if $\sigma_r^T[h(\xi^r) - T(\xi^r)x^l] > 0$ **then**
        Add the feasibility cut (2.14) to the successors of l;
      else
        Optimality test
        Solve the program (2.16), calculate Q(x);
        if $Q(x^l) > \theta^l$ **then**
          Add the optimality cut (2.19) to the successors of l;
        else
          $X_{sol} = x^l$
        end
      end
    end
  end
end
The corresponding node l is fathomed;

MOSLP problem

A MOSLP problem is defined, in general, as follows [Stancu-Minasian, 1990]:

$$\begin{cases} \text{“min”} \quad C_i(\xi)x \quad i \in \{1, 2, \ldots, K\} \\ s.t., \\ \qquad\qquad Ax \leq b, \\ \qquad\qquad T(\xi)x = h((\xi), \\ \qquad\qquad x \geq 0, \end{cases} \tag{2.20}$$

where, the coefficients (T, c, h) are random variables with known distributions, defined on a probability space (Ω, Ξ, P), with (c, T, h) of respective dimension (p, n), (m_0, n), $(m_0, 1)$.

MOSILP problem:

A MOSILP problem can be formulated as follows [Amrouche and Moulaï, 2012, Badaoui et al., 2024]:

$$\begin{cases} \text{“min”} & C_i(\xi)x \quad i \in \{1, 2, \ldots, K\}, \\ s.t., & \\ & Ax \leq b, \\ & T(\xi)x = h((\xi), \\ & x \in \mathbb{N}. \end{cases} \tag{2.21}$$

Efficient solution concepts

In this paragraph we give some definitions of efficient solutions. These definitions are consistent with the objective approach and are determined by transforming a MOSLP problem into a deterministic MOLP problem, using one of the approaches presented in the previous chapter. Deterministic problem solving yields a set of efficient solutions considered to be efficient solutions of the initial problem.

Definition 2.3.1. *(**Average efficiency of solution** ([White, 1974]))*
$x^\star$ is said to be an efficient solution on average for the problem (2.20), if it is efficient for the problem

$$\begin{cases} \text{“min”} & \widetilde{F}_i = \widetilde{C}_i(\xi)x \quad i \in \{1, 2, \ldots, K\}, \\ \textit{such that} & \widetilde{F}_i = E(F_i) = \sum_{j=1}^{N} p_j F_{ij} = \sum_{j=1}^{N} p_j C_i(\xi_j)x = E(C_i(\xi)x). \end{cases}$$

Definition 2.3.2. *(**Efficient solution with minimum variance** ([White, 1974]))*
A solution $x^\star$ is said to be efficient with minimum variance for the problem (2.20) if it is efficient for the problem

$$\min_{x \in S}(\sigma_1^2(x), ..., \sigma_K^2(x)).$$

$\sigma_k^2(x)$ is the variance of the K^{th} objective

Definition 2.3.3. *(**Efficient solution with minimum risk** ([Stancu-Minasian, 1990]))*
A solution $x^\star$ is said to be efficient with minimum risk for the problem (2.20) if it is efficient for the problem

$$\max_{x \in \mathcal{S}} \left(P(F_1 \leq \alpha_1), \cdots, P(F_K \leq \alpha_K)\right).$$

Applying this criterion requires knowledge of the distribution functions or probability laws of stochastic objectives, as well as the aspiration level set by the decision-maker.

Definition 2.3.4. *(**Efficient solution with Probability** ([Stancu-Minasian, 1990]))*
$x^\star$ is an efficient solution with probality $(\alpha_1, \cdots, \alpha_K)$ for the problem (2.20) if there exists $\beta^\star = \left(\beta_1^\star, \cdots, \beta_K^\star\right)$ such that $(x^\star, \beta^\star)$ is efficient for the following program:

$$\begin{array}{ll} \min\limits_{(x,\beta)} & (\beta_1, \cdots, \beta_K) \\ s.t., & \\ & P\left(C^T(\xi) \leq \beta_i\right) \geq \alpha_i, \quad i \in \{1, 2, \ldots, K\}, \end{array}$$

where, $(\alpha_1, \cdots, \alpha_K)$ are the probability thresholds set a priori by the decision-maker.

2.3.2 Equivalent deterministic problem of MOSILP problem

Let us commence by considering a probability space (Ω, Ξ, P) that encompasses random variables controlled by a finite discrete probability distribution $(\xi^r, P^r), r \in \{1, \ldots, R\}$. This distribution comprises a limited number of realizations, denoted by R, with each realization corresponding to a distinct scenario.
For each realization ξ^r of ξ, we associate a criterion $f_i^r = C_i(\xi^r)x$, where $C_i(\xi^r)x$ represents the objective function for the i^{th} objective under the r^{th} scenario. Furthermore, we introduce a matrix $T(\xi^r)$ and a vector $h(\xi^r)$ to incorporate the diverse scenarios that impact the K objectives and stochastic constraints. By employing this approach, we can effectively integrate the uncertainty within the problem and reformulation of the problem (3.1) as follows:

$$(MOSILP_R)\begin{cases} \min\ f_i^r = C_i(\xi^r)x;\ \ i \in \{1, 2, \ldots, K\};\ r \in \{1, \ldots, R\}, \\ s.t., \\ \quad A\,x = b, \\ \quad T(\xi^r)\,x = h(\xi^r)\ \ r \in \{1, \ldots, R\}, \\ \quad x \in \mathbb{N}. \end{cases} \tag{2.22}$$

We adopt adopts the concept of recourse from single-criterion stochastic programming, as previously explored by [Van Slyke and Wets, 1969, Teghem, 1983, Teghem, 1990, Higle and Sen, 1991], using a deterministic recourse matrix W. To effectively address constraint violations, penalties denoted as $q^r = q(\xi^r)$ are assigned to each realization f_i^r, where $r \in \{1, \ldots, R\}$. Moreover, to integrate these penalties into the problem formulation, a recourse function $\mathbb{Q}(x, \xi^r)$ is introduced, as defined in (2.23). This function encapsulates the penalties associated with constraint violations and is incorporated into each criterion f_i^r.

$$\mathbb{Q}(x, \xi^r) = \min_y \{(q^r)^T y | Wy = h(\xi^r) - T(\xi^r)x, y \geq 0\}. \tag{2.23}$$

The deterministic equivalent of the problem presented in (2.22) takes the form of a multi-objective integer linear program (MOILP). This program can be defined as follows:

$$(MOILP_D)\begin{cases} \min\ \ \widetilde{f_i} + Q(x);\quad i \in \{1, 2, \ldots, K\}, \\ s.t., \\ \quad Ax = b, \ , \\ \quad x \in \mathbb{N}, \end{cases} \tag{2.24}$$

where:

$$\begin{cases} \widetilde{f_i} = \mathbb{E}[f_i^r] = \sum_{r=1}^{R} P^r f_i^r = \sum_{r=1}^{R} P^r C_i(\xi^r)x = \widetilde{C}_i x;\ i \in \{1, 2, \ldots, K\}. \\ Q(x) = \mathbb{E}[Q(x, \xi)] = \sum_{r=1}^{R} P^r (q^r)^T y^r. \end{cases}$$

2.3.3 Interactive solving methods for MOSLP

In recent years, there has been particular interest in solving stochastic multi-objective problems. The goal of these methods is to provide decision-makers with a decision-making tool. Most methods for solving stochastic multi-objective problems first transform the problems into deterministic ones and then solve them using iterative methods: PROTRADE method [Goicoechea, 1980], STRANGE method [Teghem Jr et al., 1986], and PROMISE method [Urli and Nadeau, 2004].

PROTRADE Method

This method was developed by [Goicoechea et al., 1976] and is inspired by the Step Model (STEM) for deterministic MOP by [Benayoun et al., 1971]. It focuses on problems where the coefficients c_k are random, the constraints may be random, linear, or nonlinear, and the random variables can be continuous or discrete. We can distinguish two steps in this method. The first step aims to determine the efficient solution, and for this, a function is defined to determine the weights involved in the definition of the auxiliary single-criterion problem that is optimized. The second step consists of reducing the admissible domain by adding constraints based on information. However, its use in solving problems is very challenging.

STRANGE Method

This method was developed by [Teghem Jr et al., 1986] to solve problems that arose during that period, specifically regarding the use of nuclear energy in Belgium. This method has three distinct phases.

1. **Step 1: Transformation of the Stochastic Problem into a Deterministic Problem**
   We assume that each objective $F_k(x)$ depends on a set of scenarios $\{s_k | s_k = 1, \cdots, S_k\}$, and each scenario is associated by experts with levels of plausibility p_{k,s_k}. Let c_{k,s_k} be the realization of the vector $c_k(\xi)$ under scenario s_k, such that

   $$P(C_k(\xi) = C_{k,s_k}) = p_{k,s_k}, \quad \text{and} \quad \sum_{s_k=1}^{S_k} p_{k,s_k} = 1$$

   Each criterion is multiplied for each scenario in order to obtain $\sum_{k=1}^{P} S_k$ new objectives, namely:

   $$F_{k,s_k} = C_{k,s_k} x, \quad s_k = 1, \cdots, S_k$$

For the coefficients of $T(\xi)$ and $h(\xi)$, multiple realizations are considered. Let $(T_r,\ h_r),\ r \in \{1,\ldots,R\}$ be these realizations and q_r their corresponding subjective probabilities:

$$P(T(\xi) = T_r, h(\xi) = h_r) = q_r,\ \ \sum_{r=1}^{R} q_r = 1$$

Stochastic programming with recourse allows the introduction into the constraints of $(m \times 1)$ vectors of variables $y^{(r)+},\ y^{(r)-}$, which measure the excess and deficit between $T_r x$ and h_r in the case of realization r

$$T_r x + y^{(r)+} + y^{(r)-} = h_r,\ \ r \in \{1,\ldots,R\}$$

and the new criterion to minimize is defined by:

$$F_{k+1} = \sum_{r=1}^{R} q_r(\beta^{(r)} y^{(r)-})$$

Where $\beta^{(r)}$ is a $(1 \times m)$ vector of possible penalties, allowing for different discrimination of constraint violations in each scenario r if necessary. The criterion F_{k+1} does not depend on any scenario, so we note

$$F_{k+1} = F_{\{(k+1),s_{k+1}\}},\ \ \text{with}\ \ S_{k+1} = 1$$

Thus, the deterministic multi-objective problem associated with problem (2.20) is written as:

$$\begin{cases} \text{"min"}\ F_{k,s_k} = c_{k,s_k} x\ \ k \in \{1,2,\ldots,K+1\},\ \text{and}\ s_k = 1,\cdots,S_k \\ \qquad (x, y^{(r)+}, y^{(r)-}) \in D^{(o)} \end{cases} \tag{2.25}$$

Where $D^{(o)}$ is defined by:

$$D^{(o)} = \left\{ \begin{array}{c} (x, y^{(r)+}, y^{(r)-}),\ r \in \{1,\ldots,R\} \mid T_r x + y^{(r)+} + y^{(r)-} = h_r \\ x \geq o,\ y^{(r)+} \geq o,\ y^{(r)-} \geq o \end{array} \right\}$$

2. **Step 2: Determination of the First Compromise**

The principle is based on the STEM method (see [?]), particularly for defining the payoff table.

- *Payoff Table*

  For each criterion (k, s_k), where $k \in \{1,2,\ldots,K+1\}$ and $s_k = 1,\cdots,S_k$, and for each scenario r, $r = 1,\cdots,R$, the following single-criterion problem is solved:

$$\begin{cases} \text{`min"}\ F_{k,s_k}(x) = C_{k,s_k}^T x\ \ k \in \{1,2,\ldots,K+1\},\ \ et\ \ s_k = 1,\cdots,S_k \\ T_r x + y^{(r)+} - y^{(r)-} = h_r \\ x \geq o,\ \ y^{(r)+} \geq o,\ \ y^{(r)-} \geq o \end{cases} \tag{2.26}$$

  $x_{k,s_k}^{(r)}$ are the optimal solutions found after solving the problem (2.26). Define $\widetilde{x}_{k,s_k}$ as the best solution $x_{k,s_k}^{(r)}$ from the point of view of criterion (k, s_k) for scenario r. This determines the component of the ideal point in the objective space:

$$M_{ks_k} = F_{k,s_k}(\widetilde{x}_{k,s_k}) = \min_{r \in \{1,\cdots,R\}} (F_{k,s_k}(x_{k,s_k}^{(r)}))$$

If $\widetilde{x}_{(k,s_k)}$ is a unique solution, the other coefficients in column (k, s_k) of the payoff table are obtained by evaluating the objectives (ℓ, t_i) at point $\widetilde{x}_{(k,s_k)}$.
However, if $\widetilde{x}_{(k,s_k)}$ is not a unique solution, to uniquely define this column of the payoff table, the values $Z_{(\ell,t_i)(k,s_k)}$ are obtained by solving the following problem:

$$\begin{cases} \min\ F_{\ell,t_i}(x) \\ (x, y^{(r)+}, y^{(r)-}) \in D^{(o)} \\ F_{k,s_k} = M_{k,s_k} \end{cases}$$

- *Weights Associated with Objectives*
The same principle as the STEM method by Benayoun, given

$$m_{(k,s_k)} = \max_{\ell,t_i} F_{(k,s_k)(\ell,t_i)}$$

Variation coefficients π_{k,s_k} are associated with each objective:

$$\pi_{k,s_k} = \frac{\alpha_{k,s_k}}{\sum_{k=1}^{p+1}\sum_{s_k=1}^{S_k}\alpha_{k,s_k}} \text{ with } \alpha_{k,s_k} = \frac{m_{k,s_k} - M_{k,s_k}}{m_{k,s_k}} \frac{1}{||C_k, s_k||}$$

- *First Compromise*
$\widetilde{x}^{(1)}$ is the optimal solution of the first compromise obtained by solving the following single-criterion problem:

$$(P)\begin{cases} \min\ M\delta - \sum_{k=1}^{p+1}\varepsilon_k \\ \sum_{s_k=1}^{S_k} p_{k,s_k}(C_{k,s_k}^T x - M_{k,s_k})\pi_{k,s_k} \le \delta - \varepsilon_k;\ \ k \in \{1, 2, \ldots, K+1\} \\ (x, y^{(r)+}, y^{(r)-}) \in D^{(o)},\ \ \varepsilon_k \ge 0,\ \ k \in \{1, 2, \ldots, K+1\} \end{cases}$$

M is a very large positive number.

3. Étape 3: **Interactive Phases**

- *Information Provided to the Decision-Maker*
For each compromise $\widetilde{x}^{(m)}$, the decision-maker receives three types of information:

(1) The values of the criteria (k, s_k) at the point $\widetilde{x}_{k,s_k}^{(m)}$:

$$F_{k,s_k}^{(m)} = F_{k,s_k}(\widetilde{x}_{k,s_k}^{(m)}),\ \ k \in \{1, 2, \ldots, K+1\};\ \ s_k = 1, \cdots, S_k$$

This information is provided to the decision-maker along with a range of variation for this criterion $[M_{k,s_k}, m_{k,s_k}]$. Thus, the decision-maker has a complete view of the impact of the compromise $\widetilde{x}_{k,s_k}^{(m)}$ on the objectives.

(2) The decision-maker may be interested in the average value of each criterion k:

$$\overline{F}_k^{(m)} = \sum_{s_k=1}^{S_k} p_{k,s_k} F_{k,s_k}^{(m)}$$

(3) Finally, to provide a simple measure of the possible variation of each objective, this information can be supplemented with a confidence threshold $\alpha_k^{(m)}$:

$$\alpha_k^{(m)} = P(C_k x > \overline{F}_k^{(m)})$$

This information is useful unless the number of scenarios is sufficiently large.

- *First Intervention of the Decision-Maker*
  Based on the provided information, the decision-maker must indicate whether they are satisfied or if they want to determine another compromise that is more satisfactory. In this case, they must designate an objective $(k, s_k)^\star$ to relax and an upper bound $\Delta_{(k,s_k)^\star}$ for the value $\overline{F}_{(k,s_k)^\star}^{(m+1)}$, so that this value lies within the interval $[\overline{F}_{(k,s_k)}^{(m+1)},\ \Delta_{(k,s_k)^\star}]$.

- *Calculation Phase*
  This phase involves a parametric analysis that fully explores the path indicated by the decision-maker, i.e., the consequences of all possible levels of relaxation of the criterion $(ks_k)^{(\star)}$ by solving the linear problem, so that the possible values for $F_{(ks_k)^\star}^{(m+1)}$ correspond to the values $\lambda \in [\underline{\lambda}, \overline{\lambda}]$ of the parameter λ.

$$(P_m^{(\star)}) \begin{cases} \min\ M\delta - \displaystyle\sum_{k=1}^{K+1} \varepsilon_k \\ \displaystyle\sum_{s_k=1}^{S_k} p_{ks_k}(C_{k,s_k}x - M_{k,s_k})\pi_{k,s_k} \leq \delta - \varepsilon_k \quad k \in \{1, 2, \ldots, K+1\} \\ (C, y^{(r)+}, y^{(r)-}) \in D^{(m)};\ \varepsilon_k \geq 0, \quad k \in \{1, 2, \ldots, K+1\} \end{cases}$$

To facilitate the decision-maker's task, the relative values of each criterion are presented on the same scale and graph.

- *Second Intervention of the Decision-Maker*
  Based on all the provided information, the decision-maker sets the level $\widetilde{\lambda} = [\underline{\lambda}, \overline{\lambda}]$ of relaxation they prefer, and thus the level of compromise is chosen by the decision-maker themselves.

$$\widetilde{x}^{m+1} = F^{(m+1)}(\widetilde{\lambda})$$

The Kataoka Method

Kataoka-type models with probabilistic constraints are given by [KATAOKA, 1962]:

$$(P) \begin{cases} \max(u_1, u_2, \cdots, u_p) \\ \text{s.t. } P\left(C_k^T(\xi)x \geq u_k\right) = \beta_k \quad k \in \{1, 2, \ldots, K\} \\ \quad P\left(T_i(\xi)x \leq h_i(\xi)\right) = \alpha_i \quad i \in \{1, \ldots, R\} \\ \quad x \geq 0 \end{cases} \tag{2.27}$$

$C_k(\xi)$ is a normal random vector with mean $\overline{C_k}$ and variance matrix V_k, and T_i, h_i is a multivariate normal random vector with mean $\mu_i \in \mathbb{R}^{n+1}$ and covariance matrix VC_i.
Problem (2.27) can be transformed into a nonlinear multi-objective problem using the Kataoka criterion in the following form:

$$(P_T) \begin{cases} \max\ u_k(x) = \overline{C}_k^T x - \phi^{-1}(\beta_k)\sqrt{x^T V_k x}, \quad k \in \{1, 2, \ldots, K\} \\ \text{s.t. } x \in D(\alpha_i), \quad i \in \{1, \ldots, R\} \end{cases} \tag{2.28}$$

such that:

$$D(\alpha_i) = \{x \in \mathbb{R} \mid m_i(x) + \phi^{-1}(\alpha_i)\sigma_i(x) \leq 0\}$$

with:

$$m_i(x) = \sum_{j=1}^{n} \mu_{ij} x_j - \mu_{i,n+1}, \ \ \sigma_i(x) = \sqrt{f^t V C_i f}, \ \text{ and } f = (x_1, \cdots, x_n, -1)^T$$

When $\beta_k \geq \frac{1}{2}$, the functions u_k are concave.
For solving nonlinear multi-objective problems, the weighted sum method is commonly used to reduce the problem to a single-objective problem, followed by the application of one of the single-criterion nonlinear programming methods, for example: [Geoffrion et al., 1972] method, [Wierzbicki, 1980] method, [Nakayama and Sawaragi, 1984], and [Steuer et al., 1986].

2.3.4 Solving methods of MOSILP problem

As far as we know, there are few exact methods in the literature. This is likely due to the mathematically ill-posed nature of the problem, despite its high realism. The solution of these problems has been particularly addressed interactively or approximately. We briefly mention in the following two sections the STRANGE-MOMIX method [Teghem, 1990] and the PROMISE method [Urli and Nadeau, 2004]. Finally, we explicitly detail the most recent method [Amrouche and Moulaï, 2012] by illustrating it with a didactic example.

PROMISE Method

The principle of this method is that the decision-maker is placed in a situation of partial uncertainty (or incomplete information), which allows only for determining the bounds of variation of random parameters. Unlike other methods, PROMISE does not impose probabilities on each scenario but leaves the choice to the decision-maker to specify a preference order on the scenarios. To solve a *MOILPS* problem, the proposed method considers all available information. Thus, instead of introducing a global penalty at the objective function level, it introduces a partial penalty function for each constraint and asks the decision-maker to manage the penalties while considering their preferences relative to the scenarios during the interactive phases of the method.

STRANGE-MOMIX Method

This method is a combination of the STRANGE and MOMIX methods [Teghem Jr and Kunsch, 1987]. Teghem modified the STRANGE method in its interactive phase by introducing the MOMIX method. This new method is called STRANGE-MOMIX. The principle of the method is that of the STRANGE method, but in the interactive phase, detailed information on a wide set of efficient solutions must be provided to the decision-maker.

Amrouche & Moulaï method

[Amrouche and Moulaï, 2012] propose an exact method to solve the MOSILP problem **??**, where their equivalent deterministic problem can be written as follows:

$$(P)\begin{cases} \max \quad \tilde{f_i} = f_i + \theta, \quad i \in \{1, 2, \ldots, K\}, \\ s.t., \\ \qquad x \in \mathcal{S}, \\ \qquad \theta \geq Q(x), \\ \qquad x \text{ Integer}. \end{cases} \tag{2.29}$$

Description of the method:
Starting with $\theta = -\infty$ without feasibility and optimality cuts. The objective $\mathbb{E}[C_1(\xi)x]$, or any other objective function $\mathbb{E}[C_i(\xi)x]\,; i \in \{1, 2, \ldots, K\}$ in place of $\mathbb{E}[C_1(\xi)x]$, is minimized under the deterministic constraints, and the domain of feasible integer solutions is partitioned into sub-domains using the principle of branching to search for integer solutions.
As soon as an integer solution x is found in a new domain, the method tests if for some realizations ξ_r, $r \in \{1, \ldots, R\}$, the second-stage problems yielded by this integer solution are not feasible. Then a feasibility cut 2.14 is introduced, and the problem is optimized again to obtain another integer feasible point x'. Using x', we solve the dual programs of problem **??** for all realizations ξ_r, $r \in \{1, \ldots, R\}$, and compute a recourse $Q(x')$. If $\theta < Q(x')$, an optimality cut 2.19 is introduced, and the problem (EP) is optimized, and the process is iterated. If not, the integer solution x' is compared to solutions already found, and hence, the set of all potentially efficient solutions E is updated.
An efficient cut described below is then added to eliminate integer solutions that are not efficient. To construct this cut, the growth directions of the criteria are used. The search for efficient solutions is made in each sub-domain created. A given domain contains no efficient solutions when none of the criteria can grow. This last one is said to be an explored domain. The search for efficient solutions is stopped only if all created domains are explored domains.

Illustrative Example: We present an example of a MOSILP problem with a structure similar to that of problem (P) [Abbas and Bellahcene, 2006, Amrouche and Moulaï, 2012], with $k = 3$, $n_0 = 4$, $m_0 = m = n = 2$.
Two deterministic constraints:

$$\begin{cases} -4x_1 + 2x_2 \geq -8, \\ x_1 + x_2 \leq 5. \end{cases} \Leftrightarrow \begin{cases} 4x_1 - 2x_2 \leq 8, \\ x_1 + x_2 \leq 5. \end{cases}$$

The fixed-recource matrix is given by:

$$W(\xi) = W = \begin{pmatrix} -2 & -1 & 2 & 1 \\ 3 & 2 & -5 & -6 \end{pmatrix},$$

Two scenarios ($R = 2$) affect the three objectives and the stochastic constraints. Assume that the two realizations of ξ are equally probability:
$\mathbb{P}(\xi^1) = \frac{1}{2}$, $\mathbb{P}(\xi^2) = \frac{1}{2}$.

$$q(\xi^1) = (1, 0, 6, 2)^T, \quad q(\xi^2) = (5, 3, 2, 1)^T.$$

The stochastic constraints are given by:

$$T(\xi^1) = \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}, \quad T(\xi^2) = \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix},$$

$$h(\xi^1) = \begin{pmatrix} 3 \\ 5 \end{pmatrix}, \quad h(\xi^2) = \begin{pmatrix} 6 \\ 1 \end{pmatrix}.$$

The problem becomes:

Scenario 1

$$(P1)\begin{cases} \min -9x_1 + 4x_2, \\ \min 3x_1 - 5x_2, \\ \min 8x_1 - 11x_2, \\ \text{s.t.}, \\ 4x_1 - 2x_2 \leq 8, \\ x_1 + x_2 \leq 5, \\ x_1 + 2x_2 = 3, \\ -2x_1 + x_2 = 5, \\ x_1, x_2 \in \mathbb{N}. \end{cases}$$

Scenario 2

$$(P2)\begin{cases} \min 3x_1 - 2x_2, \\ \min 7x_1 + x_2, \\ \min -4x_1 + 9x_2, \\ \text{s.t.}, \\ 4x_1 - 2x_2 \leq 8, \\ x_1 + x_2 \leq 5, \\ x_1 = 6, \\ 3x_1 + 4x_2 = 1, \\ x_1, x_2 \in \mathbb{N}. \end{cases}$$

We calculate the expectation of each objective function:

$$\widetilde{f_1} = \mathbb{E}[C_1(\xi)] = \mathbb{P}(\xi^1) C_1(\xi^1) + \mathbb{P}(\xi^2) C_1(\xi^2) = (-3, 1),$$

$$\widetilde{f_2} = \mathbb{E}[C_2(\xi)] = \mathbb{P}(\xi^1) C_2(\xi^1) + \mathbb{P}(\xi^2) C_2(\xi^2) = (5, -2),$$

$$\widetilde{f_3} = \mathbb{E}\left[C_3(\xi)\right] = \mathbb{P}(\xi^1) C_3(\xi^1) + \mathbb{P}(\xi^2) C_3(\xi^2) = (2, -1).$$

Formulating the equivalent deterministic problem:

$$(P)\begin{cases} \min -3x_1 + x_2, \\ \min 5x_1 - 2x_2, \\ \min 2x_1 - x_2, \\ \text{s.t.}, \\ 4x_1 - 2x_2 \leq 8, \\ x_1 + x_2 \leq 5, \\ x_1, x_2 \in \mathbb{N}. \end{cases}$$

Create the first node with the relaxed program with a single objective (P_0), $F = \{P_0\}$:

$$(P_0)\begin{cases} \min \mathbb{E}\left[C_1(\xi)\, x\right], \\ 4x_1 - 2x_2 + x_3 = 8, \\ x_1 + x_2 + x_4 = 5, \\ x_1, x_2, x_3, x_4 \geq 0. \end{cases}$$

$Eff = \emptyset$: Set of efficient integer solutions of problem (P).
The solution of problem (P_0) yields the following simplex tableau:

B	Rhs	x_3	x_4
x_1	3	$\frac{1}{6}$	$\frac{2}{6}$
x_2	2	$-\frac{1}{6}$	$\frac{4}{6}$
$\widetilde{f_1}$	7	$\frac{4}{6}$	$\frac{2}{6}$
$\widetilde{f_2}$	-11	$-\frac{7}{6}$	$-\frac{2}{6}$
$\widetilde{f_3}$	-4	$-\frac{3}{6}$	0

Table 2.1: Optimal simplex table at node 0

The obtained solution $x = (3, 2)$ is an optimal integer solution for the first stage of (P_0). To test the feasibility of the second-stage problem.
We solve the sub-problems (2.13) (corresponding to scenarios ξ^1 and ξ^2) with the solution $x = (3, 2)$:

$$h(\xi^1) - T(\xi^1)\, x = \begin{pmatrix} 3 \\ 5 \end{pmatrix} - \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}\begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} -4 \\ 9 \end{pmatrix}.$$

$$h(\xi^2) - T(\xi^2)\, x = \begin{pmatrix} 6 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix}\begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 3 \\ -16 \end{pmatrix}.$$

$$(P_{\sigma_1})\begin{cases} \max -4\sigma_1^1 + 9\sigma_1^2, \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0, \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0, \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0, \\ \sigma_1^1 - 6\sigma_1^2 \leq 0, \\ \sigma_1^1 + \sigma_1^2 \leq 1. \end{cases}$$

The maximum is achieved at: $\sigma_1^t = \left(\sigma_1^1, \sigma_1^2\right) = \left(\frac{2}{3}, \frac{1}{3}\right)$.

$$(P_{\sigma_2}) \begin{cases} \max 3\sigma_2^1 - 16\sigma_2^2, \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0, \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0, \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0, \\ \sigma_2^1 - 6\sigma_2^2 \leq 0, \\ \sigma_2^1 + \sigma_2^2 \leq 1. \end{cases}$$

The maximum is achieved at: $\sigma_2^t = \left(\sigma_2^1, \sigma_2^2\right) = (0, 0)$

$$\sigma_1^t \left[h\left(\xi^1\right) - T\left(\xi^1\right) x\right] = \left(\frac{2}{3}, \frac{1}{3}\right)\begin{pmatrix} -4 \\ 9 \end{pmatrix} = \frac{1}{3} \mid \ \sigma_2^t \left[h\left(\xi^2\right) - T\left(\xi^2\right) x\right] = (0, 0)\begin{pmatrix} 3 \\ -16 \end{pmatrix} = 0.$$

$\sigma_1^t \left[h\left(\xi^1\right) - T\left(\xi^1\right) x\right] > 0$, which means that the second-stage problem is not feasible for ξ^1. The feasiblity is crated by: $\frac{5}{3}x_2 \geq \frac{11}{3}$, and add this cut to the first-stage problem. The solution of the main problem with the new cut added yields the following simplex tableau:

B	Rhs	x_4	x_5
x_1	$\frac{14}{5}$	1	$\frac{1}{5}$
x_2	$\frac{11}{5}$	0	$-\frac{1}{5}$
x_3	$\frac{6}{5}$	-4	$-\frac{6}{5}$
$\widetilde{f_1}$	$\frac{31}{5}$	3	$\frac{4}{5}$
$\widetilde{f_2}$	$-\frac{48}{5}$	-5	$-\frac{7}{5}$
$\widetilde{f_3}$	$-\frac{17}{5}$	-2	$-\frac{3}{5}$

Table 2.2: Optimal simplex table after feasibility cut

The minimum is achieved at $x = \left(\frac{14}{5}, \frac{11}{5}\right)$.

$x = \left(\frac{14}{5}, \frac{11}{5}\right)$ is optimal for the first stage of the program, but it is not integer. Therefore, we perform separation with respect to variable x_1 and obtain two sub-problems. The separation process begins:
Node 1: $S_1' = \left\{x \in S_0 : x_1 \geq \left\lfloor \frac{14}{5} \right\rfloor + 1\right\}$ and $x_1 \geq 3 \Longleftrightarrow x_4 + \frac{3}{5}x_5 + x_6 = -\frac{1}{5}$.
We add this constraint and solve the new program (P_1), but it is infeasible, so the corresponding node is fathomed.
Node 2: $S_1 = \left\{x \in S_0 : x_1 \leq \left\lfloor \frac{14}{5} \right\rfloor\right\}$ and $x_1 \leq 2 \Longleftrightarrow -x_4 - \frac{3}{5}x_5 + x_6 = -\frac{4}{5}$.
This constraint is added and the dual simplex method is applied. The integer optimal solution of program (P_2) is obtained in the following table:

B	Rhs	x_5	x_6
x_1	2	0	1
x_2	$\frac{11}{5}$	$-\frac{1}{5}$	0
x_3	$\frac{22}{5}$	$-\frac{2}{5}$	-4
x_4	$\frac{4}{5}$	$\frac{1}{5}$	-1
$\widetilde{f_1}$	$\frac{19}{5}$	$\frac{1}{5}$	3
$\widetilde{f_2}$	$-\frac{28}{5}$	$-\frac{2}{5}$	-5
$\widetilde{f_3}$	$-\frac{9}{5}$	$-\frac{1}{5}$	-2

Table 2.3: Optimal simplex table at node 2

The minimum is achieved at $x = \left(2, \frac{11}{5}\right)$, but it is not integer.The separation is then done with respect to variable x_2, yielding two sub-problems:
Node 3: $S_2' = \left\{x \in S_1 : x_2 \leq \left\lfloor \frac{11}{5} \right\rfloor\right\}$ and $x_2 \leq 2 \Rightarrow x_2 + x_7 = 2 \Rightarrow \frac{3}{5}x_5 + x_7 = -\frac{1}{5}$.
We add this constraint and solve the new program (P_3), but it is infeasible. Thus, this node is fathomed.
Node 4: $S_2 = \left\{x \in S_1 : x_2 \geq \left\lfloor \frac{11}{5} \right\rfloor + 1\right\}$ and $x_2 \geq 3 \Rightarrow -x_2 + x_7 = -3 \Rightarrow \frac{-3}{5}x_5 + x_7 = -\frac{4}{5}$.
This constraint is added and the dual simplex method is applied. The integer optimal solution of program (P_4) is obtained in the following table is $x = (2, 3)$.

B	Rhs	x_6	x_7
x_1	2	1	0
x_2	3	0	-1
x_3	6	-4	-2
x_4	0	-1	1
x_5	4	0	-5
$\widetilde{f_1}$	3	3	1
$\widetilde{f_2}$	-4	-5	-2
$\widetilde{f_3}$	-1	-2	-1

Table 2.4: Optimal simplex table at node 4

To test the feasibility of the second-stage problem, we solve the sub-problems (2.13) (corresponding to scenarios ξ^1 and ξ^2) with the solution $x = (2, 3)$:

$$h\left(\xi^1\right) - T\left(\xi^1\right)x = \begin{pmatrix} 3 \\ 5 \end{pmatrix} - \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}\begin{pmatrix} 2 \\ 3 \end{pmatrix} = \begin{pmatrix} -5 \\ 6 \end{pmatrix}.$$

$$h\left(\xi^2\right) - T\left(\xi^2\right)x = \begin{pmatrix} 6 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix}\begin{pmatrix} 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 4 \\ -17 \end{pmatrix}.$$

$$\begin{cases} \max -5\sigma_1^1 + 6\sigma_1^2, \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0, \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0, \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0, \\ \sigma_1^1 - 6\sigma_1^2 \leq 0, \\ \sigma_1^1 + \sigma_1^2 \leq 1. \end{cases}$$

The maximum is achieved at: $\sigma_1^t = \left(\sigma_1^1, \sigma_1^2\right) = (0, 0)$.

$$\begin{cases} \max 4\sigma_2^1 - 17\sigma_2^2, \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0, \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0, \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0, \\ \sigma_2^1 - 6\sigma_2^2 \leq 0, \\ \sigma_2^1 + \sigma_2^2 \leq 1. \end{cases}$$

The maximum is achieved at: $\sigma_2^t = \left(\sigma_2^1, \sigma_2^2\right) = (0, 0)$. $\sigma_1 = \sigma_2 = 0$, which means that the new solution $x = (2, 3)$ obtained in Table 6 is the feasible solution for the second stage problem.

To test the optimality of $x = (2, 3)$ for the second-stage problem, we solve the subproblems (2.16) (corresponding to scenarios ξ^1 and ξ^2) with the solution $x = (2, 3)$:

$$\begin{cases} \max -5\pi_1^1 + 6\pi_1^2, \\ -2\pi_1^1 + 3\pi_1^2 \leq 1, \\ -\pi_1^1 + 2\pi_1^2 \leq 0, \\ 2\pi_1^1 - 5\pi_1^2 \leq 6, \\ \pi_1^1 - 6\pi_1^2 \leq 2. \end{cases}$$

The maximum is achieved at: $\pi_1^t = \left(\pi_1^1, \pi_1^2\right) = \left(-1, -\frac{1}{2}\right)$.

$$\begin{cases} \max 4\pi_2^1 - 17\pi_2^2, \\ -2\pi_2^1 + 3\pi_2^2 \leq 5, \\ -\pi_2^1 + 2\pi_2^2 \leq 3, \\ 2\pi_2^1 - 5\pi_2^2 \leq 2, \\ \pi_2^1 - 6\pi_2^2 \leq 1. \end{cases}$$

The maximum is achieved at: $\pi_2^t = \left(\pi_2^1, \pi_2^2\right) = (1, 0)$.

$$Q\left(x, \xi^1\right) = \pi_1^t \left[h\left(\xi^1\right) - T\left(\xi^1\right) x\right] = 2,$$

$$Q\left(x, \xi^2\right) = \pi_2^t \left[h\left(\xi^2\right) - T\left(\xi^2\right) x\right] = 4,$$

$$Q\left(x\right) = \frac{1}{2} Q\left(x, \xi^1\right) + \frac{1}{2} Q\left(x, \xi^2\right) = 3.$$

$$\theta = -\infty < Q\left(x\right).$$

We introduce the optimality cut of the form (2.19), $\theta \geq \frac{1}{4} - \frac{1}{2} x_1 + \frac{5}{4} x_2$, and we resolve the previous problem.

Table 5

B	Rhs	x_6	x_7	s_1
x_1	2	1	0	0
x_2	3	0	-1	0
x_3	6	-4	-2	0
x_4	0	-1	1	0
x_5	4	0	-5	0
θ	3	$-\frac{1}{2}$	$-\frac{5}{4}$	-1
$\widetilde{f_1}$	3	3	1	0
$\widetilde{f_2}$	-4	-5	-2	0
$\widetilde{f_3}$	-1	-2	-1	0

Table 2.5: Optimal siplex table after optimality cut

Thus, $x^1 = (2, 3)$ is an efficient integer solution to the program (P), with the criterion vector $\left(\mathbb{E}\left[C_1\left(\xi\right)\right] x^1, \mathbb{E}\left[C_2\left(\xi\right)\right] x\right.$ $(-3, 4, 1)$ and a penalty $\theta^1 = 3$. Thus, the set of efficient solutions is $Eff = \{(2, 3)\}$.

- $N_3 = \{6, 7\}$, $H_3 = \{6, 7\} \neq \emptyset$

$S_4 = \left\{x \in S_3 \mid \sum_{j \in H_3} x_j \geq 1\right\} = \left\{x \in S_3 \mid x_6 + x_7 \geq 1\right\}$ and $x_6 + x_7 \geq 1 \Leftrightarrow -x_6 - x_7 + x_8 = -1$.
We add this efficient cut of the form (1.17) $-x_6 - x_7 + x_8 = -1$ and resolve the corresponding node problem.

The MOSILP algorithm continues with the same steps to find the final efficient set and terminates as all nodes are explored. The set of all efficient solutions and their penalties for the program (P) are given in the following table 2.6:

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
x^i	$(2, 3)$	$(1, 3)$	$(1, 4)$	$(0, 4)$	$(0, 5)$
θ^i	3	$\frac{7}{2}$	$\frac{19}{4}$	$\frac{21}{4}$	$\frac{13}{2}$
f_i'	$(-3, 4, 1)$	$(0, -1, -1)$	$(-1, 3, 2)$	$(4, -8, -4)$	$(5, -10, -5)$
$\tilde{f}_i = f_i' + \theta^i$	$(0, 7, 4)$	$\left(\frac{7}{2}, \frac{5}{2}, \frac{5}{2}\right)$	$\left(\frac{23}{4}, \frac{5}{4}, \frac{11}{4}\right)$	$\left(\frac{37}{4}, -\frac{11}{4}, \frac{5}{4}\right)$	$\left(\frac{23}{2}, -\frac{7}{2}, \frac{3}{2}\right)$

Table 2.6: Final table of all afficient solution Eff

Algorithm: All steps of the previous method are presented in the algorithm **??**:

Algorithm 11: Amrouche and Moulaï Method

Data: Initialization: $\theta = -\infty$, $l = 0$,
$M = m_0, N = n_0 = 0, E = \infty$
Result : E: The final efficient set of the problem (2.29).
while *As long as a unfathomed node l exists in the tree* **do**
  Solve the problem (P_l):

$$\begin{aligned} \min \quad & \mathbb{E}[C_1(\xi)x] \\ \text{s.t.} \quad & x \in \mathcal{S}_l \end{aligned}$$

  Using simplex or dual simplex method;
  if *the problem (11) has a feasible solution x^l* **then**
    if *x^l is integer* **then**
      Feasibility test:
      Solve the program (2.13);
      if $\sigma_r^t[h(\xi^r) - T(\xi^r)x^l] > 0$ **then**
        Add the feasibility cut (2.14) to the successors of l;
      else
        Optimality test:
        Solve the program (2.16), calculate Q(x);
        if $Q(x^l) > \theta^l$ **then**
          Add the optimality cut (2.19) to the successors of l;
        else
          Efficiency
          if *$f(x^l)$ is not dominated by $f(x)$ for all $x \in E$* **then**
            $E = E \cup \{x^l\}$
          else
            $E = E\ \{x^l\} \cup \{x\}$
          end
          Update set $\mathcal{S}$.
          Construct the sets $\mathcal{H}_l$;
          if $\mathcal{H}_l$ **then**
            Fathom the node l;
          else
            Add the efficient cut (3.11) to the successors of l;
          end
        end
      end
    else
      Branching process
      Choose an index j such that α_j is the most fractional number.
      Split the program P^l into two sub programs, by adding respectively the constraints:
      $x_j \leq \lfloor \alpha_j \rfloor$ and $x_j \geq \lfloor \alpha_j \rfloor + 1$.
      To obtain (P^{l_1}) and (P^{l_2}) $(l_1 \neq l_2$ and $l_1 \geq l+1,\ l_2 \geq l+1)$;
    end
  else
    The corresponding node l is fathomed;
  end
end

2.4 Optimization Over an Efficient Set of a Multi-Objective Stochastic Integer Linear Programming Problem.

2.4.1 Problem formulation

The main problem studied is described by:

$$(P_S)\begin{cases}\min \widetilde{\phi} = & d(\xi)x, \\ s.t., & \\ & x \in \mathcal{X}_E, \end{cases} \tag{2.30}$$

where, $d(\xi)$ is the random vector for the linear preference objective function. And $\mathcal{X}_E$ is the efficent solutions set of the MOSILP problem.
Let $\left(\widetilde{P}_E\right)$ be the following equivalent deterministic problem:

$$\left(\widetilde{P}_E\right)\begin{cases}\min \widetilde{\phi} = & \mathbb{E}\left(d\right)x, \\ s.t., & \\ & x \in \mathcal{X}_E. \end{cases} \tag{2.31}$$

$\mathbb{E}\left(d\right)$ is the row vector of dimension n, where the j^{th} component is $\mathbb{E}\left(d_j\right)$.
The relaxed problem is given by:

$$(P_R)\begin{cases}\min \widetilde{\phi} = & \mathbb{E}\left(\phi(x)\right), \\ s.t., & \\ & x \in \mathcal{S} = \{x \in \mathbb{R}^n \mid Ax \leq b, x \in \mathbb{N}^n\} \end{cases} \tag{2.32}$$

2.4.2 Mabrek & Chabaane method

Description of the method

The method consists of three main parts: the first part involves solving the relaxed deterministic problem (2.32). Next, the method performs the feasibility test using the problem (2.2.1). If the solution is feasible, it moves on to the optimality test (2.15). Otherwise, a feasibility cut (2.14) must be added until the solution found is feasible for all scenarios of the problem. Then, the method calculates the penalty for violating the random constraints using the expression (2.16) to test the optimality of the solution. If the solution is not optimal, the optimality cut is added.

The method passes to the next step, which is testing whether the obtained solution x^o is efficient or not by solving the problem $(EQ\,(x^o))$ (1.12). For an efficient solution, we search for an equivalent solution, if it exists, by solving the problem (P_{equiv}) given by equation (2.33)

$$\left(P_{equiv}(\overline{x})\right)\begin{cases}\min\ \widetilde{\phi}(x),\\ s.t.,\\ x\in\mathcal{S},\\ \widetilde{C}x=\widetilde{C}\overline{x}.\end{cases}\tag{2.33}$$

Then, the feasible domain is reduced by adding constraints to eliminate infeasible solutions (dominated solutions) and solving the problem 2.34

$$(P_R^{\ell})\equiv\min\{dx\mid x\in\mathcal{S}-\bigcup_{s=1}^{\ell}\mathcal{S}_s\},\tag{2.34}$$

where, $\mathcal{S}_s=\{x\mid x\in\mathbb{Z}_+^n, Cx\leq Cx^s\}$ and $\{x^s\}_{s=1}^{\ell}$ are the solutions obtained in the iterations $\{1,2,\cdots,\ell-1\}$ respectively.

$$\mathcal{S}-\bigcup_{s=1}^{\ell}\mathcal{S}_s=\left\{\begin{array}{l}C_ix\leq(C_ix^s-1)y_i^s+M_i(1-y_i^s),\\ i\in\{1,2,\ldots,K\}, s\in\{1,\cdots,\ell\},\\ \\ y_i^s\in\{0,1\},\\ i\in\{1,2,\ldots,K\}, s\in\{1,\cdots,\ell\},\\ \\ \sum_{i=1}^{p}y_i^s\geq 1;\ s\in\{1,\cdots,\ell\},\\ x\in\mathcal{S}.\end{array}\right\}\tag{2.35}$$

M_i is the upper bound of the i^{th} objective function.
Finally, the optimal efficient solution is evaluated based on the main linear criterion and updated as long as the current domain is non-empty.

Algorithm

All steps of the method are presented in the following algorithm.

Algorithm 12: Optimizing a Linear Function over an Integer Efficient Set

Inputs

$\downarrow K, m, m_1, n, n_1$: The dimensions of the problem;

$\downarrow S, Pr$: Number of scenarios and their probabilities;

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

$\downarrow T_{(m_1\times n)}(sc), h_{(m_1\times 1)}(sc)\ \forall sc \in \{1,\cdots,R\}$: Parameters of the random constraints;

$\downarrow d_{(1\times n)}(sc),\ \forall sc \in \{1,\cdots,R\}$: Vector of the main random criterion;

$\downarrow Cr_{(K\times n)}(sc),\ \forall sc \in \{1,\cdots,R\}$: Matrix of random criteria;

$\downarrow W_{(m_1\times n_1)}$: The recourse matrix;

$q'_{(1\times n_1)}$: Penalties for violating the random constraints;

Outputs

$\uparrow X_{opt}$: Optimal solution of the problem (P_E); $\uparrow \mathbb{E}\left(\widetilde{\phi}_{opt}\right)$: Optimal value of the criterion $\widetilde{\phi}$;

Initialization $\mathbb{E}\left(\widetilde{\phi}_{opt}\right) \leftarrow +\infty$, $\ell \leftarrow 1$, $End \leftarrow false$, $\theta \leftarrow -\infty$, and $D^1 \leftarrow D$;

Solve the relaxed deterministic problem

$\uparrow x_o, \uparrow \mathbb{E}(f_o) =$ Pb_relaxed $(\downarrow d, \downarrow A, \downarrow b)$;

if *the problem has no feasible solution* **then**

  the problem (P) is not feasible: **Terminate**;

else

  while *End=false* **do**

    Let x^ℓ be a solution of $\left(P_R^\ell(D^\ell)\right)$

    Feasibility and optimality test for the solution

    Solve the problem (1.12); Ψ is the optimal solution of the criterion;

    if $\Psi \neq o$ **then**

      x^ℓ is not efficient, $\overline{x}^\ell$, the optimal solution of $\left(P\left(x^\ell\right)\right)$, is efficient;

      Feasibility and optimality test for the solution $\overline{x}^\ell$;

      $X_{opt} \leftarrow \overline{x}^\ell$, $\phi_{opt} \leftarrow \widetilde{\phi}(\overline{x}^\ell)$, $\ell \leftarrow \ell + 1$;

    else

      x^ℓ is an efficient solution, $X_{opt} = x^\ell$, $\widetilde{\phi}_{opt} = \widetilde{\phi}(x^\ell)$, $\ell \leftarrow \ell + 1$;

    end

    $D^\ell \leftarrow D^\ell - \bigcup_{s=1}^{\ell-1} D_s$ *and solve* $\left(P_R^\ell(D^\ell)\right)$;

    if $D^\ell = \varnothing$ ***or*** $\widetilde{\phi}(x^\ell) > \widetilde{\phi}_{opt}$ **then**

      X_{opt} is an optimal efficient solution with value $\widetilde{\phi}_{opt}$;

      $End \leftarrow true$;

    else

      $End \leftarrow false$;

    end

  end

end

PART

II

Contributions

Chapter

3 Optimizing a linear function over the stochastic efficient set

"Life is about decisions. "
"La vie est une question de décisions."
Mattias ehrgott.
"The essence of mathematics lies in its freedom."
"L'essence des mathématiques réside dans sa liberté."
Georg Cantor

Abstract: In this chapter, we present a new exact method called "A Branch and Cut (B & C) based Method (BCM)" for optimizing a linear function over the efficient set of a MOSILP problem. The proposed method combines two principles: the first is called the L-method while the second is an adaptation of the branch-and-bound procedure. This adaptation is reinforced by efficient cuts and tests, which allow the method to eliminate a large number of inefficient solutions in the search tree. A didactic example is given to detail the different steps of the method. Following this, the method is tested using a set of randomly generated instances, where the results obtained are compared with the most recent method. In addition, the experimental study shows that the proposed method remains competitive and delivers satisfactory results.

3.1 Introduction

MOLP problems have long attracted growing interest due to their relevance in many real-world situations. This attention has led to significant methodological and research progress, as numerous experts and researchers have made significant contributions to a multitude of projects addressing these real-world challenges such as plans for the treatment of cancer by radiotherapy [Obal et al., 2013], supply chain network design [Liao et al., 2017], post-departure aircraft rerouting problem [Bongo and Sy, 2023], risk assessment in supply chain management [Vafadarnikjoo et al., 2023]. Nevertheless, certain real-life

scenarios include variables that are not deterministic and are not predictable, referred to as stochastic data. This gives rise to challenges in solving MOSLP problems. These variables introduce uncertainty, and their values are not known with precision at the time of decision-making. This uncertainty can stem from external factors like weather conditions or decisions made by other individuals. In addition, it emerges from estimating parameters through statistical samples or treating them as random variables based on the model design. Moreover, the MOSLP class has become a practical tool for modeling and resolving diverse applications and challenges. where several technical methods have been developed to deal with such problems: green supply chain design [Moayedi and Sadeghian, 2023], hospital bed planning [Ben Abdelaziz and Masmoudi, 2012], forward/reverse logistic network design [Ramezani et al., 2013], spatial conservation planning [Sierra-Altamiranda et al., 2020].
It is crucial to highlight that balancing the conflicts between these multiple objectives within MOLP problems has led to a significant challenge known as "the identification of the efficient solution set" or "finding all non-dominated points" as has been mentioned in some works. This challenge has given rise to a considerable amount of research and practical application, including significant contributions made by [Ecker and Kouada, 1978, Boland et al., 2015, Rasmi and Türkay, 2019, Tamby and Vanderpooten, 2021]. As far as stochastic cases, we mention the notable work proposed by [Abbas and Bellahcene, 2006, Amrouche and Moulaï, 2012]. Despite the significant role played by finding the set of efficient solutions to solve many real-life problems. However, generating this whole efficient set in some cases can be quite challenging for practical reasons, primarily due to the substantial computational burden linked to these algorithms. In addition, decision-makers face often a substantial set of efficient solutions which puts them in challenging situations, where they must choose the best solution from this set. One way to overcome such cases involves the use of optimization techniques. Through these techniques, decision-makers can identify trade-offs between various objectives, adjust their choices based on different circumstances and preferences, and optimize these preferences over this efficient set to evaluate and select the favorite solution without generating the whole set of efficient solutions. In the literature, this approach is referred to as "*optimization over efficient set problems*" and falls into the NP-hard category. Optimization over the efficient set holds undeniable mathematical and applied significance in multi-objective decision-making. This problem has been extensively studied, where it was first considered by [Philip, 1972], who made a noteworthy contribution. Over time, this domain has attracted the attention of numerous experts due to its crucial implications for various research fields. It has been studied in several notable works, such as [Benson, 1984, Ecker and Song, 1994]. Based on the previous works, [Abbas and Chaabane, 2006] develop the first method for optimizing a linear function over the efficient set without finding all non-dominated points. The proposed method may provide a shorter way to the optimal one but it generates several dominant solutions. In addition, they do not present any computational study. Afterward, [Jorge, 2009] proposes an exact algorithm to optimize a linear function over the integer efficient set of a MOILP problem, the algorithm defined a sequence of increasingly constrained single-objective integer problems to eliminate not only all previously generated efficient solutions but also any other feasible solutions whose criteria vectors are dominated. The performance of the algorithm is tested on randomly generated instances of the MOILP problem. Later, [Ouail et al., 2017]

proposed a branch and bound-based method to optimize a linear function over the efficient set of a MOILP problem. Two types of cuts are used to avoid searching in areas where no efficient solutions are contained, and deleting domains not containing the optimal solution.The experimentation results show that the proposed method outperforms the Jorge method by 89% in terms of CPU time. Later on, [Boland et al., 2017] Describe a new algorithm to solve the same problem by modifying the algorithm of Jorge. The authors developed a novel criteria space decomposition scheme and search procedure that limits the number of subspaces that are created and the number of sets of disjunctive constraints required to define the single-objective integer program that searches for a non-dominated solution. The computational study shows that the new algorithm outperforms Jorge algorithm. Following this, [Lokman, 2021] develop two algorithms to optimize a linear function over the non-dominated set of the multi-objective integer programming (MOIP) problem. The algorithms iteratively generate non-dominated points and reduce the feasible set by excluding not only the dominated regions but also the inferior regions concerning the linear function. In addition, the reduced feasible set is decomposed into subsets and a search is conducted over all these subsets to find a new point. The performances of the algorithms are tested and compared with the existing studies on different-sized multi-objective combinatorial optimization problems. Recently, [Belkhiri et al., 2022] proposed a new methodology to search for an efficient extreme point that optimizes a linear function over the set of efficient extreme points of a convex polyhedron. The proposed methodology uses a branch and bound-based technique, in which, at each node of the search tree, new customized bounds are established to delete uninteresting areas from the decision space. A comparative study shows that the methodology outperforms the most recent.

When it comes to the stochastic case, there is a notable rarity of works dealing with such problems, because of their difficulty due to the stochastic nature of these problems. To the best of our knowledge, the only work that deals with this type of problem was discussed by [Chaabane and Mebrek, 2014], where the authors proposed an exact method to solve a single objective linear function over the efficient set of a MOSILP problem. This method will be referred to as CMM in the remainder of this article. Furthermore, the principle of their method can be described as a combination of two techniques. The first one is the L-shaped method, an iterative algorithm designed for solving two-stage stochastic linear programming problems, such as [Li and Grossmann, 2018, Dos Santos and Oliveira, 2019]. The second technique is a combined method developed by [Djamal and Marc, 2010] that uses the ideas of [Sylva and Crema, 2004] to reduce the feasible region gradually by eliminating infeasible solutions. This approach also incorporates the concept of searching for the equivalent efficient solution. Furthermore, the proposed method may provide a pathway to the optimal solution. However, it's worth noting that as the number of objective functions increases, the scale of the problem also expands. In contrast, our contribution relies on another concept, namely the efficiency cut. In each step of the method, we use only one efficient cut to eliminate infeasible solutions, thereby maintaining the scalability of the problem within reasonable bounds.

In this chapter, we present a new branch-and-cut method designed to optimize a linear stochastic function over the efficient set of the MOSILP problem. Our method is based on the branch-and-bound

technique which has been strengthened by the incorporation of efficiency tests and cuts during the process. This allows the method to prune a substantial number of nodes in the search tree to avoid many inefficient solutions. Moreover, The search process in our method combines two techniques, the one previously mentioned and the known L-shaped method which can handle large-scale problems with numerous scenarios. Instead of solving the entire problem at once, it decomposes the original stochastic problem into two manageable levels: a master deterministic problem to provide an initial solution and subproblems that are solved iteratively with incorporating information from realized scenarios to improve the solution [Van Slyke and Wets, 1969, Kall et al., 1994].

3.2 Definitions and preliminaries

This section introduces some basic notations and definitions needed to understand the contents of this article. It is divided into two principal parts. The first part describes the MOSILP problem in the general context, as well as the definition of our main problem, called "A branch and cut based method for optimizing a linear function over the stochastic efficient set". Following this, the methodology for transforming the stochastic problem into its deterministic equivalent is explained. The second part discusses the various methods and strategies incorporated in the proposed method. It begins with a brief detailed description of the L-shaped method. Next, the section looks at the modifications adopted in our algorithm by clarifying the definition of efficiency in a broader context [Ecker and Kouada, 1978], followed by a detailed explanation of the efficient cutting strategy [Ouail et al., 2017].

3.2.1 Problem formulation

The MOSILP problem with random variable coefficients in objective functions and/or some constraints is formulated as follows:

$$(MOSILP)\begin{cases} \min\ f_i = C_i(\xi)x; \quad i = \{1, 2, \ldots, K\}, \\ s.t., \\ \quad A\,x = b, \\ \quad T(\xi)\,x = h(\xi), \\ \quad x \in \mathbb{N}, \end{cases} \tag{3.1}$$

where, $K \geq 2$ represents the number of objectives, ξ represents the possible realization. $C_i(\xi), T(\xi), h(\xi)$ are random matrices of dimensions $(1 \times n)$, $(m1 \times n)$ and $(m1 \times 1)$ respectively, with a known joint probability distribution which is not influenced by the choice of the decision x and defined on a probability space (Ω, Ξ, P). The vector b of size $m \times 1$ and the real matrix A of size $m \times n$ are deterministic, whereas the decision vector variables x belong to $\mathbb{R}^n$ which are to be determined.

Our main objective is to solve the problem of optimizing a linear preference function over the efficient set of the MOSILP problem (3.1). It can be formulated as follows:

$$(P)\begin{cases} min \quad \phi(x) = d(\xi)x, \\ s.t., \\ \qquad x \in \mathcal{X}_E, \end{cases} \tag{3.2}$$

where, $d(\xi)$ is a random line vector of dimension $(1 \times n)$ and $\mathcal{X}_E$ denotes the efficient solution set of the MOSILP problem (3.1).

Before starting to solve the problem (3.2), we need to pass through an important step, which is the transformation of a stochastic problem into an equivalent deterministic problem. This strategy has been used most in the work of [Caballero et al., 2001, Caballero et al., 2004]. This transformation is necessary because stochastic problems contain uncertain parameters, which cannot be directly optimized. It will be discussed in detail in the next subsection.

Equivalent deterministic problem

The deterministic equivalent of the problem presented in (2.22) takes the form of a MOILP. It can be defined as follows (see, Subsection 2.3.2):

$$(MOILP_D)\begin{cases} \min \quad \widetilde{f_i} + Q(x); \quad i = \{1, 2, \ldots, K\}, \\ s.t., \\ \quad Ax = b, \\ \quad x \in \mathbb{N}, \end{cases} \tag{3.3}$$

where:

$$\begin{cases} \widetilde{f_i} = \mathbb{E}[f_i^r] = \sum_{r=1}^{R} P^r f_i^r = \sum_{r=1}^{R} P^r C_i(\xi^r)x = \widetilde{C_i}x; \ i = \{1, 2, \ldots, K\}, \\ Q(x) = \mathbb{E}[Q(x, \xi)] = \sum_{r=1}^{R} P^r (q^r)^T y^r. \end{cases}$$

In the same way (see, Subsection 2.3.2), when considering the linear preference objective, we express $\phi^r = d(\xi^r)x$, where $d(\xi^r)x$ represents the preference objective in the rth scenario. In this context, the deterministic equivalent of our main problem is defined as follows:

$$(P_D)\begin{cases} \min \quad \widetilde{\phi} + Q(x), \\ s.t., \\ \quad x \ \in \mathcal{X}_{ED}, \end{cases} \tag{3.4}$$

where:

$$\begin{cases} \widetilde{\phi} = \mathbb{E}[\phi^r] = \mathbb{E}[d(\xi)x] = \sum_{r=1}^{R} P^r d(\xi^r)x = \widetilde{d}x, \\ Q(x) = \mathbb{E}[Q(x,\xi)] = \sum_{r=1}^{R} P^r (q^r)^T y^r. \end{cases}$$

Here, $\mathcal{X}_{ED}$ signifies the set of efficient solutions for the deterministic equivalent problem (3.3). The relaxed problem of our main problem can be defined as follows:

$$(P_R) \begin{cases} \min & \widetilde{\phi} = \widetilde{d}x + \mathbb{E}[Q(x,\xi)], \\ s.t., & x \in \mathcal{S} = \{x \in \mathbb{R}^n / Ax = b \, ; x \in \mathbb{N}^n\}, \end{cases} \tag{3.5}$$

3.2.2 The principle of the combined technique

In what follows, we present the efficient cut strategies incorporated in the proposed algorithm, for a brief overview of the basic principle underlying the L-shaped method and the efficiency test (see, Subsections (2.2.4),(1.3.1), respectively).

Efficient Cut

In each iteration l of the optimization process, the efficient cut is utilized to eliminate all inefficient solutions from the feasible region and remove areas that do not improve the value of the utility function. Assuming that x^l is the l^{th} generated optimal solution of main problem, the following definitions and notations are used:

- $\mathcal{B}_l$ is the indexes sets of the basic variables of x^l.
- $\mathcal{N}_l$ is the indexes sets of the non-basic variables of x^l.

The decrease direction of each criterion $\widetilde{f_i}, i \in \{1, ..., K\}$ of the problem (3.3) is determined by using their reduced gradient vectors. On the other hand, the method uses this information to build an efficient cut to remove integer solutions that are not efficient for the problem (3.3) and determine an efficient new integer solution. To do so, we define a new set of all decreasing directions of the criteria as follows:

- $\mathcal{H}_l = \left\{ j \in \mathcal{N}_l / \exists i \in \{1, 2, \ldots, K\}, \text{ with } \bar{C}_i^j < 0 \right\} \bigcup \left\{ j \in \mathcal{N}_l / \bar{C}_i^j = 0; \forall i \in \{1, 2, \ldots, K\} \right\}.$

Here, $\bar{C}_i^j$ is the j^{th} component of reduced gradient vectors of the objective function $\widetilde{f_i}$. Once this set has been identified, we proceed to formulate the efficient cut (3.6) designed to eliminate all inefficient solutions. This cut plays an essential role in the optimization process, enabling us to reduce the search space, improve the overall efficiency of the optimization process, and ultimately, find optimal solutions.

$$\sum_{j \in \mathcal{H}_l} x_j \geq 1. \tag{3.6}$$

The second cut of type (3.7) is constructed according to the following inequality:

$$\phi(x) \leq \phi_{opt}, \tag{3.7}$$

where, the value ϕ_{opt} represents the optimal value of the main problem at the efficient point. The successor of l is obtained by applying the efficient cuts (3.6) and (3.7) to S_l.

$$S_{l+1} = S^1_{l+1} \cup S^2_{l+1},$$

where:

$$\begin{cases} S^1_{l+1} = \{x \in S_l \ / \ \sum_{j\in\mathcal{H}_l} x_j \geq 1\}, \\ S^2_{l+1} = \{x \in S_l \ / \phi(x) \leq \phi_{opt}\}. \end{cases}$$

Theorem 3.2.1. *Assume that $\mathcal{H}_l \neq \emptyset$ at the current integer solution x^l. If $x \neq x^l$ is an efficient solution in domain $\mathcal{S}_l$, then $x \in \mathcal{S}_{l+1}$ ($l+1$ is the successor of l).*

Proof. Let $x \neq x^l$ be an integer solution in domain $\mathcal{S}_l$ such that $x \notin \mathcal{S}_{l+1}$. In this situation, two cases can occur: $x \notin S^1_{l+1}$ or $x \notin S^2_{l+1}$.
$x \notin \mathcal{S}^1_{l+1}$, which implies that:
$x \notin \{x \in S_l / \sum_{j\in\mathcal{H}_l} x_j \geq 1\}$ and $x \in \left\{x \in \mathcal{S}_l | \ \sum_{j\in\mathcal{N}_l\setminus\mathcal{H}_l} x_j \geq 1\right\}$. In this situation, the components of vector x satisfy the following inequalities:

$$\begin{cases} \sum\limits_{j\in\mathcal{H}_l} x_j < 1, \\ \sum\limits_{j\in\mathcal{N}_l\setminus\mathcal{H}_l} x_j \geq 1. \end{cases}$$

This means that $x_j = 0$ for all $j \in \mathcal{H}_l$, and $x_j \geq 1$ for at least one index $j \in \mathcal{N}_l \setminus \mathcal{H}_l$. In addition, by using the simplex table in x^l, the following equality is supported for all criterion $i \in \{1, 2, \ldots, K\}$:

$$\widetilde{f_i}(x) = \widetilde{C_i}x = \sum_{j\in\mathcal{B}_l} \overline{C}^j_i x_j + \sum_{j\in\mathcal{N}_l} \overline{C}^j_i x_j, \ \text{ where } \ \sum_{j\in\mathcal{B}_l} \overline{C}^j_i x_j = \widetilde{f_i}(x^l).$$

We can then write:

$$\begin{aligned} \widetilde{f_i}(x) - \widetilde{f_i}(x^l) &= \sum_{j\in\mathcal{N}_l} \overline{C}^j_i x_j \\ &= \sum_{j\in\mathcal{H}_l} \overline{C}^j_i x_j + \sum_{j\in\mathcal{N}_l\setminus\mathcal{H}_l} \overline{C}^j_i x_j \\ &= \sum_{j\in\mathcal{N}_l\setminus\mathcal{H}_l} \overline{C}^j_i x_j. \end{aligned}$$

This implies that $\widetilde{f_i}(x) \leq \widetilde{f_i}(x^l)$ for all criterion $i \in \{1, 2, \ldots, K\}$, with $\widetilde{f_i}(x) < \widetilde{f_i}(x^l)$ for at least one criterion since $\overline{C}^j_i \leq 0$ for all $j \in \mathcal{N}_l \setminus \mathcal{H}_l$. Hence, $\widetilde{f_i}(x)$ is dominated by $\widetilde{f_i}(x^l)$ and x is not efficient.
$x \notin S^2_{l+1}$, $\phi(x) < \phi_{opt}$ Thus, x is not optimal, This contradicts the hypothesis. □

Corrolary 3.2.1. *The constraint $\sum_{j\in\mathcal{H}_l} x_j \geq 1$ defines an efficient cut.*

Proof. It is clear that $\sum_{j\in\mathcal{H}_l} x_j \geq 1$ is an efficient valid constraint by the theorem (3.2.2) since all integer efficient solutions in the current domain S_l verify this constraint. In addition, this constraint is not verified by the current integer solution x^l since $x_j = 0$ for all $j \in \mathcal{H}_l$. Thus, the constraint $\sum_{j\in\mathcal{H}_l} x_j \geq 1$ is a efficient cut. □

Proposition 3.2.1. *If $\mathcal{H}_l = \emptyset$ at the current integer solution x^l, then $S_l \backslash \{x^l\}$ is an explored domain.*

Proof. $\mathcal{H}_l = \emptyset$ which means that x^l is an optimal integer solution for each criterion, hence x^l is an ideal point to the domain S and $S_l \backslash \{x^l\}$ does not contain any efficient solution. □

3.3 Methodology description and algorithm

This section explains the methodology of the proposed method in detail, followed by a presentation of the algorithm that summarizes all the different steps of the method.

3.3.1 Methodology description

In the following, we introduce in detail our exact method called "A Branch and Cut based Method". As we explained previously, our BCM method is used to find the optimal efficient solution among several efficient solutions to a MOSILP problem. Moreover, the method is based on a branching process enhanced by feasibility, optimality, and efficiency tests, as well as by the use of efficient cutting.
Let's start with $\theta = -\infty$ and $l = 0$. The following relaxed problem is solved using the simplex or the dual simplex method:

$$(P^l)\begin{cases} min \;\; \widetilde{\phi} = \widetilde{d}x, \\ s.t., \\ \quad x \in S_l = \{x \in \mathbb{R}^n / Ax = b \,; x \in \mathbb{N}\}. \end{cases} \tag{3.8}$$

The algorithm now checks for the existence of a feasible solution. If the program has no solution, the node is fathomed. Otherwise, there are two possible situations:
The first is that the optimal solution x^l is not an integer. In this case, the algorithm follows a basic branching process with the following steps: Identify a component x_j of x^l such that $x_j = \alpha_j$, where α_j represents a fractional value. After that, the node l of the tree is separated into two nodes which are imposed by the following two additional constraints:

$$\begin{cases} x_j \;\leq \lfloor \alpha_j \rfloor, \\ x_j \;\geq \lfloor \alpha_j \rfloor + 1, \end{cases} \tag{3.9}$$

where, $\lfloor \alpha_j \rfloor$ indicates the greatest integer less than α_j. The linear program obtained must be solved until an integer feasible solution is found (if it exists).
The second situation is that the solution x^l is an integer. The process then passes two tests: the feasibility

test and the optimality test, which are detailed in subsection (2.2.4).
As soon as an optimal integer solution x^l is found after the feasibility and optimality tests, then the efficiency test is performed to test the efficiency of the solution x^l for the deterministic equivalent problem (3.3) by solving the following programs 3.10 (see, subsection 1.3.1):

$$(EK(x^l)) \begin{cases} \max \sum_{i=1}^{K} \psi_i, \\ s.t., \\ \widetilde{C}_i x + \psi_i = \widetilde{C}_i x^l, \ i = \{1, \ldots, K\}, \\ x \in \mathcal{S}_l \cap \mathbb{Z}^n, \\ w_i \geq 0, \ i = \{1, 2, \ldots, K\}. \end{cases} \tag{3.10}$$

As a well-known result, x^l is efficient if and only if $EK(x^l)$ has a maximum value of zero. See, [Ecker and Kouada, 1978].
The process continues to search for other efficient solutions in other nodes (if they exist) by applying efficient cut to the related program. First, the set $\mathcal{H}_l$ is created. Then, the efficient cut (3.11) is constructed and incorporated into subsequent nodes l.

$$\mathcal{H}_l = \left\{ j \in \mathcal{N}_l | \exists i \in \{1, 2, \ldots, K\}; \bar{C}_i^j \ < \ 0 \right\} \cup \left\{ j \in \mathcal{N}_l | \ \bar{C}_i^j = 0, \ \forall i = \{1, 2, \ldots, K\} \right\},$$

$$\sum_{j \in \mathcal{H}_l} x_j \geq 1. \tag{3.11}$$

If the solution x^l is not efficient for the deterministic equivalent problem (3.3). In addition, the given solution by the efficiency test is feasible and optimal. In this situation, we update the value of the objective function $\widetilde{\phi}_{opt}$. Another cut (3.12) is added to the related program.

$$\widetilde{\phi}(x) < \widetilde{\phi}_{opt}. \tag{3.12}$$

The method ends when all created nodes are fathomed. The optimal integer solution of the problem (3.4) is x_{opt} and corresponds to the value $\widetilde{\phi}_{opt}$.

3.3.2 Algorithm

The following algorithm (13) summarizes the main steps of the proposed method, these steps are treated according to the backtracking principle.

Algorithm 13: A Branch and Cut Strategies based Method

Data: Initialization $\theta = -\infty$, $l = 0$.

Result : x_{opt}: Optimal solution of the problem (3.4).
ϕ_{opt}: Optimal value of the objective function.

while *As long as a unfathomed node l exists in the tree* **do**
- solve the problem (3.4) using simplex or dual simplex method;
- **if** *the problem (3.4) has a feasible solution* x^l **then**
  - **if** x^l *is integer* **then**
    - **Feasibility test:** Solve the program (2.13);
    - **if** $\sigma_r^t[h(\xi^r) - T(\xi^r)x^l] > 0$ **then**
      - Add the feasibility cut (2.14) to the successors of l;
    - **else**
      - **Optimality test:** Solve the program (2.16), calculate Q(x);
      - **if** $Q(x^l) > \theta^l$ **then**
        - Add the optimality cut (2.19) to the successors of l;
      - **else**
        - **Efficiency test**
        - Solve the program (3.10), v is the optimal solution criteria;
        - **if** $v = 0$ **then**
          - $x_{opt} = x^l$, $\widetilde{\phi}_{opt} = \widetilde{\phi}(x^l)$
        - **else**
          - $\bar{x}^l$ is given by the Efficiency test (3.10).
          - **Feasibility and optimality test**
          - **if** $\bar{x}^l$ *Feasible and optimal* **then**
            - $x_{opt} = \bar{x}^l$, $\widetilde{\phi}_{opt} = \widetilde{\phi}(\bar{x}^l)$
            - Add the cut (3.12) $\widetilde{\phi}(x) < \widetilde{\phi}_{opt}$ to the successors of l
          - **end**
        - **end**
        - **Update set** $\mathcal{S}$. Construct the sets $\mathcal{H}_l$;
        - **if** $\mathcal{H}_l$ **then**
          - Fathom the node l;
        - **else**
          - Add the efficient cut (3.11) to the successors of l;
        - **end**
      - **end**
    - **end**
  - **else**
    - **Branching process**
    - Choose an index j such that α_j is the most fractional number. Split the program P^l into two sub programs, by adding respectively the constraints: $x_j \leq \lfloor \alpha_j \rfloor$ and $x_j \geq \lfloor \alpha_j \rfloor + 1$. To obtain (P^{l_1}) and (P^{l_2}) $(l_1 \neq l_2$ and $l_1 \geq l+1,\ l_2 \geq l+1)$;
  - **end**
- **else**
  - The corresponding node l is fathomed;
- **end**

end

Theorem 3.3.1. *The algorithm converges to the optimal solution of the problem 3.4, if it exists, in a finite number of iterations.*

Proof. Let $\mathcal{S}$ be the set of feasible integer solutions of the deterministic equivalent problem 3.3 a bounded set contained in S^l. In addition, the cardinality of the efficient set $\mathcal{X}_{ED}$ is a finite number too. During the process of the algorithm 13, when an integer solution is found, there are only a finite number of cuts that eliminate x^l and all dominated solutions from the search tree (see proposition 3.2.1).
Similarly, the algorithm used a limited number of feasibility and optimality cuts. which means that the search tree would have a finite number of branches and that the algorithm terminates in a finite number of steps. □

3.4 Didactic Example

Consider the following multi-objective integer linear programming stochastic problems presented as follows [Amrouche and Moulaï, 2012]:

Scenario 1

$$P(\xi^2)\begin{cases} \min f_1^1 = -9x_1 + x_2, \\ \min f_2^1 = 8x_1 - 5x_2, \\ \min f_3^1 = 2x_1 + 4x_2, \\ \text{s.t.,} \\ x_1 - 2x_2 \leq 5, \\ -x_1 + x_2 \leq 6, \\ x_1 + x_2 \leq 8, \\ x_1 + 2x_2 \leq 3, \\ -2x_1 + x_2 \leq 5, \\ x_1,\ x_2 \in \mathbb{N}. \end{cases}$$

Scenario 2

$$P(\xi^2)\begin{cases} \min f_1^3 = 3x_1 + x_2, \\ \min f_3^3 = 2x_1 + x_2, \\ \min f_3^2 = 6x_1 - 2x_2, \\ \text{s.t.,} \\ x_1 - 2x_2 \leq 5, \\ -x_1 + x_2 \leq 6, \\ x_1 + x_2 \leq 8, \\ x_1 \leq 6, \\ 3x_1 + 4x_2 \leq 1, \\ x_1,\ x_2 \in \mathbb{N}. \end{cases}$$

The recourse matrix and the deterministic constraints matrices are given for both scenarios by:

$$A = \begin{pmatrix} 1 & -2 \\ -1 & 1 \\ 1 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 5 \\ 6 \\ 8 \end{pmatrix}, W = \begin{pmatrix} -2 & -1 & 2 & 1 \\ 3 & 2 & -5 & -6 \end{pmatrix}.$$

The stochastic constraints matrices, the penalties of constraint violations, and the probability distribution are given for both scenarios by:

$$q(\xi^1) = (1, 0, 6, 2)^t, \quad \mathbb{P}(\xi^1) = \tfrac{1}{2}, \quad q(\xi^2) = (5, 3, 2, 1)^t, \quad \mathbb{P}(\xi^2) = \tfrac{1}{2},$$

$$T(\xi^1) = \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}, T(\xi^2) = \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix}, h(\xi^1) = \begin{pmatrix} 3 \\ 5 \end{pmatrix}, h(\xi^2) = \begin{pmatrix} 6 \\ 1 \end{pmatrix}.$$

The deterministic multi-objective integer linear programming problem:

$$\widetilde{f_1} = \mathbb{E}\left[C_1(\xi)\, x\right] = \mathbb{P}(\xi^1)\, C_1(\xi^1)\, x + \mathbb{P}(\xi^2)\, C_1(\xi^2)\, x,$$
$$\widetilde{f_2} = \mathbb{E}\left[C_2(\xi)\, x\right] = \mathbb{P}(\xi^1)\, C_2(\xi^1)\, x + \mathbb{P}(\xi^2)\, C_2(\xi^2)\, x,$$
$$\widetilde{f_3} = \mathbb{E}\left[C_3(\xi)\, x\right] = \mathbb{P}(\xi^1)\, C_3(\xi^1)\, x + \mathbb{P}(\xi^2)\, C_3(\xi^2)\, x,$$

$$(MOSILP_D)\begin{cases} \min\ \widetilde{f_1} = -3x_1 + x_2, \\ \min\ \widetilde{f_2} = 5x_1 - 2x_2, \\ \min\ \widetilde{f_3} = 2x_1 - x_2, \\ \text{s.t.,} \\ x_1 - 2x_2 \leq 5, \\ -x_1 + x_2 \leq 6, \\ x_1 + x_2 \leq 8. \end{cases}$$

Consider now the preference function of the decision-maker for each scenario:

Scenario 1 Scenario 2

$$\phi(\xi^1)\begin{cases} \min \phi^1 = -x_1 + 3x_2, \\ \text{s.t.,} \\ x_1,\ x_2 \in \mathcal{X}_{ED}. \end{cases} \qquad \phi(\xi^2)\begin{cases} \min \phi^2 = 5x_1 - 1x_2, \\ \text{s.t.,} \\ x_1,\ x_2 \in \mathcal{X}_{ED}. \end{cases}$$

The main deterministic relaxed problem is defined by:

$$\widetilde{\phi} = \mathbb{E}\left[d(\xi^r)\, x\right] = \mathbb{P}(\xi^1)\, d(\xi^1)\, x + \mathbb{P}(\xi^2)\, d(\xi^2)\, x,$$

$$(P_{DR})\begin{cases} \min\ \widetilde{\phi} = 2x_1 + x_2, \\ \text{s.t.,} \\ x \in S, \end{cases}$$

where, $S_0 = \{x \in \mathbb{R}^n |\ x_1 - 2x_2 \leq 5, -x_1 + x_2 \leq 6, x_1 + x_2 \leq 8, x_1, x_2 \in \mathbb{N}\}$.

Initial iteration: $\phi_{opt} = +\infty,\ \theta = -\infty$.

Step 1: General step
The relaxed problem (P^0) is solved.

$$(P^0)\begin{cases} \min\ \widetilde{\phi} = 2x_1 + x_2, \\ \text{s.t.,}\ x \in S_0. \end{cases}$$

The first optimal solution is $x^0 = (0, 0)$. In this situation, x^0 will be tested for feasibility and optimality for each scenario.

Step 3: Feasibility and Optimality Tests

At first, we evaluate the feasibility of the solution x^0 by solving the following two problems:

$$h(\xi^1) - T(\xi^1)\,x^0 = \begin{pmatrix} 3 \\ 5 \end{pmatrix} - \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}.$$
$$h(\xi^2) - T(\xi^2)\,x^0 = \begin{pmatrix} 6 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix}\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 6 \\ 1 \end{pmatrix}.$$

Scenario 1 Scenario 2

$$\sigma(\xi^1)\begin{cases} \max\ 3\sigma_1^1 + 5\sigma_1^2, \\ -2\sigma_1^1 + 3\sigma_1^2 \le 0, \\ -\sigma_1^1 + 2\sigma_1^2 \le 0, \\ 2\sigma_1^1 - 5\sigma_1^2 \le 0, \\ \sigma_1^1 - 6\sigma_1^2 \le 0, \\ \sigma_1^1 + \sigma_1^2 \le 1. \end{cases} \quad \sigma_1^t = \left(\tfrac{2}{3}, \tfrac{1}{3}\right), \qquad \sigma(\xi^2)\begin{cases} \max\ 6\sigma_2^1 + 1\sigma_2^2, \\ -2\sigma_2^1 + 3\sigma_2^2 \le 0, \\ -\sigma_2^1 + 2\sigma_2^2 \le 0, \\ 2\sigma_2^1 - 5\sigma_2^2 \le 0, \\ \sigma_2^1 - 6\sigma_2^2 \le 0, \\ \sigma_2^1 + \sigma_2^2 \le 1. \end{cases} \quad \sigma_2^t = \left(\tfrac{5}{7}, \tfrac{2}{7}\right).$$

Note that, $\sigma_1^t\left[h(\xi^1) - T(\xi^1)\,x^0\right] = \left(\frac{2}{3}, \frac{1}{3}\right)\binom{3}{5} = \frac{11}{3} > 0$, this means that x^0 is not feasible for the ξ^1. In this situation, we introduce the following feasibility cut and addit to the current problem.

$$\left(\tfrac{2}{3}, \tfrac{1}{3}\right)\begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} > \left(\tfrac{2}{3}, \tfrac{1}{3}\right)\begin{pmatrix} 3 \\ 5 \end{pmatrix} = \frac{11}{3} \iff 5x_2 \ge 11.$$

After adding the feasibility cut, we solve the following new problem:

$$(P^1)\begin{cases} \min\ \widetilde{\phi} = 2x_1 + x_2, \\ \text{s.t.},\ x \in S_1 = \{x \in S_0 |\ 5x_2 \ge 11, x \in \mathbb{N}^n\}. \end{cases}$$

Minimum is reached at $x^1 = (0, \frac{11}{5})$ but it is not an integer. The algorithm passes throughthe branching process.

Step 2: The branching process

Node 2: $S_2 = \left\{x \in S_1\ :\ x_2 \ge \left\lfloor \frac{11}{5} \right\rfloor + 1\right\}$ et $x_2 \ge 3$.

Node 3: $S_3 = \left\{ x \in S_1\ :\ x_2 \le \left\lfloor \frac{11}{5} \right\rfloor \right\}$ et $x_2 \le 2$. No solution exists in this node. then, the node is **fathomed**.

We resolve the new program in node 1.

$$(P^2)\begin{cases} \min\ \widetilde{\phi} = 2x_1 + x_2, \\ \text{s.t.},\ x \in S_2. \end{cases}$$

The integer solution obtained is $x^2 = (0, 3)$. We continue the process by testing the feasibility and optimality of the solution x^2.

Step 3: Feasibility and Optimality Tests

As previously mentioned, this step commences with a feasibility test, where the feasibility of the solution x^2 is examined by solving the following pair of problems.

Scenario 1 Scenario 2

$$\sigma(\xi^1)\begin{cases} \max\ -3\sigma_1^1 + 2\sigma_1^2, \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0, \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0, \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0, \\ \sigma_1^1 - 6\sigma_1^2 \leq 0, \\ \sigma_1^1 + \sigma_1^2 \leq 1. \end{cases} \quad \sigma_1^t = (0, 0)\,, \qquad \sigma(\xi^2)\begin{cases} \max\ 6\sigma_2^1 - 11\sigma_2^2, \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0, \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0, \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0, \\ \sigma_2^1 - 6\sigma_2^2 \leq 0, \\ \sigma_2^1 + \sigma_2^2 \leq 1. \end{cases} \quad \sigma_2^t = \left(\tfrac{5}{7}, \tfrac{2}{7}\right).$$

We have here $\sigma_2^t \left[h\left(\xi^2\right) - T\left(\xi^2\right) x^2\right] = \left(\frac{5}{7}, \frac{2}{7}\right)\binom{6}{-11} = \frac{8}{7} > 0$. x^2 is not feasible for the second scenario, we construct then the feasible constraint for the second scenario.

$$\begin{pmatrix} \frac{5}{7}, \frac{2}{7} \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} > \begin{pmatrix} \frac{5}{7}, \frac{2}{7} \end{pmatrix}\begin{pmatrix} 6 \\ 1 \end{pmatrix} = \frac{32}{7} \iff 11x_1 + 8x_2 \geq 32.$$

We resolve the new program:

$$(P^3)\begin{cases} \min\ \widetilde{\phi} = 2x_1 + x_2, \\ \text{s.t.},\ x \in S_4 = \{x \in S_2, \mid 11x_1 + 8x_2 \geq 32,\ x \in \mathbb{N}^n.\} \end{cases}$$

The minimum is reached at $x^3 = (0, 4)$, we pass again through the feasibility and optimality test step.

Step 3: Feasibility and Optimality Tests

We test the feasibility of x^3 by solving the following two problems:

Scenario 1 Scenario 2

$$\sigma(\xi^1)\begin{cases} \max\ -5\sigma_1^1 + \sigma_1^2, \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0, \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0, \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0, \\ \sigma_1^1 - 6\sigma_1^2 \leq 0, \\ \sigma_1^1 + \sigma_1^2 \leq 1. \end{cases} \quad \sigma_1^t = (0, 0)\,, \qquad \sigma(\xi^2)\begin{cases} \max\ 6\sigma_2^1 - 15\sigma_2^2, \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0, \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0, \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0, \\ \sigma_2^1 - 6\sigma_2^2 \leq 0, \\ \sigma_2^1 + \sigma_2^2 \leq 1. \end{cases} \quad \sigma_2^t = \left(\tfrac{5}{7}, \tfrac{2}{7}\right).$$

Here, $\sigma_1^t = \sigma_2^t = 0$, which means that, x^3 is feasible for both scenarios. Following this, we perform the

optimality test for x^3 by solving the following two problems:

Scenario 1 Scenario 2

$$\pi(\xi^1)\begin{cases} \max\ -5\pi_1^1+\pi_1^2, \\ -2\pi_1^1+3\pi_1^2\le 1, \\ -\pi_1^1+2\pi_1^2\le 0, \quad \pi_1^t=\left(-1,-\frac{1}{2}\right), \\ 2\pi_1^1-5\pi_1^2\le 6, \\ \pi_1^1-6\pi_1^2\le 2. \end{cases} \qquad \pi(\xi^2)\begin{cases} \max\ 6\pi_1^1-15\pi_1^2, \\ -2\pi_1^1+3\pi_1^2\le 5, \\ -\pi_1^1+2\pi_1^2\le 3, \quad \pi_2^t=(1,0). \\ 2\pi_1^1-5\pi_1^2\le 2, \\ \pi_1^1-6\pi_1^2\le 1. \end{cases}$$

At first, we calculate the expected recourse function value $Q(x)$ by:

$$Q\left(x,\xi^1\right)=\pi_1^t\left[h\left(\xi^1\right)-T\left(\xi^1\right)x^3\right]=\frac{9}{2},$$
$$Q\left(x,\xi^2\right)=\pi_2^t\left[h\left(\xi^2\right)-T\left(\xi^2\right)x^3\right]=6,$$
$$Q\left(x\right)=\frac{1}{2}Q\left(x,\xi^1\right)+\frac{1}{2}Q\left(x,\xi^2\right)=\frac{21}{4}.$$

We notice that: $Q\left(x\right) > theta = -\infty$. Consequently, we construct the optimality cut as outlined below. Then we add it to the current problem.

$$\theta \ge \frac{1}{4}-\frac{1}{2}x_1+\frac{5}{4}x_2.$$

After adding the optimality constraint, we resolve the new problem.

$$(P^4)\begin{cases} \min\ \widetilde{\phi}=2x_1+x_2, \\ \text{s.t.},\ x\in S_4^{'}=\{x\in S_4|\ \theta\ge\frac{1}{4}-\frac{1}{2}x_1+\frac{5}{4}x_2,\ x\in\mathbb{N}^n\}. \end{cases}$$

We obtain the same optimal solution $x^3=(0,4)$. Moreover, in this situation, this obtained solution is feasible and optimal for both scenarios with penalty value $\theta=\frac{21}{4}$, see Table 3.1.
After this, the algorithm proceeds to another step, which is the efficiency test. This test is conducted to evaluate the efficiency of the attained solution the deterministic multiobjective problems.

Table 3.1: Optimal simplex table after introducing optimality cut

B	Rhs	x_8	x_1
x_3	13	$\frac{15}{4}$	$-\frac{7}{4}$
x_4	2	$-\frac{19}{8}$	$\frac{7}{8}$
x_5	4	$-\frac{3}{8}$	$\frac{7}{8}$
x_2	4	$\frac{11}{8}$	$-\frac{7}{8}$
x_6	3	$-\frac{55}{24}$	$\frac{35}{24}$
x_7	1	$\frac{11}{8}$	$-\frac{7}{8}$
θ	$\frac{21}{4}$	$\frac{71}{32}$	$-\frac{35}{32}$
$\widetilde{f_1}$	4	$-\frac{35}{8}$	$\frac{7}{8}$
$\widetilde{f_2}$	-8	$\frac{31}{4}$	$-\frac{7}{4}$
$\widetilde{f_3}$	-4	$\frac{27}{8}$	$-\frac{7}{8}$

Step 4: Efficiency test

We test the efficiency of x^3 for the deterministic multi-objective problem by solving the following problem:

$$(EK(x^3))\begin{cases} \max \Psi = \psi_1 + \psi_2 + \psi_3, \\ \text{s.t.,} \quad x \in S_4, \\ \theta \geq \frac{1}{4} - \frac{1}{2}x_1 + \frac{5}{4}x_2, \\ -3x_1 + x_2 + \psi_1 = 4, \\ 5x_1 - 2x_2 + \psi_2 = -8, \\ 2x_1 - x_2 + \psi_3 = -4, \\ x_1, x_2 \in \mathbb{N},\ \theta, \psi_1, \psi_2, \psi_3 \in \mathbb{R}. \end{cases}$$

Note that, $\max \Psi = 2 \neq 0$, this means that x^3 is not efficient. The solution $\overline{x}^3 = (1, 7)$ given by the efficiency test is efficient. In this situation, we test the feasibility and the optimality of the solution $\overline{x}^3$. As a result, $\overline{x}^3$ is feasible and optimal for the second stage-problems. With $\theta = \frac{17}{2}$. We notice that $\phi_{\overline{x}^3} = 9 < \phi_{opt} = +\infty$. Now we modify our best solution and the objective function: $x_{opt} = (1, 7)$, $\phi_{opt} = 9$ and we continue the algorithm process through the last step

Step 5: The efficient cut

In this step, we construct the set $\mathcal{H}_1$ by using the decreasing direction of each criterion $\widetilde{f}_{i\in\{1,\ldots,3\}}$ for the deterministic multi-objective problem. From the table 3.1, $\mathcal{H}_1 = \{1, 8\} \neq \emptyset$.

After adding the cuts $x_1 + x_8 \geq 1$ and the second cut $\phi(x) \leq \phi_{opt} \Rightarrow 2x_1 + x_2 \leq 9$ to the current problem. The algorithm continues the search process for other efficient solutions by resolving the following new problem.

$$(P^5)\begin{cases} \min\ \widetilde{\phi} = 2x_1 + x_2, \\ \text{s.t.,}\ x \in S_5 = \{x \in S_4^{'} \mid x_1 + x_8 \geq 1,\ 2x_1 + x_2 \leq 9\}. \end{cases}$$

The solution obtained $x^4 = (\frac{5}{6}, 3)$ is not an integer.

Step 2: The branching process starts

Node 6: $S_6 = \left\{x \in S_5 \ :\ x_1 \geq \left\lfloor \frac{5}{6} \right\rfloor + 1\right\}$ et $x_1 \geq 1$.

Node 7: $S_7 = \left\{x \in S_5 \ :\ x_1 \leq \left\lfloor \frac{5}{6} \right\rfloor\right\}$ et $x_1 \leq 0$.

The new problem is Solved at node 6.

$$(P^6)\begin{cases} \min\ \widetilde{\phi} = 2x_1 + x_2, \\ \text{s.t.,}\ x \in S_6. \end{cases}$$

The obtained integer solution is $x^5 = (1, 3)$, see Table 3.2. We continue the process by testing the feasibilityand optimality of the solution x^5.

Table 3.2: Optimal simplex table after branching process at node 3

B	Rhs	x_{12}	x_7
x_3	10	1	-2
x_4	4	-1	1
x_5	4	1	1
θ	$\frac{7}{2}$	$\frac{1}{2}$	$-\frac{5}{4}$
x_8	$\frac{3}{7}$	$-\frac{11}{7}$	$-\frac{8}{7}$
x_6	$\frac{4}{3}$	0	$-\frac{5}{3}$
x_1	1	-1	0
x_2	3	0	-1
x_{11}	3	2	1
x_{10}	$\frac{3}{7}$	$-\frac{18}{7}$	$-\frac{8}{7}$
$\widetilde{f_1}$	0	1	-3
$\widetilde{f_2}$	-1	-2	5
$\widetilde{f_3}$	-1	-1	2

Step 3: Feasibility and Optimality Tests

The feasibility of the solution x^5 is examined by solving the following pair of problems.

Scenario 1

$$\sigma(\xi^1)\begin{cases} \max\ -4\sigma_1^1 + 4\sigma_1^2, \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0, \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0, \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0, \\ \sigma_1^1 - 6\sigma_1^2 \leq 0, \\ \sigma_1^1 + \sigma_1^2 \leq 1. \end{cases} \quad \sigma_1^t = (0,0)\,,$$

Scenario 2

$$\sigma(\xi^2)\begin{cases} \max\ 5\sigma_2^1 - 14\sigma_2^2, \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0, \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0, \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0, \\ \sigma_2^1 - 6\sigma_2^2 \leq 0, \\ \sigma_2^1 + \sigma_2^2 \leq 1. \end{cases} \quad \sigma_2^t = (0,0)\,.$$

Note that, $\sigma_1^t = \sigma_2^t = 0$, which means that x^5 is feasible for both scenarios. In this case, we perform the optimality test.

Scenario 1

$$\pi(\xi^1)\begin{cases} \max\ -4\pi_1^1 + 4\pi_1^2, \\ -2\pi_1^1 + 3\pi_1^2 \leq 1, \\ -\pi_1^1 + 2\pi_1^2 \leq 0, \\ 2\pi_1^1 - 5\pi_1^2 \leq 6, \\ \pi_1^1 - 6\pi_1^2 \leq 2. \end{cases} \quad \pi_1^t = \left(-1, -\tfrac{1}{2}\right),$$

Scenario 2

$$\pi(\xi^2)\begin{cases} \max\ 5\pi_1^1 - 14\pi_1^2, \\ -2\pi_1^1 + 3\pi_1^2 \leq 5, \\ -\pi_1^1 + 2\pi_1^2 \leq 3, \\ 2\pi_1^1 - 5\pi_1^2 \leq 2, \\ \pi_1^1 - 6\pi_1^2 \leq 1. \end{cases} \quad \pi_2^t = (1,0)\,.$$

In this situation, $Q(x) = \frac{7}{2}$, and we note that $Q(x) = \theta$ (see table 3.2), this means that $x^5 = (1,3)$ is optimal for both scénarios with $\theta = \frac{7}{2}$. Following this, we proceed to the next step of the algorithm.

Step 4: Efficiency test

We test the efficiency of x^5 for the deterministic multi-objective problem by solving the following problem:

$$(EK(x^5))\begin{cases} \max \Psi = \psi_1 + \psi_2 + \psi_3, \\ \text{s.t.,} \quad x \in S_4, \\ \theta \geq \frac{1}{4} - \frac{1}{2}x_1 + \frac{5}{4}x_2, \\ -3x_1 + x_2 + \psi_1 = 0, \\ 5x_1 - 2x_2 + \psi_2 = -1, \\ 2x_1 - x_2 + \psi_3 = -1, \\ x_1, x_2 \in \mathbb{N},\ \theta, \psi_1, \psi_2, \psi_3 \in \mathbb{R}. \end{cases}$$

Here, $\Psi = 2 \neq 0$, this means x^5 is not efficient. In addition, the solution $\overline{x}^5 = (2, 6)$ given by the efficiency test is not optimal for the second-stage problems. In this situation, we continue the search process for the optimal solution by adding an efficient cut.

Step 5: The efficient cut

In this step, we construct two new set $\mathcal{H}_2 = \{13, 7\} \neq \emptyset$. See Table 3.2. After that, we add the new efficient cut $x_7 + x_{13} \geq 1$ to the current problem.

Once again, the algorithm continues the search process of finding other efficient solutions by solving the new problem:

$$(P^7)\begin{cases} \min\ \widetilde{\phi} = 2x_1 + x_2, \\ \text{s.t.,}\ x \in S_8 = \{x \in S_6 \mid x_{13} + x_7 \geq 1\}. \end{cases}$$

The integer solution obtained is $x^6 = (1, 4)$.

Step 3: Feasibility and Optimality Tests

We test the feasibility of x^6 by solving the following problems:

Scenario 1

$$\sigma(\xi^1)\begin{cases} \max\ -6\sigma_1^1 + 3\sigma_1^2, \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0, \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0, \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0, \\ \sigma_1^1 - 6\sigma_1^2 \leq 0, \\ \sigma_1^1 + \sigma_1^2 \leq 1. \end{cases} \quad \sigma_1^t = (0, 0)\,,$$

Scenario 2

$$\sigma(\xi^2)\begin{cases} \max\ 5\sigma_2^1 - 18\sigma_2^2, \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0, \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0, \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0, \\ \sigma_2^1 - 6\sigma_2^2 \leq 0, \\ \sigma_2^1 + \sigma_2^2 \leq 1. \end{cases} \quad \sigma_2^t = (0, 0)\,.$$

In this case, $\sigma_1^t = \sigma_2^t = 0$, this mean x^6 is feasible for both scenarios. Now, we test the optimality of x^6 is by solving the following problems:

Scenario 1

$$\pi(\xi^1)\begin{cases} \max\ -6\pi_1^1 + 3\pi_1^2, \\ -2\pi_1^1 + 3\pi_1^2 \leq 1, \\ -\pi_1^1 + 2\pi_1^2 \leq 0, \\ 2\pi_1^1 - 5\pi_1^2 \leq 6, \\ \pi_1^1 - 6\pi_1^2 \leq 2. \end{cases} \quad \pi_1^t = \left(-1, -\tfrac{1}{2}\right),$$

Scenario 2

$$\pi(\xi^2)\begin{cases} \max\ 5\pi_1^1 - 18\pi_1^2, \\ -2\pi_1^1 + 3\pi_1^2 \leq 5, \\ -\pi_1^1 + 2\pi_1^2 \leq 3, \\ 2\pi_1^1 - 5\pi_1^2 \leq 2, \\ \pi_1^1 - 6\pi_1^2 \leq 1. \end{cases} \quad \pi_2^t = (1, 0)\,.$$

After calculating $Q(x) = \frac{64}{9}$, we found that $Q(x) = \theta$ which means that x^6 is the integer optimal solution with $\theta = \frac{19}{4}$.

Step 4: Efficiency test

We test the efficiency of x^6 by solving the following problem:

$$(EK(x^6)) \begin{cases} \max \Psi = \psi_1 + \psi_2 + \psi_3, \\ \text{s.t.,} \quad x \in S_4, \\ \theta \geq \frac{1}{4} - \frac{1}{2}x_1 + \frac{5}{4}x_2, \\ -3x_1 + x_2 + \psi_1 = 1, \\ 5x_1 - 2x_2 + \psi_2 = -3, \\ 2x_1 - x_2 + \psi_3 = -2, \\ x_1, x_2 \in \mathbb{N},\ \theta, \psi_1, \psi_2, \psi_3 \in \mathbb{R}. \end{cases}$$

Note that the max $\Psi = 0$, that means x^6 is efficient with $\theta = \frac{19}{4}$. In addition, we notice that $\phi_{x^6} = 6 < \phi_{opt} \Longrightarrow x_{opt} = (1, 4),\ \phi_{opt} = 6$. **Stop, the node** 8 **is fathomed.**

At this step, the algorithm tries to find another solution better than the previous one if it exists, by visiting other nodes. Here, we resolve the program at node 4.

$$(P^8) \begin{cases} \min\ \widetilde{\phi} = 2x_1 + x_2, \\ \text{s.t.,}\ x \in S_7. \end{cases}$$

The minimum is reached at $x^7 = (0, \frac{39}{8})$ it's not an integer solution, In this situation, we start the branching process.

Step 2: The branching process starts

Node 9: $S_9 = \left\{ x \in S_7 \ :\ x_2 \geq \left\lfloor \frac{11}{5} \right\rfloor + 1 \right\}$ et $x_2 \geq 5$.

Node 10: $S_{10} = \left\{ x \in S_7 \ :\ x_2 \leq \left\lfloor \frac{11}{5} \right\rfloor \right\}$ et $x_2 \leq 4$. No solution exists in this node. then, the node is **fathomed**.

The new problem is solved at node 5, and the new optimal solution is $x^8 = (0, 5)$.

$$(P^9) \begin{cases} \min\ \widetilde{\phi} = 2x_1 + x_2, \\ \text{s.t.,}\ x \in S_{11},\ \ x \in \mathbb{N}^n. \end{cases}$$

Step 3: Feasibility and Optimality Tests

We resolve the next following problems to test the feasibility of x^8.

Scenario 1 Scenario 2

$$\sigma(\xi^1)\begin{cases}\max\ -7\sigma_1^1,\\ -2\sigma_1^1+3\sigma_1^2\le 0,\\ -\sigma_1^1+2\sigma_1^2\le 0,\\ 2\sigma_1^1-5\sigma_1^2\le 0,\\ \sigma_1^1-6\sigma_1^2\le 0,\\ \sigma_1^1+\sigma_1^2\le 1.\end{cases}\quad \sigma_1^t=(0,0)\,,\qquad \sigma(\xi^2)\begin{cases}\max\ 6\sigma_2^1-19\sigma_2^2,\\ -2\sigma_2^1+3\sigma_2^2\le 0,\\ -\sigma_2^1+2\sigma_2^2\le 0,\\ 2\sigma_2^1-5\sigma_2^2\le 0,\\ \sigma_2^1-6\sigma_2^2\le 0,\\ \sigma_2^1+\sigma_2^2\le 1.\end{cases}\quad \sigma_2^t=(0,0)\,.$$

Here, $\sigma_1^t=\sigma_2^t=0$, this mean x^8 is feasible for both scenarios.In this situation, we check the optimality of x^8 for the second-stage problems.

Scenario 1 Scenario 2

$$\pi(\xi^1)\begin{cases}\max\ -7\pi_1^1,\\ -2\pi_1^1+3\pi_1^2\le 1,\\ -\pi_1^1+2\pi_1^2\le 0,\\ 2\pi_1^1-5\pi_1^2\le 6,\\ \pi_1^1-6\pi_1^2\le 2.\end{cases}\quad \pi_1^t=\left(-1,-\tfrac{1}{2}\right),\qquad \pi(\xi^2)\begin{cases}\max\ 6\pi_1^1-19\pi_1^2,\\ -2\pi_1^1+3\pi_1^2\le 5,\\ -\pi_1^1+2\pi_1^2\le 3,\\ 2\pi_1^1-5\pi_1^2\le 2,\\ \pi_1^1-6\pi_1^2\le 1.\end{cases}\quad \pi_2^t=(1,0)\,.$$

In this situation, $Q(x)=\frac{13}{2}=\theta$ and this mean that x^8 is the integer optimal solution with $\theta=\frac{13}{2}$.

Step 4: Efficiency test

We test the efficiency of x^8 by solving the following problem:

$$(EK(x^8))\begin{cases}\max\Psi=\psi_1+\psi_2+\psi_3,\\ \text{s.t.,}\quad x\in S_4,\\ \theta\ge\frac{1}{4}-\frac{1}{2}x_1+\frac{5}{4}x_2,\\ -3x_1+x_2+\psi_1=5,\\ 5x_1-2x_2+\psi_2=-10,\\ 2x_1-x_2+\psi_3=-5,\\ x_1,x_2\in\mathbb{N},\ \theta,\psi_1,\psi_2,\psi_3\in\mathbb{R}.\end{cases}$$

Note that, $\Psi=0$, which means that x^8 is efficient with $\theta=\frac{13}{2}$. In this situation, $\phi_{x^4}=5<\phi_{opt}\Longrightarrow x_{opt}=(0,5),\ \phi_{opt}=5$, **Stop, this node 9 is fathomed.**

The algorithm terminates with $x_{opt}=(0,5)$ and $\phi_{opt}=5$.

To summarize the proposed method throughout this example, we present a tree Figure 4.1 showing the states of the nodes during the process:

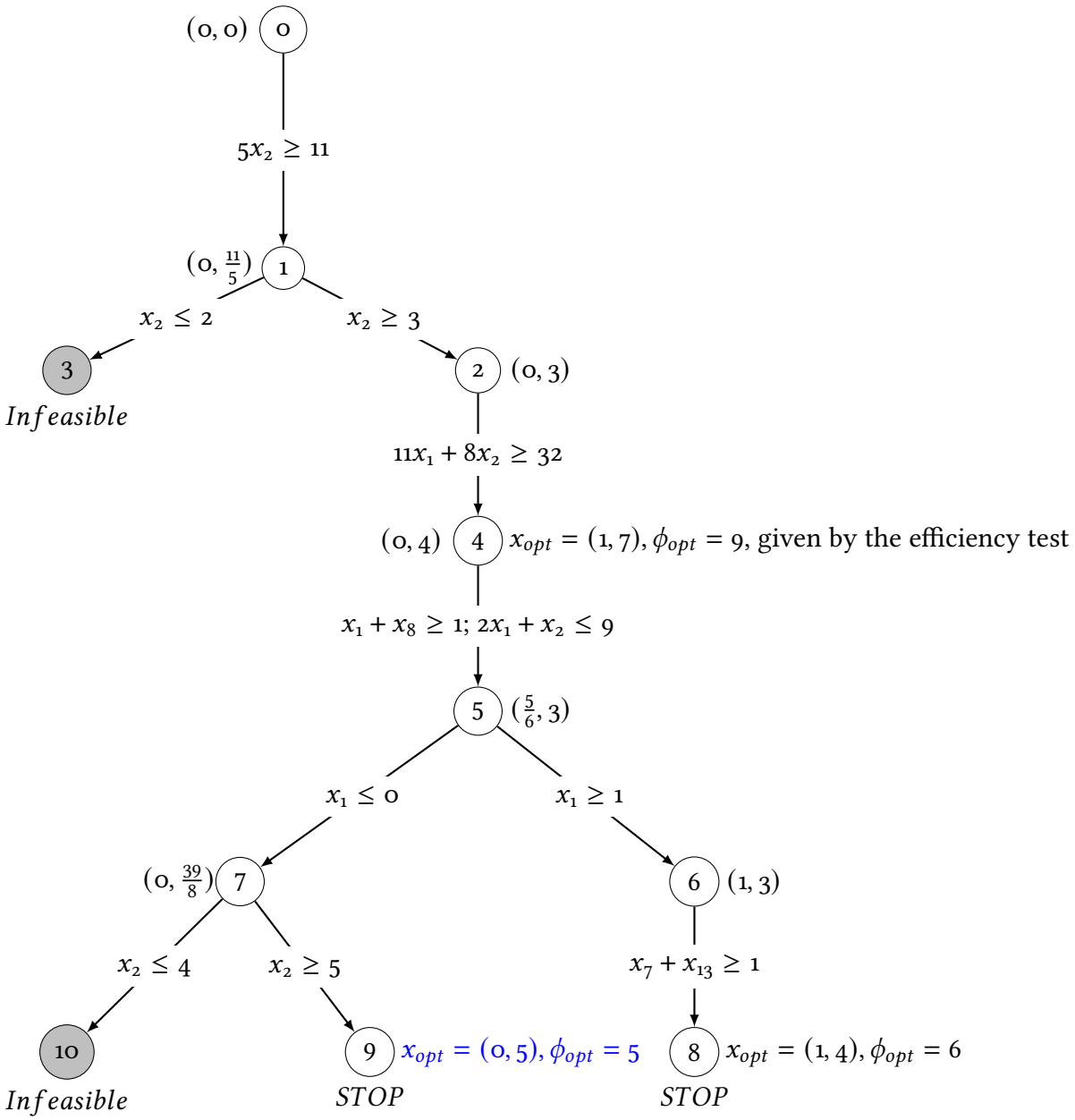


Figure 3.1: Search tree of the example

3.5 Computational Results

According to the available literature, no benchmark instances have been considered for testing the performance of the only method [Chaabane and Mebrek, 2014] (CMM) presented for solving this problem. Nonetheless, no experimental study has been conducted by the authors in their paper. For this reason, we thought about creating randomly generated instances for testing the behavior of our method (BCM) for solving the case of this problem and to make an experimental comparative study with their method. We have coded and implemented both algorithms in a MATLAB environment without making use of any solver and tested over the same randomly generated instances with the same software environment. For the efficiency and optimality tests, we have used the linprog solver. The deterministic data (constraint coefficients) are uncorrelated and uniformly distributed. Each component of the vector b, the entries of the matrix A, and the coefficients of the objective functions C and d were randomly drawn from discrete uniform distributions in the following ranges $[100, 200]$, $[1, 100]$, $[-100, 100]$ and $[-100, 100]$ respectively.

The stochastic data are generated in the same way as the deterministic ones. For each component of the recourse matrix W, the penalties for constraint violations q, the two matrices T and h are in the intervals $[-50, 50]$, $[1, 50]$, $[-50, 50]$ and $[-50, 50]$ respectively. The probability for each scenario is randomly generated with the sum of probabilities equal to 1.

We have worked on 44 different groups of instances of (n, m, k, R) type, where n is the number of variables, m is the number of constraints, k is the number of objectives and R is the number of scenarios, $R = \{2, 3, 5, 10\}$. However, five randomly generated instances were created for each group, which will result in an outcome of 220 global instances, these instances are solved by BCM and CMM methods, where their results are compared in Table 3.3.

Note that: The generation of random instances does not guarantee the feasibility of the instance. For this study, we have kept only the feasible instances. Also, all proposed solution procedures of the computational results section are performed on a computer with Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz processor and 4 GB RAM.

Results discussion: Table 3.3 shows the results achieved by both methods BCM and CMM on five instances for each group. Column 1 represents the number of existing scenarios R, column 2 presents the size of each instance $(n \times m \times k)$, where n is the number of variables, m is the number of constraints and k is the number of objectives, columns 3 and 4 of the table display the average CPU time (in seconds) for both methods BCM and CMM respectively, columns 5 and 6 report the average number of nodes required (Nbr of Nodes) for both methods, columns 7 and 8 represent the average number of visited solutions (Nbr of Sols) before finding the optimal one, columns 9 and 10 indicate the average number of efficient cuts (Nbr of Cuts) used by BCM, and the [Sylva and Crema, 2004] used by CMM.

Table 3.3: The behavior of the BCM method compared to the CCM method on medium and large scale instances over different scenarios and based upon different criteria.

		CPU (Sec)		*Nbr of Nodes*		*Nbr of Sols*		*Nbr of Cuts*	
R	$n \times m \times k$	*BCM*	*CMM*	*BCM*	*CMM*	*BCM*	*CMM*	*BCM*	*CMM*
2	$10 \times 5 \times 3$	1.23	2.42	48.40	94.40	0.20	0.20	1.80	3.40
	$20 \times 10 \times 5$	2.60	7.41	82.00	193.20	1.00	1.00	1.60	4.00
	$30 \times 15 \times 5$	7.29	17.55	144.80	230.40	1.00	1.20	1.40	4.80
	$40 \times 20 \times 5$	22.47	50.17	339.60	592.00	1.40	1.40	2.60	5.00
	$50 \times 25 \times 10$	37.40	42.68	396.00	410.00	0.80	0.80	0.00	1.80
	$60 \times 30 \times 10$	126.93	136.61	769.20	799.60	2.40	2.40	0.20	3.60
	$70 \times 35 \times 10$	130.71	168.55	681.20	760.00	1.20	1.20	0.40	2.60
	$80 \times 40 \times 20$	227.21	239.62	892.00	905.20	1.40	1.40	0.20	2.60
	$90 \times 45 \times 20$	347.77	356.07	1026.80	1034.40	1.00	1.00	0.00	2.00
	$100 \times 50 \times 20$	429.25	457.62	1094.40	1118.40	2.00	2.00	0.40	3.40
	$110 \times 55 \times 30$	518.67	529.81	994.00	1001.60	1.60	1.60	0.20	2.80
	Average ($R = 2$)	168.32	182.59	588.04	649.02	1.27	1.29	0.80	3.27
3	$10 \times 5 \times 3$	2.60	7.75	30.40	126.40	0.80	1.00	3.20	7.00
	$20 \times 10 \times 5$	3.00	6.59	82.00	147.60	0.80	0.80	0.80	2.80
	$30 \times 15 \times 5$	11.25	14.97	252.00	292.00	0.00	0.00	0.80	1.80
	$40 \times 20 \times 5$	26.45	97.26	392.20	788.00	0.60	0.60	3.80	7.00
	$50 \times 25 \times 10$	37.19	37.39	351.20	361.60	1.20	1.20	0.00	2.20
	$60 \times 30 \times 10$	55.08	91.54	376.80	528.40	1.40	1.40	0.60	3.00
	$70 \times 35 \times 10$	138.96	144.93	632.60	654.80	2.20	2.20	0.40	3.60
	$80 \times 40 \times 20$	218.11	225.08	852.80	871.20	0.80	0.80	0.40	2.20
	$90 \times 45 \times 20$	222.42	236.52	968.00	998.00	1.20	1.20	0.20	2.40
	$100 \times 50 \times 20$	355.92	379.03	930.80	962.00	2.40	2.40	0.00	3.40
	$110 \times 55 \times 30$	850.92	853.24	1605.20	1619.20	2.00	2.00	0.00	3.00
	Average ($R = 3$)	174.72	190.30	563.27	647.38	1.22	1.24	0.93	3.49
5	$10 \times 5 \times 3$	3.84	12.63	55.60	267.60	0.80	1.00	4.60	11.60
	$20 \times 10 \times 5$	5.26	15.22	106.00	311.60	1.80	2.20	3.40	6.40
	$30 \times 15 \times 5$	7.70	21.48	153.60	322.00	1.40	1.40	2.00	5.00
	$40 \times 20 \times 5$	17.77	44.88	233.60	445.60	1.20	1.20	2.20	4.40
	$50 \times 25 \times 10$	16.20	18.39	188.00	197.20	1.00	1.00	0.00	2.00
	$60 \times 30 \times 10$	51.60	60.27	303.60	330.00	1.40	1.40	0.20	2.60
	$70 \times 35 \times 10$	110.80	151.64	602.40	684.40	1.40	1.40	0.20	2.60
	$80 \times 40 \times 20$	150.99	161.82	646.40	665.20	1.00	1.00	0.00	4.58
	$90 \times 45 \times 20$	229.11	247.07	767.60	787.60	1.60	1.60	0.00	2.60
	$100 \times 50 \times 20$	345.16	355.52	902.40	924.00	1.60	1.60	0.40	3.00
	$110 \times 55 \times 30$	452.84	476.31	944.80	973.60	2.40	2.40	0.00	3.40
	Average ($R = 5$)	126.48	142.30	445.82	537.16	1.42	1.47	1.18	4.15
10	$10 \times 5 \times 3$	19.64	27.84	124.00	313.60	1.00	1.00	15.00	20.60
	$20 \times 10 \times 5$	7.05	20.94	106.40	326.80	1.00	1.00	3.00	7.80
	$30 \times 15 \times 5$	9.02	16.40	156.80	244.80	0.60	0.60	1.60	3.20
	$40 \times 20 \times 5$	12.19	35.90	152.80	317.20	1.20	1.40	1.20	4.40
	$50 \times 25 \times 10$	22.96	29.52	228.00	255.20	2.00	2.00	0.20	3.20
	$60 \times 30 \times 10$	58.15	90.74	391.20	483.20	2.00	2.00	0.40	3.40
	$70 \times 35 \times 10$	84.72	146.53	468.40	621.20	1.60	1.60	0.80	3.40
	$80 \times 40 \times 20$	110.78	119.45	479.20	500.00	1.80	1.80	0.00	2.80
	$90 \times 45 \times 20$	251.28	266.10	740.80	762.00	1.80	1.80	0.60	3.40
	$100 \times 50 \times 20$	416.68	436.81	936.80	955.60	2.20	2.20	0.00	3.20
	$110 \times 55 \times 30$	376.67	396.00	713.20	718.00	1.00	1.00	0.00	2.00
	Average ($R = 10$)	124.47	144.20	408.87	499.78	1.47	1.49	2.07	5.22
	Global Average	148.50	164.85	501.50	583.34	1.35	1.37	1.25	4.03

In what follows, we comment on the results reported in Table 3.3, where the term "group of instances" will be simply replaced by instance, since results are shown in averages:

- The value of the CPU time provided by BCM (column 3) is greater than the one achieved by CMM (column 4) for all tested instances.
  The following figure 3.2 illustrates the performance of our proposed BCM method compared to the CMM based on the CPU (in seconds)for all studied instances, where R=2 and R=10 respectively.

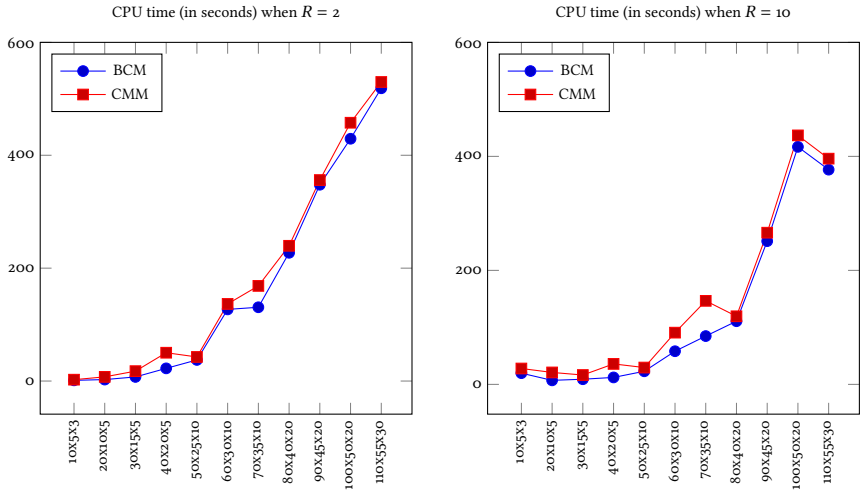


Figure 3.2: Illustration of the CPU (in seconds) achieved by both methods for all studied instances, where $R = 2$ and $R = 10$ respectively.

- In columns 5 and 6, we notice that the BCM method is competitive when compared to the performance of the CCM method in terms of the number of nodes during the branching process. Indeed, it achieves the optimal solutions with a number of nodes less than the ones generated by the CMM method throughout the searching procedure, and for all tested instances. Figure 3.3 demonstrates the average number of nodes required to achieve the optimal solution by both BCM and CMM methods for all studied instances for R=2 and R=10 respectively.

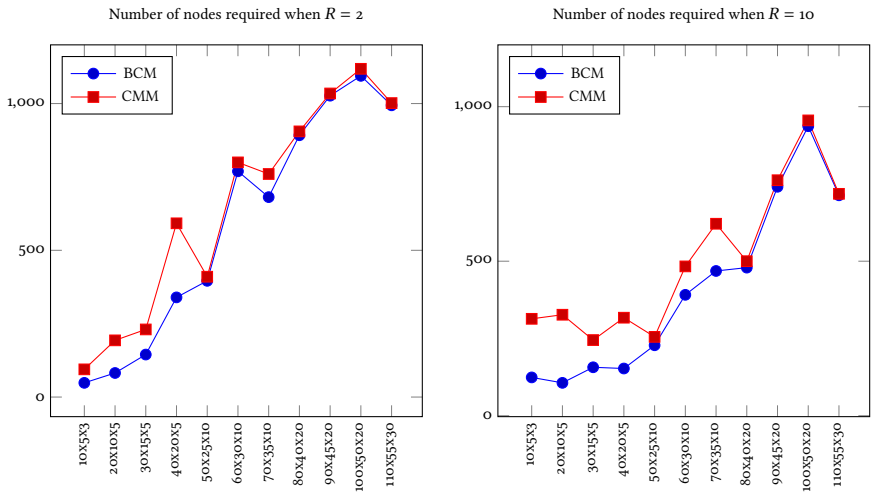


Figure 3.3: Illustration of the average number of nodes required to achieve the optimal solution by both methods for all studied instances for R=2 and R=10 respectively

- When we consider the number of solutions visited before finding the optimal solution (see, columns 7 and 8), we notice that our proposed method performs better according to this criterion over five cases (line 3, R=2), (line 1, R=3), (line 1 and 2, R=5) and (line 4, R=10) respectively, and remains very competitive for the remaining ones, where it is still able to achieve the same values found by CMM.

- Columns 9 and 10 show the number of cuts generated by both algorithms for each instance, where BCM is also able to achieve better performance. Besides being able to generate a more optimal number of cuts (less), we would like to point out that CMM generates a set of binary variables whenever a set of cuts are being added, which causes a higher complexity of the algorithm, on the other hand, in the case of BCM, no additional binary variables are added to the cutting process.

BCM versus CMM: we can observe that despite the power of CMM, BCM is able to provide improvements on different criteria for the majority of instances. In what follows, we comment on the average and global average results reported in Table 3.3:

- The average CPU time: BCM remains competitive even when it comes to the average time for each class of instances ((R=2), (R=3), (R=5), and (R=10)). Where it's able to provide an average CPU time of 168.32, 174.72, 126.48 and 124.47 seconds for each scenario, on the other hand, CMM was capable of reaching a CPU time of 182.59, 190.30, 142.30 and 144.20 seconds for the 4 existing variety on the number of scenarios (R) (see figure 3.2).

- The average number of nodes: BCM is also able to provide better average values on the number of nodes generated during the searching process, where BCM is capable of achieving an average of 588.04, 563.27, 445.82 and 408.87 created nodes for all scenarios, in parallel CMM was able to generate the following averages: 649.02, 647.38, 537.16 and 499.78 for all groups of instances (see figure 3.3).

- The average number of solutions: BCM is still competitive when it comes to the average number of solutions compared to those obtained by CMM. However, BCM is able to provide an average of 1.27, 1.22, 1.42 and 1.47 solution, where CMM was able to generate the following averages: 1.29, 1.24, 1.47 and 1.49 of solutions.

- The average number of cuts: For this criteria, BCM still outperforms the CMM method for all averages of 0.80, 0.93, 1.18 and 2.07 number of cuts, compared to 3.27, 3.49, 4.15 and 5.22 found by the CMM method.

- Global Average: Last but not least, a global average for each criterion has been also added on the last line of the table for a global comparison of the behavior of both methods for all studied criteria, where BCM was able to provide significant global averages of 148.50, 501.50, 1.35 and 1.25 compared to those obtained by CMM, 164.85, 583.34, 1.37 and 4.03.

The following figure 3.4 displays the average and global average (*Glb_avrg*) of the CPU (in seconds) achieved by the BCM and CMM methods for all the studied groups.

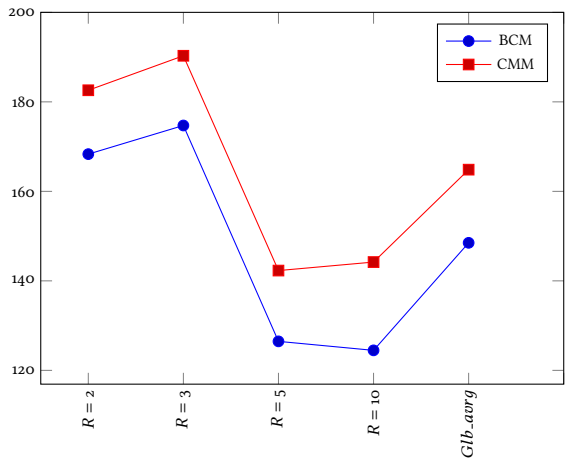


Figure 3.4: Illustration of the average and the global average (*Glb_avrg*) of CPU (in seconds) achieved by both methods for all studied groups

3.6 Conclusion

This chapter introduces a novel exact method that optimizes a linear function over an efficient set of a MOSILP problem. Our proposed method combines two established methods. The first is known as the L-shaped method, which is an algorithm adept at addressing two-stage stochastic programming problems. Its primary function involves assessing both the feasibility and optimality of the second-stage problems. The second one known as the branch-and-cut strategy, whose general principle is based on the branch-and-bound technique strengthened by the efficient tests and cuts, which allows the exploration of a limited number of nodes in the search tree. This combination of methods allows the optimal solution to be obtained without visiting the entire efficient set. Furthermore, our method underwent testing using randomly generated instances, incorporating variant data dimension sizes. The computational results showed the competitive nature of the proposed method, showcasing its performance relative to the existing CCM method. According to our modest experience in this field, we affirm that the class of optimization over an efficient set in the stochastic environment is extensive and has received comparatively less attention than other problem categories. The inclusion of the nonlinear variant in this study is particularly intriguing, as it mirrors various practical scenarios for research. Consequently, we propose investigating the multiobjective stochastic model with the incorporation of the nonlinear form. Additionally, exploring this problem with nonlinear preference functions of decision-makers, alongside continuous variables, would contribute valuable insights.

Chapter

4

Biobjective integer stochastic optimization over the integer stochastic efficient set

> *"Mathematics is the art of giving the same name to different things."*
> *"Les mathématiques sont l'art de donner le même nom à des choses différentes."*
> **Henri Poincaré**

Abstract: In this chapter, we focus on presenting an exact method to optimize two preference functions over the efficient set of a MOSILP problem in order to find the best compromise solutions without enumerating all elements of this efficient set. The proposed method is based on the combination of two techniques: the first one is called the L-shaped method, while the second one is an adaptation of the branch-and-bound strategy by reinforcing it with efficient cuts and tests, which allows the method to remove a large number of inefficient solutions within the search tree. Moreover, the method exhibits adaptability in addressing the general case involving multiple preference functions. A didactic example is presented for illustration, followed by a computational study evaluating the method's efficacy and performance by solving randomly generated instances.

4.1 Introduction

MOP plays an important role in dealing with real-world problems especially when multiple decision-makers or stakeholders are involved in the same problem, making it difficult to decide which single goal to achieve since there may be conflicting objectives that need to be balanced. Imagine a situation where one group wants to maximize profits, while another wants to minimize costs, which makes balancing these conflicting objectives quite challenging. Furthermore, MOP can handle effectively both deterministic environments which refer to situations where all data are known with certainty, such as the works of [Obal et al., 2013, Shidpour et al., 2013, Cao et al., 2018, Ren et al., 2021, Zhang et al., 2021, Halffmann et al., 2022, Motahari et al., 2023], and stochastic cases, which present intricate challenges due to the involvement of uncertain or random parameters, including works of [Goicoechea et al., 1976, Urli and Nadeau, 1990, Abdelaziz and Masri, 2010, Bozorgi-Amiri et al., 2013,

Ramezani et al., 2013, Moayedi and Sadeghian, 2023].

The biggest challenge in the MOP realm is the necessity of balancing the multiple conflicting objectives, this challenge can be tackled by identifying a set of solutions known as an efficient solution set. The size of this efficient set is influenced by both the complexity of the problem and the computing resources available, and as the problem size grows, the efficient set has the potential to expand significantly. Over the last few years, numerous authors have addressed this challenge of identifying the efficient set in deterministic cases, for example, [Jahanshahloo et al., 2004, Lokman and Köksalan, 2013, Rasmi and Türkay, 2019, Tamby and Vanderpooten, 2021]. In contrast, for stochastic cases, the presence of these stochastic parameters heightens the complexity of the problem. It is noteworthy that a limited number of studies have addressed this specific challenge, such as [Abbas and Bellahcene, 2006, Amrouche and Moulaï, 2012].

In numerous real-world problems, decision-makers often find themselves in front of a multitude of efficient solutions, which places them in difficult situations where they must carefully evaluate and select the best solution among all these efficient solutions. One effective approach to address this problem is to employ optimization techniques that allow decision-makers to fine-tune their preference function, and optimize it over the set of efficient solutions to ultimately obtain the optimal solution that aligns with their preferences. It is worth noting that the classic approach of enumerating all the efficient solutions is not recommended, due to its impracticality and the significant computational complexity it entails.

Optimization over the efficient set is a well-established and dynamic research field that is constantly evolving. It was first considered by [Philip, 1972], who made a significant contribution to this field. Because of its substantial implications for various research endeavors, this field has garnered the attention of numerous scholars from its inception. Indeed, it has been studied in several notable works such as [Benson, 1984, Ecker and Song, 1994]. Subsequently, based on the previous works, [Abbas and Chaabane, 2006] developed the first method for optimizing a linear function over the efficient set without finding all non-dominated points. Following that, [Jorge, 2009] proposed an exact algorithm to optimize a linear function over the integer efficient set of a MOILP problem. Later, [Ouaïl et al., 2017] introduced a branch and bound-based method to optimize a linear function over the efficient set of a MOILP problem. After that, [Boland et al., 2017] described a new algorithm to solve the same problem by modifying the algorithm of [Jorge, 2009]. In a different direction, [Sierra Altamiranda and Charkhgard, 2019] developed the first criterion space search algorithm for optimizing a linear function over the set of efficient solutions of Bi-objective Mixed Integer Linear Programs (BOMILP). Later on, [Lokman, 2021], based on the work of [Jorge, 2009] and [Boland et al., 2017] developed two algorithms to optimize a linear function over the nondominated set of MOIP problem. Most recently, [Belkhiri et al., 2022] proposed a new methodology to search for an efficient extreme point that optimizes a linear function over the set of efficient extreme points of a convex polyhedron. Furthermore, this area of research has extended further than linear cases, with many authors looking at non-linear scenarios, such as the work of [Zerdani and Moulai, 2011], where they developed an algorithm that optimizes an arbitrary linear function over an integer efficient set of a

Multi-objective Linear Fractional Programming (MOLFP) problem. Later on, [Drici et al., 2018] proposed a new exact method to maximize a linear fractional function over the integer efficient set of a MOILP problem. After that, [Moulaï and Drici, 2018] presented a new exact method for solving the maximization of an indefinite quadratic utility function over the efficient set of a MOILP problem. Recently, [Chaiblaine and Moulaï, 2021] described an exact method to optimize a quadratic function over the efficient set of a Multi-objective Integer Linear Fractional Programming (MOILFP) problem. However, it's worth noting that in stochastic cases, this specific problem has attracted less attention compared to its deterministic counterparts. To the best of our knowledge, the only work that addresses this problem is by [Chaabane and Mebrek, 2014], where their research is primarily centered on optimizing a linear function over the efficient set of MOSILP problem.

In this research endeavor, our focus is directed toward a new challenge known as bi-objective optimization over an efficient set. Essentially, this challenge can arise when two decision-makers or more collaborate on the same MOILP problem, with each of them possessing their unique preference objectives and visions. This situation presents a complex task of optimizing their individual preference objectives over the efficient solutions set of this problem in order to ultimately identify the best compromise solutions that satisfy the preferences of each decision-maker. In addition, this challenge also occurs when it comes to identifying the intersection between two efficient solution sets arising from two distinct MOILP and Bi-objective Integer Linear Programming (BOILP) problems to determine the common optimal solutions shared between these two sets. Nowadays, BOILP has become a sophisticated and well-established field in the operations research domain, marked by significant contributions from various authors, including [Soylu, 2015, Boland et al., 2015, Adelgren and Gupte, 2022, Bongo and Sy, 2023]. Despite the important roles played by both BOILP and optimization over an efficient set in various fields, the intersection of these two areas remains comparatively less explored and developed. Notably, few references delve into this particular area, such as the work of [Chaiblaine et al., 2020], where the authors introduced an exact method for optimizing two fractional linear functions over the efficient set of a MOILFP problem. After that, [Cherfaoui and Moulaï, 2021] presented an exact method for optimizing two preference functions over the efficient set of a MOILP problem. This significant gap between BOILP and optimization over the efficient set offers an interesting direction for further exploration and progress in the research operations domain.

The main objective of our contribution is to address this new challenge, specifically tackling the resolution of the problem mentioned above within a stochastic environment. To the best of our knowledge, no prior study has addressed this specific problem thus far. The only study that tackled a similar problem is the previously mentioned work by [Chaabane and Mebrek, 2014], It is worth noting that their work focused on optimizing a single linear function over an efficient set of a MOSILP problem. In contrast, our research takes a different approach by concentrating on the optimization of two preference functions over the efficient set. To be precise, we introduces an exact method to optimize two preference functions over an efficient set of a MOSILP problem. The proposed method is a combination of two approaches: the first one is known as the L-shaped method which is an iterative algorithm used to solve the two-stage stochastic linear programming problems, see

[Li and Grossmann, 2018, Ušpurienė et al., 2018, Dos Santos and Oliveira, 2019, Torres et al., 2022]. In this context, first-stage decisions precede the realization of uncertainty, while second-stage decisions occur after the realization of uncertain parameters. One of the most significant roles of the L-shaped method is its efficiency in handling large-scale problems with numerous scenarios. Instead of solving the entire problem at once, it decomposes the original stochastic problem into two manageable levels: a master deterministic problem and subproblems, where the master problem is solved in the first level and provides an initial solution. After that, the subproblems are solved iteratively, incorporating information from realized scenarios to improve the solution. For more details see [Kall, 1976, Van Slyke and Wets, 1969, Kall et al., 1994, Birge and Louveaux, 2011]. The second approach involves adapting the branch-and-bound procedure to suit our needs, this adaptation is enhanced by the implementation of efficient cuts and tests, enabling the removal of a substantial number of inefficient solutions within the search tree.

4.2 Preliminaries

4.2.1 Problem formulation

Our primary objective is to solve the problem involving the optimization of two preference functions over the efficient set of the MOSILP problem (3.1). This can be formulated as follows:

$$(P)\begin{cases} \min \quad \phi_1 = d_1(\xi)x, \\ \min \quad \phi_2 = d_2(\xi)x, \\ s.t., \\ \quad x \in \mathcal{X}_E, \end{cases} \tag{4.1}$$

where $d_1(\xi)$ and $d_2(\xi)$ are two random vectors in $\mathbb{R}^n$ that represent the preference functions of decision-makers. $\mathcal{X}_E$ denotes the efficient solution set of the MOSILP problem (3.1).
The main difficulty in solving problem (4.1) is the fact that it considers several stochastic objective functions simultaneously.

4.2.2 Equivalent deterministic problem

Considering preference objectives, we express $\phi_i^r = d_i(\xi^r)x$ for $i \in \{1, 2\}$, where $d_i(\xi^r)x$ represents the preference objective for the ith criterion in the rth scenario. In this context, the deterministic equivalent of our primary problem is defined as follows:

$$(P_D)\begin{cases}\min & \widetilde{\phi}_1 + Q(x),\\ \min & \widetilde{\phi}_2 + Q(x),\\ s.t., & \\ & x \in \mathcal{X}_{ED},\end{cases} \tag{4.2}$$

where

$$\begin{cases}\widetilde{\phi}_1 = \mathbb{E}[\phi_1^r] = \mathbb{E}[d_1(\xi)x] = \sum_{r=1}^{R} P^r d_1(\xi^r)x = \widetilde{d}_1 x,\\ \widetilde{\phi}_2 = \mathbb{E}[\phi_2^r] = \mathbb{E}[d_2(\xi)x] = \sum_{r=1}^{R} P^r d_2(\xi^r)x = \widetilde{d}_2 x,\\ Q(x) = \mathbb{E}[Q(x,\xi)] = \sum_{r=1}^{R} P^r (q^r)^\intercal y^r.\end{cases}$$

Here, $\mathcal{X}_{ED}$ signifies the set of efficient solutions for the deterministic equivalent problem (3.3). Furthermore, the relaxed problem of our main problem can be defined as follows:

$$(P_R)\begin{cases}\min & \widetilde{\phi}_1 = \widetilde{d}_1 x + \mathbb{E}[Q(x,\xi)],\\ \min & \widetilde{\phi}_2 = \widetilde{d}_2 x + \mathbb{E}[Q(x,\xi)],\\ s.t, & \\ & x \in \mathcal{S} = \{x \in \mathbb{N}^n | Ax = b\},\end{cases} \tag{4.3}$$

4.3 Methodology, algorithm and theoretical results

In the following subsection, we detail each step of our methodology that is designed to solve the problem described in (4.2). Following this, we describe the algorithm, giving a comprehensive overview of the essential steps in our proposed method. After that, we turn to a discussion of various theoretical results. Finally, we demonstrate how to extend the method to general multi-decision cases.

4.3.1 Methodology description

The proposed algorithm aims to generate a subset $\mathcal{X}_B \subset \mathcal{X}_E$, where $\mathcal{X}_E$ is the efficient set of the deterministic equivalent problem (2.24). The elements in $\mathcal{X}_B$ are selected to be also efficient in terms of both objective functions d_1 and d_2. Contrary to the classical approach, our algorithm avoids the exhaustive enumeration of all elements in $\mathcal{X}_E$ to generate $\mathcal{X}_B$. The classical method for solving problem (4.2) is to determine two sets $\mathcal{X}_E$ and $\mathcal{X}_C$, where $\mathcal{X}_C$ represents the efficient solutions in terms of d_1 and d_2. Then we have to select the elements that are in both sets $\mathcal{X}_E$ and $\mathcal{X}_C$ simultaneously, which is very computationally expensive. As mentioned previously, the proposed algorithm employs two distinct techniques to address the problem. These include the well-known L-shaped method and an

adaptation of the branch-and-bound procedure, reinforced by efficient cuts and tests. To start the process, we initialize the two variables, θ and l, to $-\infty$ and 0, respectively. Furthermore, at each node l, the algorithm resolves the linear program (4.4) using the simplex or dual simplex method.

$$(P^l)\begin{cases}\min \quad \widetilde{\phi}_1 = \widetilde{d}_1 x, \\ s.t., \\ \quad x \ \in \mathcal{S}_l = \{x \in \mathbb{N}^n | Ax = b\}.\end{cases} \tag{4.4}$$

Once the linear program (4.4) has been solved, the algorithm checks for the existence of a solution. In case the program has no solution, the node is fathomed. Otherwise, there are two possible situations:

1. The optimal solution x^l is not an integer. in this situation, the algorithm follows a basic branching process according to the following steps: Identify a component x_j of x^l such that $x_j = \alpha_j$, where α_j represents a fractional value. After that, the node l of the tree is separated into two nodes which are imposed by the following two additional constraints:

$$\begin{cases} x_j \ \leq \lfloor \alpha_j \rfloor, \\ x_j \ \geq \ \lfloor \alpha_j \rfloor + 1, \end{cases} \tag{4.5}$$

   where, $\lfloor \alpha_j \rfloor$ indicates the greatest integer less than α_j. The linear program obtained must be solved until an integer feasible solution is found (if it exists).

2. The solution x^l is an integer. Here, the process passes through two tests: the feasibility and optimality tests, which are detailed in subsections (2.2.4).

As soon as an optimal integer solution x^l is found after the feasibility and optimality tests, then two efficiency tests are performed by solving the programs 4.6 and 4.7 (see, subsection 1.3.1).

1. In the first step, we test the efficiency of the solution x^l in terms of $\widetilde{d}_1$ and $\widetilde{d}_2$ by solving the following first program:

$$(EK^1(x^l))\begin{cases} \max \sum_{i=1}^{2} v_i, \\ s.t., \\ \widetilde{d}_i x + v_i = \widetilde{d}_i x^l, \ i \in \{1, 2\}, \\ x \in \mathcal{S}_l, \\ v_i \geq 0, \ i \in \{1, 2\}. \end{cases} \tag{4.6}$$

   As a well-known result, x^l is efficient if and only if $EK^1(x^l)$ has a maximum value of zero. See, [Ecker and Kouada, 1978].

2. The process continues by passing the second efficiency test for the deterministic equivalent problem (2.24).

$$(EK^2(x^l))\begin{cases} \max \sum_{i=1}^{K} w_i, \\ s.t., \\ \widetilde{C}_i x + w_i = \widetilde{C}_i x^l,\ i \in \{1, 2, \ldots, K\}, \\ x \in \mathcal{S}_l, \\ w_i \geq 0,\ i \in \{1, 2, \ldots, K\}. \end{cases} \tag{4.7}$$

Based on the same previous concept, the solution x^l is efficient for the deterministic equivalent problem (2.24) if and only if $EK^2(x^l)$ has a maximum value of zero.

As a result, if both $EK^1(x^l)$ and $EK^2(x^l)$ have zero maximum value, then x^l belongs to $\mathcal{X}_B$. The process continues to search for other efficient solutions in other nodes (if they exist) by applying efficient cuts to the related program (see, subsection 3.2.2). Initially, two sets $\mathcal{H}_l$ and $\mathcal{H}_l'$ are constructed. After that, the efficient cuts (4.8) and (4.9) are constructed and incorporated into the successor nodes of l.

$$\mathcal{H}_l = \left\{ j \in \mathcal{N}_l | \exists i \in \{1, 2, \ldots, K\},\ \bar{C}_i^j < 0 \right\} \cup \left\{ j \in \mathcal{N}_l |\ \bar{C}_i^j = 0,\ \forall i \in \{1, 2, \ldots, K\} \right\},$$

$$\sum_{j \in \mathcal{H}_l} x_j \geq 1, \tag{4.8}$$

And

$$\mathcal{H}_l^{'} = \left\{ j \in \mathcal{N}_l |\ \bar{d}_2^j < 0 \right\} \cup \left\{ j \in \mathcal{N}_l |\ \bar{d}_1^j = 0 \text{ and } \bar{d}_2^j = 0 \right\},$$

$$\sum_{j \in \mathcal{H}_l'} x_j \geq 1. \tag{4.9}$$

Hence, the domain of the successor node $l+1$ is determined by applying the efficient cuts (4.8) and (4.9) to $\mathcal{S}_l$.

$$\mathcal{S}_{l+1} = \mathcal{S}_{l+1}^1 \cap \mathcal{S}_{l+1}^2, \tag{4.10}$$

where

$$\begin{cases} \mathcal{S}_{l+1}^1 = \left\{ x \in \mathcal{S}_l |\ \sum_{j \in \mathcal{H}_l} x_j \geq 1 \right\}, \\ \\ \mathcal{S}_{l+1}^2 = \left\{ x \in \mathcal{S}_l |\ \sum_{j \in \mathcal{H}_l'} x_j \geq 1 \right\}. \end{cases}$$

The method ends when all the created nodes are fathomed or when one of the sets ($\mathcal{H}_l$ or $\mathcal{H}_l^{'}$) is empty, which means that $\mathcal{H}_l = \emptyset$ or $\mathcal{H}_l^{'} = \emptyset$.

4.3.2 Algorithm

The following algorithm 14 resumes in detail the main steps of the proposed method, these steps are treated according to the backtracking principle.

Algorithm 14: Biobjective Stochastic Programming Method

Data: $\mathcal{X}_B = \emptyset$, $\theta = -\infty$, $l = 0$.
Result: $\mathcal{X}_B$: The solutions set of problem (4.2).
while *As long as a unfathomed node l exists in the tree* **do**
  Solve problem (4.4) using simplex or dual simplex method;
  if *Problem (4.4) has a feasible solution x^l* **then**
    if *x^l is integer* **then**
      Feasibility test.
      Solve program (2.13);
      if $\sigma^\intercal [h(\xi^r) - T(\xi^r)x^l] > 0$ **then**
        Add the feasibility cut (2.14) to the successors of l;
      else
        Optimality test.
        Solve program (2.16), calculate Q(x);
        if $Q(x^l) > \theta^l$ **then**
          Add the optimality cut (2.19) to the successors of l;
        else
          Efficiency test.
          Solve program (4.6);
          v is the optimal solution criteria;
          if $v = 0$ **then**
            Solve program (4.7);
            w is the optimal solution criteria;
            if $w = 0$ **then**
              $\mathcal{X}_B = \mathcal{X}_B \cup \{x^l\}$;
            end
          end
          Update set $\mathcal{S}$.
          Construct the sets $\mathcal{H}_l$ and $\mathcal{H}'_l$;
          if *$\mathcal{H}_l$ or $\mathcal{H}'_l = \emptyset$* **then**
            Fathom the node l;
          else
            Add the efficient cuts (4.8) and (4.9) to the successors of l;
          end
        end
      end
    else
      Branching process.
      Choose an index j such that α_j is the most fractional number, then:
      Split the program P^l into two sub programs, by adding respectively
      the constraints $x_j \leq \lfloor \alpha_j \rfloor$ and $x_j \geq \lfloor \alpha_j \rfloor + 1$, to obtain (P^{l_1}) and (P^{l_2})
      $(l_1 \neq l_2$ and $l_1 \geq l + 1$, $l_2 \geq l + 1)$;
    end
  else
    The corresponding node l is fathomed;
  end
end

4.3.3 Theoretical Results

The following theoretical tools show that the algorithm yields the set of solutions for problem (4.2) in a finite number of iterations.

Theorem 4.3.1. *Assume that $\mathcal{H}_l \neq \emptyset$ and $\mathcal{H}_l^{'} \neq \emptyset$ at the current integer solution x^l. If $x \neq x^l$ is an efficient solution of problem (4.2) in domain $\mathcal{S}_l$, then, $x \in \mathcal{S}_{l+1}$ ($l+1$ is the successor of l).*

Proof. Let $x \neq x^l$ be an integer solution in domain $\mathcal{S}_l$, such that, $x \notin \mathcal{S}_{l+1}$. Two situations can occur in this case:

1. $x \notin \mathcal{S}_{l+1}^1$, implies that $x \in \left\{x \in \mathcal{S}_l |\ \sum_{j \in \mathcal{N}_l \setminus \mathcal{H}_l} x_j \geq 1\right\}$. The components of vector x here satisfy the following inequalities:

$$\begin{cases} \sum\limits_{j \in \mathcal{H}_l} x_j < 1, \\ \sum\limits_{j \in \mathcal{H}_l \setminus \mathcal{H}_l} x_j \geq 1. \end{cases}$$

   Which means that $x_j = 0$, for all $j \in \mathcal{H}_l$ and $x_j \geq 1$, for at least one index $j \in \mathcal{N}_l \setminus \mathcal{H}_l$.
   On the other hand, for all criterion $i \in \{1, 2, \ldots, K\}$, the following equality is supported using the simplex table:

$$\widetilde{f_i}(x) = \widetilde{C}_i x = \sum_{j \in \mathcal{B}_l} \overline{C}_i^j x_j + \sum_{j \in \mathcal{N}_l} \overline{C}_i^j x_j, \text{ where } \sum_{j \in \mathcal{B}_l} \overline{C}_i^j x_j = \widetilde{f_i}(x^l).$$

   Hence, we can write as follows:

$$\begin{aligned} \widetilde{f_i}(x) - \widetilde{f_i}(x^l) &= \sum_{j \in \mathcal{N}_l} \overline{C}_i^j x_j, \\ &= \sum_{j \in \mathcal{H}_l} \overline{C}_i^j x_j + \sum_{j \in \mathcal{N}_l \setminus \mathcal{H}_l} \overline{C}_i^j x_j, \\ &= \sum_{j \in \mathcal{N}_l \setminus \mathcal{H}_l} \overline{C}_i^j x_j. \end{aligned}$$

   Thus, this implies that $\widetilde{f_i}(x) \leq \widetilde{f_i}(x^l)$, for all criterion $i \in \{1, 2, \ldots, K\}$. As well as $\widetilde{f_i}(x) < \widetilde{f_i}(x^l)$, for at least one criterion of $\overline{C}_i^j \leq 0$, for all $j \in \mathcal{N}_l \setminus \mathcal{H}_l$.
   We conclude then that the solution x is not efficient for the deterministic equivalent problem 2.24. In other words, this means that $x \notin \mathcal{X}_E$, and $x \notin \mathcal{X}_B$.

2. $x \notin \mathcal{S}_{l+1}^2$ implies that $x \in \left\{x \in \mathcal{S}_l |\ \sum_{j \in \mathcal{N}_l \setminus \mathcal{H}_l^{'}} x_j \geq 1\right\}$. The components of vector x in this situation satisfy the following inequalities:

$$\begin{cases} \sum\limits_{j \in \mathcal{H}_l^{'}} x_j < 1, \\ \sum\limits_{j \in \mathcal{N}_l \setminus \mathcal{H}_l^{'}} x_j \geq 1. \end{cases}$$

Which means that $x_j = 0$, for all $j \in \mathcal{H}_l^{'}$ and $x_j \geq 1$, for at least one index $j \in \mathcal{N}_l \setminus \mathcal{H}_l^{'}$.
Furthermore, by using the simplex table in x^l, the following equality is satisfied:

$$\widetilde{d}_2 x = \sum_{j \in \mathcal{B}_l} \overline{d}_2^j x_j + \sum_{j \in \mathcal{N}_l} \overline{d}_2^j x_j \text{ where } \sum_{j \in \mathcal{B}_l} \overline{d}_2^j x_j = \widetilde{d}_2 x^l,$$

Here, we can write as follows:

$$\begin{aligned} \widetilde{d}_2 x - \widetilde{d}_2 x^l &= \sum_{j \in \mathcal{H}_l^{'}} \bar{d}_2^j x_j + \sum_{j \in \mathcal{N}_l \setminus \mathcal{H}_l^{'}} \bar{d}_2^j x_j, \\ &= \sum_{j \in \mathcal{N}_l \setminus \mathcal{H}_l^{'}} \bar{d}_2^j x_j. \end{aligned}$$

Thus, $\widetilde{d}_2 x \leq \widetilde{d}_2 x^l$, with $\widetilde{d}_2 x < \widetilde{d}_2 x^l$ and $\widetilde{d}_1 x \leq \widetilde{d}_1 x^l$, since x^l is the optimum of the current problem. This means that $x \notin \mathcal{X}_B$.
Since $x \notin \mathcal{X}_B$ in both situations. As a result, x is not an efficient solution of problem (4.2).

□

Proposition 4.3.1. *Suppose that $\mathcal{H}_l = \emptyset$ or $\mathcal{H'}_l = \emptyset$ at the current integer solution x^l. Then there is no solution in the remaining domain that is not dominated by x^l.*

Proof.

- Suppose that $\mathcal{H}_l = \emptyset$, which means that we have $\bar{C}_i^j \leq 0, \forall j \in \mathcal{N}_l, \forall i \in \{1, 2, \ldots, K\}$. As well as, $\forall j \in \mathcal{N}_l, \exists i_0 \in \{1, 2, \ldots, K\}$, such that $\bar{C}_{i_0}^j < 0$. This implies that x^l dominates all points x, $x \neq x^l$ of domain $\mathcal{S}_l$.
- Now assume that $\mathcal{H}_l^{'} = \emptyset$, and this means that $\forall j \in \mathcal{N}_l,\ \bar{d}_2^j < 0$ or $\bar{d}_2^j = 0,$ and $\bar{d}_1^j < 0$, leading to $\bar{d}_1^j < 0,\ \forall j \in \mathcal{N}_l$. Since it is an optimal solution for (P^l). Thus, x^l becomes the most preferred solution in the domain $\mathcal{S}_l$.

□

Theorem 4.3.2. *Algorithm 14 terminates in a finite number of iterations and the set $\mathcal{X}_B$ contains all the solutions of problem (4.2).*

Proof. Let $\mathcal{S}$ be the set of feasible integer solutions of the deterministic equivalent problem 2.24 a bounded set. Moreover, the efficient sets $\mathcal{X}_B$ and $\mathcal{X}_E$ contain a finite number of integer solutions which means that the cardinality of these sets is a finite number. For each step l of the algorithm 14, when an integer solution is found, there are only a finite number of cuts that eliminate x^l and all dominated solutions from the search tree (see Proposition 4.3.1). Similarly, the algorithm used a limited number of feasibility and optimality cuts. which means that the search tree would have a finite number of branches and that the algorithm terminates in a finite number of steps.
For each step l of Algorithm 14, when an integer solution x^l is found, the cuts eliminate x^l and all

dominated solutions from the search tree (see Proposition 4.3.1). In addition, for $\mathcal{X}_B$ to contain all the solutions of problem (4.2), the fathoming rules are used without loss of any elements in $\mathcal{X}_B$. The first rule is when the set $\mathcal{H}_l$ or $\mathcal{H}_l^{'}$ is empty. In this case, the current node can be fathomed since the rest of the domain contains only dominated solutions either in terms of the deterministic equivalent problem or in terms of the two preference functions. The second rule is the trivial case when the reduced domain becomes infeasible, whether it is because of previous cuts or the branching. □

4.3.4 Generalization of the method

The same algorithm can be used to solve the problem where there are p decision-makers, the general problem can be formulated as follows:

$$(P)\begin{cases} \min & \phi_1 = d_1(\xi)x, \\ \min & \phi_2 = d_2(\xi)x, \\ \vdots & \\ \min & \phi_p = d_p(\xi)x, \\ s.t., & \\ & x \in \mathcal{X}_E. \end{cases} \tag{4.11}$$

To solve this problem, we use our algorithm and modify the efficiency test $EK(x^l)$ as follows:

$$(EK(x^l))\begin{cases} \max \sum_{i=1}^{p} v_i, \\ s.t., \\ \widetilde{d}_i x - v_i = \widetilde{d}_i x^l, \ i \in \{1, 2, \ldots, p\}, \\ x \in \mathcal{S}_l, \\ v_i \geq 0, \ i \in \{1, 2, \ldots, p\}. \end{cases} \tag{4.12}$$

Also, $\mathcal{H}_l^{'}$ is calculated as follows:

$$\mathcal{H}_l^{'} = \left\{ j \in \mathcal{N}_l | \exists i \in \{1, 2, \ldots, p\}, \bar{d}_i^j \ < \ 0 \right\} \cup \left\{ j \in \mathcal{N}_l | \ \bar{d}_i^j = 0, \ \forall i \in \{1, 2, \ldots, p\} \right\}.$$

The rest of the algorithm remains the same. The algorithm returns the set of efficient solutions $\mathcal{X}_B = \mathcal{X}_E \cap \mathcal{X}_C$.

4.4 Didactic Example

Consider the following multi-objective integer linear programming stochastic problem:

Scenario 1

$$MOSILP(\xi^1)\begin{cases}\min f_1^1 = -9x_1 + 4x_2,\\ \min f_2^1 = 3x_1 - 5x_2,\\ \min f_3^1 = 5x_1 - 11x_2,\\ \text{s.t.},\\ x_1 - 4x_2 \leq 3,\\ x_1 + 3x_2 \leq 18,\\ 2x_1 + x_2 \leq 10,\\ x_1 + 2x_2 \leq 3,\\ -3x_1 + x_2 \leq 5,\\ x_1,\ x_2 \in \mathbb{N}.\end{cases}$$

Scenario 2

$$MOSILP(\xi^2)\begin{cases}\min f_1^2 = 3x_1 - 2x_2,\\ \min f_2^2 = 6x_1 + x_2,\\ \min f_3^2 = -7x_1 + 10x_2,\\ \text{s.t.},\\ x_1 - 4x_2 \leq 3,\\ x_1 + 3x_2 \leq 18,\\ 2x_1 + x_2 \leq 10,\\ 5x_1 + x_2 \leq 4,\\ -3x_1 + 2x_2 \leq 1,\\ x_1,\ x_2 \in \mathbb{N}.\end{cases}$$

The recourse matrix and the deterministic constraints matrices are given for both scenarios by:

$$A = \begin{pmatrix} 1 & -4 \\ 1 & 3 \\ 2 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 3 \\ 18 \\ 10 \end{pmatrix}, W = \begin{pmatrix} -2 & -1 & 2 & 1 \\ 3 & 2 & -5 & -6 \end{pmatrix}.$$

The penalties of constraint violations and the probability distribution are given for both scenarios by:

$$q(\xi^1) = (1, 3, 6, 2)^\intercal, \quad \mathbb{P}(\xi^1) = \tfrac{2}{3}, \quad q(\xi^2) = (5, 3, 2, 1)^\intercal, \quad \mathbb{P}(\xi^2) = \tfrac{1}{3}.$$

The stochastic constraints matrices are given for both scenarios by:

$$T(\xi^1) = \begin{pmatrix} 1 & 2 \\ -3 & 1 \end{pmatrix}, T(\xi^2) = \begin{pmatrix} 5 & 1 \\ -3 & 2 \end{pmatrix}, h(\xi^1) = \begin{pmatrix} 3 \\ 5 \end{pmatrix}, h(\xi^2) = \begin{pmatrix} 4 \\ 1 \end{pmatrix}.$$

The equivalent deterministic multiple objective integer linear programming problem:

$$\widetilde{f_1} = \mathbb{E}\left[C_1(\xi)\, x\right] = \mathbb{P}(\xi^1)\, C_1(\xi^1)\, x + \mathbb{P}(\xi^2)\, C_1(\xi^2)\, x,$$
$$\widetilde{f_2} = \mathbb{E}\left[C_2(\xi)\, x\right] = \mathbb{P}(\xi^1)\, C_2(\xi^1)\, x + \mathbb{P}(\xi^2)\, C_2(\xi^2)\, x,$$
$$\widetilde{f_3} = \mathbb{E}\left[C_3(\xi)\, x\right] = \mathbb{P}(\xi^1)\, C_3(\xi^1)\, x + \mathbb{P}(\xi^2)\, C_3(\xi^2)\, x.$$

$$(MOSILP_D)\begin{cases} \min\ \widetilde{f_1} = -5x_1 + 2x_2, \\ \min\ \widetilde{f_2} = 4x_1 - 3x_2, \\ \min\ \widetilde{f_3} = x_1 - 4x_2, \\ \text{s.t.,} \\ x_1 - 4x_2 \leq 3, \\ x_1 + 3x_2 \leq 18, \\ 2x_1 + x_2 \leq 10, \\ x_1, x_2 \in \mathbb{N}. \end{cases}$$

Consider now two decision makers with the following preference functions for each scenario.

Scenario 1 Scenario 2

$$P(\xi^1)\begin{cases} \min \phi_1^1 = -2x_1 + 4x_2, \\ \min \phi_2^1 = 2x_1 - 1x_2, \\ \text{s.t.,} \\ x_1,\ x_2 \in \mathcal{X}_E. \end{cases} \qquad P(\xi^2)\begin{cases} \min \phi_1^2 = x_1 + 4x_2, \\ \min \phi_2^2 = 2x_1 + 5x_2, \\ \text{s.t.,} \\ x_1,\ x_2 \in \mathcal{X}_E. \end{cases}$$

The main equivalent deterministic relaxed problem is defined by:

$$\widetilde{\phi_1} = \mathbb{E}\left[d_1(\xi)\, x\right] = \mathbb{P}(\xi^1)\, d_1(\xi^1)\, x + \mathbb{P}(\xi^2)\, d_1(\xi^2)\, x,$$
$$\widetilde{\phi_2} = \mathbb{E}\left[d_2(\xi)\, x\right] = \mathbb{P}(\xi^1)\, d_2(\xi^1)\, x + \mathbb{P}(\xi^2)\, d_2(\xi^2)\, x.$$

$$(P_{DR})\begin{cases} \min\ \widetilde{\phi_1} = -x_1 + 4x_2, \\ \min\ \widetilde{\phi_2} = 2x_1 + x_2, \\ \text{s.t.,} \\ x_1, x_2 \in \mathcal{S}, \end{cases}$$

where $\mathcal{S} = \{x_1 - 4x_2 \leq 3,\ x_1 + 3x_2 \leq 18,\ 2x_1 + x_2 \leq 10 \mid x_1, x_2 \in \mathbb{R}^n\}$.

Initialization: $\mathcal{X}_B = \emptyset,\ \theta = -\infty,\ \mathcal{S}_0 = \mathcal{S}$.
Step 1: We start by solving the first following relaxed problem (P^0).

$$(P^0)\begin{cases} \min\ \phi = -x_1 + 4x_2, \\ \text{s.t.,} \\ x \in \mathcal{S}_0. \end{cases}$$

The first optimal solution is $x^0 = (\frac{43}{9}, \frac{4}{9})$ which is not an integer. The algorithm starts the branching process.

Step 2: The branching process. The search nodes obtained after branching on x^0 are:
Node 1 $\mathcal{S}_1 = \{\, x \in \mathcal{S}_0 \ :\ x_1 \leq 4\}$.

Node 2 $\mathcal{S}_2 = \{x \in \mathcal{S}_0 \ : \ x_1 \geq 5\}$. No solution exists in this node. Thus, this node is **fathomed**.

Now, we continue the process by solving the new problem (P^1) at node 1.

$$(P^1)\begin{cases} \min \ \phi = -x_1 + 4x_2, \\ \text{s.t.}, \\ x \in \mathcal{S}_1. \end{cases}$$

The optimal solution is $x^1 = (4, \frac{1}{4})$. The algorithm will again perform branching process.

Step 2: The branching process. The obtained search nodes after branching on x^1 are:
Node 3 $\mathcal{S}_3 = \{x \in \mathcal{S}_2 \ : \ x_2 \geq 1\}$.
Node 4 $\mathcal{S}_4 = \{\ x \in \mathcal{S}_2 \ : \ x_2 \leq 0\}$.

In this situation, the process continues with solving problem (P^3) at node 3.

$$(P^3)\begin{cases} \min \ \phi = -x_1 + 4x_2, \\ \text{s.t.}, \\ x \in \mathcal{S}_3. \end{cases}$$

The obtained optimal integer solution is $x^2 = (4, 1)$. In this situation, x^2 is tested for second stage feasibility and optimality for each scenario.

Step 3: Feasibility and Optimality tests. At first, we evaluate the feasibility of solution x^2 by solving the following two problems:

$$h\left(\xi^1\right) - T\left(\xi^1\right) x^2 = \begin{pmatrix} 3 \\ 5 \end{pmatrix} - \begin{pmatrix} 1 & 2 \\ -3 & 1 \end{pmatrix}\begin{pmatrix} 4 \\ 1 \end{pmatrix} = \begin{pmatrix} -3 \\ 16 \end{pmatrix}.$$
$$h\left(\xi^2\right) - T\left(\xi^2\right) x^2 = \begin{pmatrix} 4 \\ 1 \end{pmatrix} - \begin{pmatrix} 5 & 1 \\ -3 & 2 \end{pmatrix}\begin{pmatrix} 4 \\ 1 \end{pmatrix} = \begin{pmatrix} -6 \\ 11 \end{pmatrix}.$$

Scenario 1

$$\sigma(\xi^1)\begin{cases} \max \ -3\sigma_1^1 + 16\sigma_1^2, \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0, \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0, \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0, \\ \sigma_1^1 - 6\sigma_1^2 \leq 0, \\ \sigma_1^1 + \sigma_1^2 \leq 1. \end{cases} \qquad \sigma_1^\intercal = \left(\tfrac{2}{3}, \tfrac{1}{3}\right),$$

Scenario 2

$$\sigma(\xi^2)\begin{cases} \max \ -6\sigma_2^1 + 11\sigma_2^2, \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0, \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0, \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0, \\ \sigma_2^1 - 6\sigma_2^2 \leq 0, \\ \sigma_2^1 + \sigma_2^2 \leq 1. \end{cases} \qquad \sigma_2^\intercal = (0, 0)\,.$$

Note that $\sigma_1^\intercal \left[h\left(\xi^1\right) - T\left(\xi^1\right) x^2\right] = \left(\frac{2}{3}, \frac{1}{3}\right)\binom{-3}{16} = \frac{10}{3} > 0$, which means that x^2 is not feasible for the first scenario ξ^1, In this case, we add the following feasibility cut to current problem (P^3).

$$\left(\tfrac{2}{3}, \tfrac{1}{3} \right) \begin{pmatrix} 1 & 2 \\ -3 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} > \left(\tfrac{2}{3}, \tfrac{1}{3} \right) \begin{pmatrix} 3 \\ 5 \end{pmatrix} = \frac{11}{3} \iff -\frac{1}{3}x_1 + \frac{5}{3}x_2 \geq \frac{11}{3}.$$

After adding the feasibility cut, we solve the following problem (P^5) in node 5:

$$(P^5) \begin{cases} \min\ \phi = -x_1 + 4x_2, \\ \text{s.t.,} \\ x \in \mathcal{S}_5 = \{x \in \mathcal{S}_3 |\ -\frac{1}{3}x_1 + \frac{5}{3}x_2 \geq \frac{11}{3}\}. \end{cases}$$

The optimal solution is $x^3 = (\frac{39}{11}, \frac{32}{11})$. The branching process is then started again.

Step 2: The branching process. After branching on x^3, we obtaine the following search nodes:
Node 6 $\mathcal{S}_6 = \{x \in \mathcal{S}_5 \ :\ x_1 \geq 4\}$. No solution exists in this node. This node is **fathomed**.
Node 7 $\mathcal{S}_7 = \{\ x \in \mathcal{S}_5 \ :\ x_1 \leq 3\}$.

The process continues in this step by solving problem (P^7) at node 7.

$$(P^7) \begin{cases} \min\ \phi = -x_1 + 4x_2, \\ \text{s.t.,} \\ x \in \mathcal{S}_7. \end{cases}$$

The obtained solution is $x^4 = (3, \frac{14}{5})$. The branching process begins again.

Step 2: The branching process. The search nodes obtained after branching on x^4 are:
Node 8 $\mathcal{S}_8 = \{x \in \mathcal{S}_7 \ :\ x_2 \geq 3\}$.
Node 9 $\mathcal{S}_9 = \{\ x \in \mathcal{S}_7 \ :\ x_2 \leq 2\}$. No solution exists in this node. Then, this node is **fathomed**.

We solve now problem (P^8) at node 8.

$$(P^8) \begin{cases} \min\ \phi = -x_1 + 4x_2, \\ \text{s.t.,} \\ x \in \mathcal{S}_8. \end{cases}$$

The obtained integer solution is $x^5 = (3, 3)$. We continue the process by testing the feasibility and optimality of the solution x^5.

Step 3: Feasibility and Optimality tests.
As previously mentioned, this step commences with a feasibility test, where the feasibility of the solution x^5 is examined by solving the following pair of problems.

Scenario 1 Scenario 2

$$\sigma(\xi^1)\begin{cases}\max\ -6\sigma_1^1+11\sigma_1^2,\\ -2\sigma_1^1+3\sigma_1^2\le 0,\\ -\sigma_1^1+2\sigma_1^2\le 0,\\ 2\sigma_1^1-5\sigma_1^2\le 0,\\ \sigma_1^1-6\sigma_1^2\le 0,\\ \sigma_1^1+\sigma_1^2\le 1.\end{cases}\quad \sigma_1^\intercal=(0,0),\qquad \sigma(\xi^2)\begin{cases}\max\ -14\sigma_2^1+4\sigma_2^2,\\ -2\sigma_2^1+3\sigma_2^2\le 0,\\ -\sigma_2^1+2\sigma_2^2\le 0,\\ 2\sigma_2^1-5\sigma_2^2\le 0,\\ \sigma_2^1-6\sigma_2^2\le 0,\\ \sigma_2^1+\sigma_2^2\le 1.\end{cases}\quad \sigma_2^\intercal=(0,0).$$

Here, $\sigma_1^\intercal=\sigma_2^\intercal=0$, which means that, x^5 is feasible for both scenarios. Following this, we perform the optimality test for x^5 by solving the following two problems.

Scenario 1 Scenario 2

$$\pi(\xi^1)\begin{cases}\max\ -6\pi_1^1+11\pi_1^2,\\ -2\pi_1^1+3\pi_1^2\le 1,\\ -\pi_1^1+2\pi_1^2\le 3,\\ 2\pi_1^1-5\pi_1^2\le 6,\\ \pi_1^1-6\pi_1^2\le 2.\end{cases}\quad \pi_1^\intercal=(7,5),\qquad \pi(\xi^2)\begin{cases}\max\ -14\pi_1^1+4\pi_1^2,\\ -2\pi_1^1+3\pi_1^2\le 5,\\ -\pi_1^1+2\pi_1^2\le 3,\\ 2\pi_1^1-5\pi_1^2\le 2,\\ \pi_1^1-6\pi_1^2\le 1.\end{cases}\quad \pi_2^\intercal=\left(-\tfrac{11}{3},-\tfrac{7}{9}\right).$$

Once the previous two problems have been solved, the process is followed by first calculating the value of the expected recourse function value $Q(x)$ by:

$Q\left(x,\xi^1\right)=\pi_1^\intercal\left[h\left(\xi^1\right)-T\left(\xi^1\right)x^3\right]=13,$
$Q\left(x,\xi^2\right)=\pi_2^\intercal\left[h\left(\xi^2\right)-T\left(\xi^2\right)x^3\right]=\frac{434}{9},$
$Q\left(x\right)=\frac{2}{3}Q\left(x,\xi^1\right)+\frac{1}{3}Q\left(x,\xi^2\right)=\frac{668}{27}.$

Note that: $Q(x)>\theta=-\infty$. Based on this, we construct the optimal cut as outlined below. We then add it to the current problem (P^8).

$$\theta\ge\frac{689}{27}-\frac{32}{3}x_1+\frac{295}{27}x_2.$$

After adding the optimality cut, we proceed to solve the new problem (P^{10}).

$$(P^{10})\begin{cases}\min\ \phi=-x_1+4x_2,\\ \text{s.t.,}\\ x\in\mathcal{S}_8'=\{x\in\mathcal{S}_8|\ \theta\ge\frac{689}{27}-\frac{32}{3}x_1+\frac{295}{27}x_2\}.\end{cases}$$

After solving problem (P^{10}), we obtain the same optimal solution $x^5=(3,3)$. As we have seen previously, this obtained solution is the feasible and optimal solution for both scenarios with the penalty value

Table 4.1: Optimal simplex table at node 8

B	Rhs	x_9	x_{10}
x_8	$\frac{1}{3}$	$\frac{-1}{3}$	$\frac{-5}{3}$
x_3	12	-1	-4
x_4	6	-1	3
x_5	1	-2	1
x_6	1	-1	0
x_2	3	0	-1
x_7	2	0	-1
θ	$\frac{668}{27}$	$\frac{32}{27}$	$\frac{295}{27}$
x_1	3	1	0
$\widetilde{f_1}$	-9	5	2
$\widetilde{f_2}$	3	-4	-3
$\widetilde{f_3}$	-9	-1	-4
$\bar{d}_1$	9	1	4
$\bar{d}_2$	9	-2	1

$\theta = \frac{668}{27}$, see table 4.1.

The algorithm now proceeds to another essential step, namely the efficiency test. This test is performed to evaluate the efficiency of the solution x^5 for both bi-objective and multi-objective problems.

Step 5: Efficiency tests.

We start first by testing the efficiency of x^5 for the bi-objective problem by solving the following problem:

$$(EK^1(x^5))\begin{cases} \max V = v_1 + v_2, \\ \text{s.t.}, \\ x \in \mathcal{S}_8, \\ \theta \geq \frac{689}{27} + \frac{32}{3}x_1 - \frac{295}{27}x_2, \\ -x_1 + 4x_2 + v_1 = 9, \\ 2x_1 + x_2 + v_2 = 9, \\ x_1, x_2 \in \mathbb{N},\ \theta, v_1, v_2 \in \mathbb{R}. \end{cases}$$

Note that $V = 0$, this means that x^5 is efficient for the bi-objective problem. Following this, we test its efficiency for the multi-objective problem by solving the following problem:

$$(EK^2(x^5))\begin{cases} \max \Psi = \psi_1 + \psi_2 + \psi_3, \\ \text{s.t.}, \\ x \in \mathcal{S}_8, \\ \theta \geq \frac{689}{27} + \frac{32}{3}x_1 - \frac{295}{27}x_2, \\ -5x_1 + 2x_2 + \psi_1 = -9, \\ 4x_1 - 3x_2 + \psi_2 = 3, \\ x_1 - 4x_2 + \psi_3 = -9, \\ x_1, x_2 \in \mathbb{N},\ \theta, \psi_1, \psi_2, \psi_3 \in \mathbb{R}. \end{cases}$$

Here, $\Psi = 0$, that means that $x^5 = (3,3)$ is efficient for the multi-objective problem. Furthermore, we notice that x^5 is efficient for both bi-ovjective and multi-objective problems. Thus, we update the efficient set $\mathcal{X}_B$ by adding the solution x^5, $\mathcal{X}_B = \{(3,3)\}$ and the algorithm continues the process through the last step.

Step 6: The efficient cuts.
We start first by constructing the two sets $\mathcal{H}_1$ and $\mathcal{H'}_1$ by using the decreasing direction of each criterion $\widetilde{f}_{i\in\{1,\dots,3\}}$ for the multi-objective problem, and the decreasing direction of each criterion $\widetilde{\phi}_{i\in\{1,2\}}$ for the bi-objective problem, respectively.

From Table 4.1, $\mathcal{H}_1 = \{9,10\} \neq \emptyset$ and $\mathcal{H'}_1 = \{9\} \neq \emptyset$.

After adding the cuts $x_9 + x_{10} \geq 1$ and $x_9 \geq 1$ to the current problem (P^{10}), the algorithm continues the search process for other efficient solutions by solving the following new problem at node 10.

$$(P^{11}) \begin{cases} \min\ \phi = -x_1 + 4x_2, \\ \text{s.t.,} \\ x \in \mathcal{S}_{10} = \{x \in \mathcal{S}'_8 \mid x_9 + x_{10} \geq 1,\ x_9 \geq 1\}. \end{cases}$$

From the table 4.2, the integer solution obtained is $x^6 = (2,3)$. In this step, we pass directly to the feasibility and optimality test.

Table 4.2: Optimal simplex table for node 10

B	Rhs	x_{10}	x_{13}
x_8	$\frac{2}{3}$	$\frac{-4}{3}$	$\frac{-1}{3}$
x_3	13	-3	-1
x_4	7	4	-1
θ	$\frac{380}{27}$	$\frac{7}{27}$	$\frac{32}{3}$
x_6	2	1	-1
x_2	3	-1	0
x_7	2	-1	0
x_9	1	-1	-1
x_1	2	-1	1
x_5	3	3	-1
x_{12}	$\frac{-1}{45}$	-1	1
$\widetilde{f_1}$	-4	-3	5
$\widetilde{f_2}$	-1	1	-4
$\widetilde{f_3}$	-10	-3	-1
$\bar{d}_1$	10	3	1
$\bar{d}_2$	7	3	-2

Step 3 Feasibility and Optimality tests

We test the feasibility of x^6 by solving the following two problems.

Scenario 1

$$\sigma(\xi^1)\begin{cases} \max\ -5\sigma_1^1+8\sigma_1^2 \\ -2\sigma_1^1+3\sigma_1^2\le 0 \\ -\sigma_1^1+2\sigma_1^2\le 0 \\ 2\sigma_1^1-5\sigma_1^2\le 0 \\ \sigma_1^1-6\sigma_1^2\le 0 \\ \sigma_1^1+\sigma_1^2\le 1 \end{cases} \quad \sigma_1^t=(0,0)$$

Scenario 2

$$\sigma(\xi^2)\begin{cases} \max\ -9\sigma_2^1+\sigma_2^2 \\ -2\sigma_2^1+3\sigma_2^2\le 0 \\ -\sigma_2^1+2\sigma_2^2\le 0 \\ 2\sigma_2^1-5\sigma_2^2\le 0 \\ \sigma_2^1-6\sigma_2^2\le 0 \\ \sigma_2^1+\sigma_2^2\le 1 \end{cases} \quad \sigma_2^t=(0,0)$$

$\sigma_1^t=\sigma_2^t=0$, that means x^6 is feasible for both scenarios. After that, we test the optimality of x^6 by solving the following two problems.

Scenario 1

$$\pi(\xi^1)\begin{cases} \max\ -5\pi_1^1+8\pi_1^2 \\ -2\pi_1^1+3\pi_1^2\le 1 \\ -\pi_1^1+2\pi_1^2\le 3 \\ 2\pi_1^1-5\pi_1^2\le 6 \\ \pi_1^1-6\pi_1^2\le 2 \end{cases} \quad \pi_1^t=(7,5)$$

Scenario 2

$$\pi(\xi^2)\begin{cases} \max\ -9\pi_1^1+\pi_1^2 \\ -2\pi_1^1+3\pi_1^2\le 5 \\ -\pi_1^1+2\pi_1^2\le 3 \\ 2\pi_1^1-5\pi_1^2\le 2 \\ \pi_1^1-6\pi_1^2\le 1 \end{cases} \quad \pi_2^t=\left(-\tfrac{11}{3},-\tfrac{7}{9}\right)$$

In this situation, $Q(x)=\frac{380}{27}$, and we note that $Q(x)=\theta$ (see table 4.2), this means that $x^6=(2,3)$ is optimal for both scénarios with $\theta=\frac{380}{27}$. We proceed to the next step of the algorithm.

Step 5 Efficiency tests

We start by testing the efficiency of x^6 for the bi-objective problem by solving the following problem:

$$(EK^1(x^6))\begin{cases} \max V=v_1+v_2 \\ \text{s.t}\quad x\in\mathcal{S}_8 \\ \theta\ge\frac{689}{27}+\frac{32}{3}x_1-\frac{295}{27}x_2 \\ -x_1+4x_2+v_1=10 \\ 2x_1+x_2+v_2=7 \\ x_1,x_2\in\mathbb{N},\ \theta,v_1,v_2\in\mathbb{R} \end{cases}$$

We found = 0, Then x^6 is efficient for the bi-objective problem. After that, we test its efficiency for the

multi-objective problem by solving the following problem:

$$(EK^2(x^6))\begin{cases} \max \Psi = \psi_1 + \psi_2 + \psi_3 \\ \text{s.t} \quad x \in \mathcal{S}_8 \\ \theta \geq \frac{689}{27} + \frac{32}{3}x_1 - \frac{295}{27}x_2 \\ -5x_1 + 2x_2 + \psi_1 = -4 \\ 4x_1 - 3x_2 + \psi_2 = -1 \\ x_1 - 4x_2 + \psi_3 = -10 \\ x_1, x_2 \in \mathbb{N},\ \theta, \psi_1, \psi_2, \psi_3 \in \mathbb{R} \end{cases}$$

Here, $\Psi = 0$, this means that $x^6 = (2, 3)$ is efficient for both problems. Before proceeding to the next step, this solution will be added to the set $\mathcal{X}_B = \{(3, 3), (2, 3)\}$.

Step 6 The efficient cuts

In this step, we construct two new sets $\mathcal{H}_2 = \{10, 14\} \neq \emptyset$ and $\mathcal{H'}_2 = \{14\} \neq \emptyset$. See Table 4.2. After that, we add the new constraints $x_{10} + x_{14} \geq 1$ and $x_{14} \geq 1$ to the current problem. Once again, the algorithm continues the search process of finding other efficient solutions by solving the new problem at node 11.

$$(P^{12})\begin{cases} \min\ \phi = -x_1 + 4x_2 \\ \text{s.t} \\ x \in \mathcal{S}_{11} = \{x \in \mathcal{S}_{10} \mid x_{10} + x_{14} \geq 1,\ x_{14} \geq 1\} \end{cases}$$

The new integer optimal solution is $x^7 = (1, 3)$.

Step 3 Feasibility and Optimality tests

We test the feasibility of x^7 by solving the following problems.

Scenario 1

$$\sigma(\xi^1)\begin{cases} \max\ -4\sigma_1^1 + 5\sigma_1^2 \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0 \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0 \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0 \\ \sigma_1^1 - 6\sigma_1^2 \leq 0 \\ \sigma_1^1 + \sigma_1^2 \leq 1 \end{cases} \quad \sigma_1^t = (0, 0)$$

Scenario 2

$$\sigma(\xi^2)\begin{cases} \max\ -4\sigma_2^1 - 2\sigma_2^2 \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0 \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0 \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0 \\ \sigma_2^1 - 6\sigma_2^2 \leq 0 \\ \sigma_2^1 + \sigma_2^2 \leq 1 \end{cases} \quad \sigma_2^t = (0, 0)$$

Note that, $\sigma_1^t = \sigma_2^t = 0$, x^7 is feasible for both scenarios. We test the optimality of x^7 by solving the following problems.

Scenario 1

$$\pi(\xi^1)\begin{cases} \max\ -4\pi_1^1+5\pi_1^2 \\ -2\pi_1^1+3\pi_1^2\le 1 \\ -\pi_1^1+2\pi_1^2\le 3 \quad \pi_1^t=\left(-\frac{4}{3},-\frac{5}{9}\right) \\ 2\pi_1^1-5\pi_1^2\le 6 \\ \pi_1^1-6\pi_1^2\le 2 \end{cases}$$

Scenario 2

$$\pi(\xi^2)\begin{cases} \max\ -4\pi_1^1-2\pi_1^2 \\ -2\pi_1^1+3\pi_1^2\le 5 \\ -\pi_1^1+2\pi_1^2\le 3 \quad \pi_2^t=\left(-\frac{11}{3},-\frac{7}{9}\right) \\ 2\pi_1^1-5\pi_1^2\le 2 \\ \pi_1^1-6\pi_1^2\le 1 \end{cases}$$

After calculating $Q(x)=\frac{64}{9}$ we found that $Q(x)>(\theta=\frac{92}{27})$. In this situation, we construct the following optimality cut and add it to the current problem.

$$\theta\ge -\frac{29}{3}-\frac{46}{9}x_1-\frac{35}{9}x_2$$

After solving the new following problem, we found the same previous solution $x^7=(1,3)$ with $\theta=\frac{64}{9}$. See table 4.3

$$(P^{13})\begin{cases} \min\ \phi=-x_1+4x_2 \\ \text{s.t} \\ x\in\mathcal{S}_{11}^{'}=\{x\in\mathcal{S}_{11},\mid \theta\ge -\frac{29}{3}-\frac{46}{9}x_1-\frac{35}{9}x_2,\ x\in\mathbb{N}^n\} \end{cases}$$

Table 4.3: Optimal simplex table after introducing optimality cut at node 11

B	Rhs	x_{10}	x_{16}
x_8	1	-1	$\frac{-1}{3}$
x_3	14	-2	-1
x_4	8	5	-1
x_5	5	5	-2
x_6	3	2	-1
x_2	3	-1	0
x_7	2	-1	0
θ	$\frac{64}{9}$	$\frac{-127}{9}$	$\frac{46}{9}$
x_{12}	$\frac{100}{27}$	$\frac{-100}{27}$	$\frac{-50}{9}$
x_9	2	2	-1
x_{14}	1	1	-1
x_{13}	1	2	-1
x_{15}	0	1	-1
x_1	1	-2	1
$\widetilde{f_1}$	1	-8	5
$\widetilde{f_2}$	-5	5	-4
$\widetilde{f_3}$	-11	-2	-1
$\bar{d}_1$	11	2	1
$\bar{d}_2$	5	5	-2

Step 5 Efficiency tests

We test the efficiency of x^7 for the bi-objective problem by solving the following problem:

$$(EK^1(x^7))\begin{cases} \max V = v_1 + v_2 \\ \text{s.t} \quad x \in \mathcal{S}_8 \\ \theta \geq -\frac{29}{3} - \frac{46}{9}x_1 - \frac{35}{9}x_2 \\ -x_1 + 4x_2 + v_1 = 11 \\ 2x_1 + x_2 + v_2 = 5 \\ x_1, x_2 \in \mathbb{N},\ \theta, v_1, v_2 \in \mathbb{R} \end{cases}$$

Here = 0, which means that x^7 is efficient for the bi-objective problem. Now we test its efficiency for the multi-objective problem by solving the following problem:

$$(EK^2(x^7))\begin{cases} \max \Psi = \psi_1 + \psi_2 + \psi_3 \\ \text{s.t} \quad x \in \mathcal{S}_8 \\ \theta \geq -\frac{29}{3} - \frac{46}{9}x_1 - \frac{35}{9}x_2 \\ -5x_1 + 2x_2 + \psi_1 = 1 \\ 4x_1 - 3x_2 + \psi_2 = -5 \\ x_1 - 4x_2 + \psi_3 = -11 \\ x_1, x_2 \in \mathbb{N},\ \theta, \psi_1, \psi_2, \psi_3 \in \mathbb{R} \end{cases}$$

We found $\Psi = 10 \neq 0$, which means x^7 is not efficient for the multi-objective problem. This solution is not added to the set $\mathcal{X}_B$. However, we continue the search process for other efficient solutions.

Step 6 The efficient cuts

In this situation, $\mathcal{H}_3 = \{10, 16\} \neq \emptyset$ and $\mathcal{H'}_3 = \{16\} \neq \emptyset$, see Table 4.3. We construct then the new two constraints $x_{10} + x_{16} \geq 1$ and $x_{16} \geq 1$ and we add it to the current problem.

$$(P^{14})\begin{cases} \min\ \phi = -x_1 + 4x_2 \\ \text{s.t} \\ x \in \mathcal{S}_{12} = \{x \in \mathcal{S}'_{11} \mid x_{10} + x_{16} \geq 1,\ x_{16} \geq 1\} \end{cases}$$

After resolving the new problem, we found a new optimal solution $x^8 = (0, 3)$ which is an integer.

Step 3 Feasibility and Optimality tests

We resolve the next following problems to test the feasibility of x^8

Scenario 1 Scenario 2

$$\sigma(\xi^1)\begin{cases}\max\ -3\sigma_1^1+2\sigma_1^2\\ -2\sigma_1^1+3\sigma_1^2\le 0\\ -\sigma_1^1+2\sigma_1^2\le 0\\ 2\sigma_1^1-5\sigma_1^2\le 0\\ \sigma_1^1-6\sigma_1^2\le 0\\ \sigma_1^1+\sigma_1^2\le 1\end{cases}\quad \sigma_1^t=(0,0) \qquad \sigma(\xi^2)\begin{cases}\max\ \sigma_2^1-5\sigma_2^2\\ -2\sigma_2^1+3\sigma_2^2\le 0\\ -\sigma_2^1+2\sigma_2^2\le 0\\ 2\sigma_2^1-5\sigma_2^2\le 0\\ \sigma_2^1-6\sigma_2^2\le 0\\ \sigma_2^1+\sigma_2^2\le 1\end{cases}\quad \sigma_2^t=(0,0)$$

We notice that x^8 is feasible for both scenarios. We proceed now with the optimality test for x^8 by solving the following problems.

Scenario 1 Scenario 2

$$\pi(\xi^1)\begin{cases}\max\ -3\pi_1^1+2\pi_1^2\\ -2\pi_1^1+3\pi_1^2\le 1\\ -\pi_1^1+2\pi_1^2\le 3\\ 2\pi_1^1-5\pi_1^2\le 6\\ \pi_1^1-6\pi_1^2\le 2\end{cases}\quad \pi_1^t=\left(-\tfrac{4}{3},-\tfrac{5}{9}\right) \qquad \pi(\xi^2)\begin{cases}\max\ \pi_1^1-5\pi_1^2\\ -2\pi_1^1+3\pi_1^2\le 5\\ -\pi_1^1+2\pi_1^2\le 3\\ 2\pi_1^1-5\pi_1^2\le 2\\ \pi_1^1-6\pi_1^2\le 1\end{cases}\quad \pi_2^t=(1,0)$$

In this situation, $Q(x)=\frac{61}{27}>\theta=2$, which means that we will add the following optimality cut to the current problem.

$$\theta\ge -\frac{86}{27}+\frac{17}{9}x_1-\frac{49}{27}x_2$$

$$(P^{14})\begin{cases}\min\ \phi=-x_1+4x_2\\ \text{s.t}\\ x\in\mathcal{S}_{12}^{'}=\{x\in\mathcal{S}_{12},\mid \theta\ge -\frac{86}{27}+\frac{17}{9}x_1-\frac{49}{27}x_2,\ x\in\mathbb{N}^n\}\end{cases}$$

We solve the new problem (P^{14}) and we found the same previous solution $x^8=(0,3)$ with $\theta=\frac{61}{27}$, see table 4.4. Following this, we pass through the efficiency tests step.

Step 5 Efficiency tests

The efficiency of x^8 for the bi-objective problem is tested by solving the following problem:

$$(EK^1(x^8))\begin{cases}\max V=v_1+v_2\\ \text{s.t}\quad x\in\mathcal{S}_8\\ \theta\ge -\frac{86}{27}+\frac{17}{9}x_1-\frac{49}{27}x_2\\ -x_1+4x_2+v_1=12\\ 2x_1+x_2+v_2=3\\ x_1,x_2\in\mathbb{N},\ \theta,v_1,v_2\in\mathbb{R}\end{cases}$$

Table 4.4: Optimal simplex table after introducing optimality cut at node 12

B	Rhs	x_1	x_{18}
x_8	$\frac{4}{3}$	$\frac{-1}{2}$	$\frac{-5}{6}$
x_3	15	-1	-1
x_4	9	$\frac{5}{2}$	$\frac{3}{2}$
x_5	7	$\frac{5}{2}$	$\frac{1}{2}$
x_6	4	-1	0
x_2	3	$\frac{-1}{2}$	$\frac{-1}{2}$
x_7	2	$\frac{-1}{2}$	$\frac{-1}{2}$
x_{15}	1	$\frac{-1}{2}$	$\frac{-1}{2}$
x_{12}	$\frac{257}{27}$	$\frac{167}{27}$	$\frac{-172}{27}$
x_9	3	1	0
x_{14}	2	$\frac{1}{2}$	$\frac{-1}{2}$
x_{13}	2	1	0
x_{10}	$\frac{-1}{72}$	$\frac{-1}{2}$	$\frac{-1}{2}$
x_{17}	$\frac{7}{27}$	$\frac{217}{27}$	$\frac{28}{27}$
x_{19}	$\frac{1}{10}$	$\frac{-1}{2}$	$\frac{-3}{2}$
x_{16}	1	0	-1
θ	$\frac{61}{27}$	$\frac{53}{54}$	$\frac{-49}{54}$
$\widetilde{f_1}$	6	-4	1
$\widetilde{f_2}$	-9	$\frac{5}{2}$	$-\frac{3}{2}$
$\widetilde{f_3}$	-12	-1	-2
$\bar{d}_1$	12	1	2
$\bar{d}_2$	3	$\frac{5}{2}$	$\frac{1}{2}$

x^8 is efficient for the bi-objective problem. After that, We test its efficiency for the multi-objective problem by solving the following problem:

$$(EK^2(x^8))\begin{cases} \max \Psi = \psi_1 + \psi_2 + \psi_3 \\ \text{s.t} \quad x \in \mathcal{S}_8 \\ \theta \geq -\frac{86}{27} + \frac{17}{9}x_1 - \frac{49}{27}x_2 \\ -5x_1 + 2x_2 + \psi_1 = 6 \\ 4x_1 - 3x_2 + \psi_2 = -9 \\ x_1 - 4x_2 + \psi_3 = -12 \\ x_1, x_2 \in \mathbb{N},\ \theta, \psi_1, \psi_2, \psi_3 \in \mathbb{R} \end{cases}$$

We found that $\Psi = 10 \neq 0$, which means x^8 is not efficient for the multi-objective problem.

Step 6 The efficient cuts

In this situation, $\mathcal{H}_4 = \{1, 18\} \neq \emptyset$ and $\mathcal{H'}_4 = \emptyset$, see table 4.4. Note that $\mathcal{H'}_4 = \emptyset$ which means that this node is fathomed. Moreover, the algorithm continues the search process and tries to find other efficient solutions if it exists by visiting other nodes. At this step, we solve the program at the node 4.

$$(P^4) \begin{cases} \min \ \phi = -x_1 + 4x_2 \\ \text{s.t} \ \ x \in \mathcal{S}_4 \end{cases}$$

The obtained integer solution is $x^9 = (3, 0)$.

Step 3 Feasibility and Optimality tests

The feasibility of x^9 is tested by resolving the following problems.

Scenario 1 Scenario 2

$$\sigma(\xi^1) \begin{cases} \max \quad 14\sigma_1^2 \\ -2\sigma_1^1 + 3\sigma_1^2 \leq 0 \\ -\sigma_1^1 + 2\sigma_1^2 \leq 0 \\ 2\sigma_1^1 - 5\sigma_1^2 \leq 0 \\ \sigma_1^1 - 6\sigma_1^2 \leq 0 \\ \sigma_1^1 + \sigma_1^2 \leq 1 \end{cases} \quad \sigma_1^t = \left(\tfrac{2}{3}, \tfrac{1}{3}\right) \qquad \sigma(\xi^2) \begin{cases} \max \ -11\sigma_2^1 + 10\sigma_2^2 \\ -2\sigma_2^1 + 3\sigma_2^2 \leq 0 \\ -\sigma_2^1 + 2\sigma_2^2 \leq 0 \\ 2\sigma_2^1 - 5\sigma_2^2 \leq 0 \\ \sigma_2^1 - 6\sigma_2^2 \leq 0 \\ \sigma_2^1 + \sigma_2^2 \leq 1 \end{cases} \quad \sigma_2^t = (0, 0)$$

Note that x^9 is not feasible for the first scenario ξ^1. We add the following feasibility cut to the current program.

$$-\frac{1}{3}x_1 + \frac{5}{3}x_2 \geq \frac{11}{3}$$

After solving the new problem in the new node. We do not find any feasible solution in this node 13. This means that this node 13 is **fathomed**.

Final result:

The final efficient set found is $\mathcal{X}_B = \{(3, 3), (2, 3)\}$.
Where:

- $\mathcal{X}_C = \{(3, 3), (2, 3), (1, 3), (0, 3)\}$, the efficient solutions in terms of d^1 and d^2.
- $\mathcal{X}_E = \{(3, 3), (2, 3), (3, 4), (2, 4), (1, 4), (0, 4), (2, 5), (1, 5), (0, 5), (0, 6)\}$, the efficient set of the deterministic equivalent problem (2.24).

4.4.1 The search tree

The search tree presented in figure (4.1) summarizes and illustrates the different steps and states of the nodes throughout the algorithm process in the previous example 4.4.

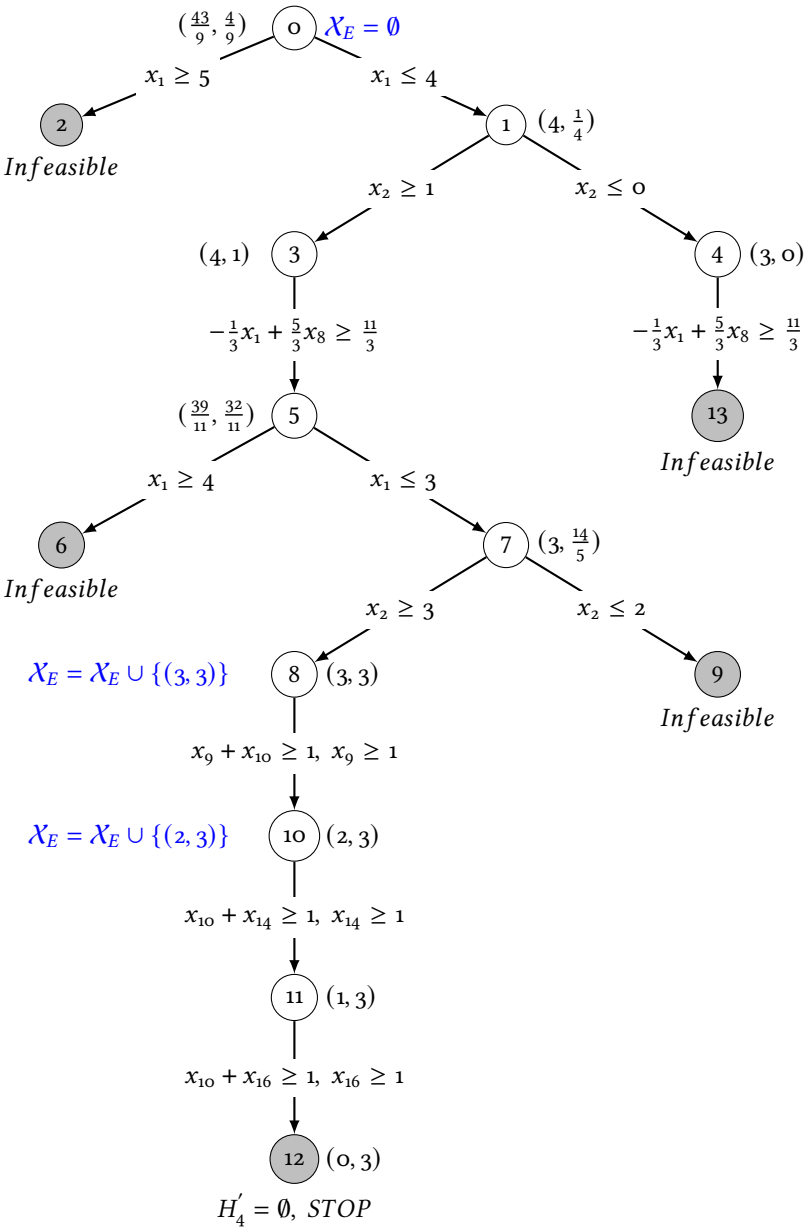


Figure 4.1: Search tree of the example

4.5 Computational Results

To the best of our knowledge, this particular problem has not been studied in existing literature. Indeed, according to the available literature, no benchmark instance has been proposed to test the performance of this type of problem. For this reason, we thought of creating randomly generated instances to test the performance of our method for solving this problem. Moreover, the algorithm has been implemented in the MATLAB 2019 environment without making use of any solver and tested over the randomly generated instances, as for the efficiency and optimality tests we have used the linprog solver. In addition, all proposed solution procedures of the computational results section are performed on a computer with Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz processor and 4 GB RAM.
Regarding deterministic data (constraint coefficients) they are uncorrelated and uniformly distributed. Each component of the vector b, the entries of the matrix A, and the coefficients of the objective functions C and d were drawn randomly from discrete uniform distributions in the following ranges $[100, 200]$, $[1, 100]$, $[-100, 100]$ and $[-100, 100]$.
The stochastic data are generated in the same way as the deterministic ones. For each component of the recourse matrix W, the penalties for constraint violations q, the matrice T and the vector h are in the intervals $[-50, 50]$, $[1, 50]$, $[-50, 50]$ and $[-50, 50]$, respectively. The probability for each scenario is randomly generated with the sum of probabilities equal to 1.
We have worked on 44 distinct groups of instances of the (n, m, k, R) type, where n represents the number of variables, m denotes the number of constraints, k indicates the number of objectives, and R signifies the number of scenarios, with $R = 2, 3, 5, 10$. Additionally, for each group, we generated five instances randomly, resulting in a total of 220 instances. Our method successfully solved these instances, and the corresponding results are presented in the following Table 4.5. Note that: The generation of random instances does not guarantee the feasibility of the instance. For this study, we have kept only the feasible instances.
Table Contents: Table 4.5 shows the results achieved by our method on five instances for each group.

- Column 1 represents the number of existing scenarios R.
- Column 2 presents the size of each instance ($n \times m \times k$), where n is the number of variables, m is the number of constraints and k is the number of objectives.
- Column 3 displays the minimum (Min), the average (Mean), and the maximum (Max) of the CPU time (in seconds).
- Column 4 reports the minimum (Min), the average (Mean), and the maximum (Max) of the number of nodes required (Nbr of Nodes) to get the final set $\mathcal{X}_B$.
- The last column ρ represents the average (Mean) of $|\mathcal{X}_B/\mathcal{X}_E|$.

Table 4.5: The behavior of our method on medium and large scale instances in different scenarios and based on different criteria

		CPU (Sec)			*Nbr of Nodes*			ρ
R	$n \times m \times k$	*Min*	*Mean*	*Max*	*Min*	*Mean*	*Max*	*Mean*
2	$5 \times 3 \times 3$	0.41	2.47	6.43	10	42	88	0.40
	$10 \times 5 \times 3$	2.08	4.49	6.80	40	79	116	0.39
	$20 \times 10 \times 5$	5.39	8.51	12.58	134	197.20	338	0.09
	$30 \times 15 \times 5$	21.15	30.11	39.15	322	490	652	0.10
	$40 \times 20 \times 5$	55.80	82.15	164.64	608	721.20	954	0.10
	$50 \times 25 \times 10$	76.74	113.55	135.27	754	918	1044	0.27
	$60 \times 30 \times 10$	138.55	181.98	243.79	916	1112.80	1478	0.12
	$70 \times 35 \times 10$	346.84	405.40	489.74	1580	1822.40	2344	0.09
	$80 \times 40 \times 20$	513.05	565.49	649.62	1446	1770	2206	0.11
	$90 \times 45 \times 20$	783.20	1070.44	1395.43	2546	2667.60	2978	0.10
	$100 \times 50 \times 20$	1003.98	1407.96	1897.15	2288	2816	3438	0.12
Glob	*Avrg*$(R = 2)$	267.93	352.05	458.24	967.64	930.49	1421.45	0.17
3	$5 \times 3 \times 3$	0.97	3.51	11.65	8	33.20	98	0.57
	$10 \times 5 \times 3$	1.06	4.23	7.50	18	66.40	118	0.37
	$20 \times 10 \times 5$	3.54	11.78	18.64	106	190.80	256	0.29
	$30 \times 15 \times 5$	18.64	23.62	26.31	322	398.80	454	0.13
	$40 \times 20 \times 5$	38.28	60.72	88.62	576	693.20	904	0.30
	$50 \times 25 \times 10$	71.32	104.45	187.82	538	862.40	1424	0.19
	$60 \times 30 \times 10$	110.91	191.95	358.30	832	1111.60	1634	0.06
	$70 \times 35 \times 10$	205.65	340.08	492.87	864	1425.20	1798	0.07
	$80 \times 40 \times 20$	349.53	612.98	876.32	1180	2019.20	2844	0.08
	$90 \times 45 \times 20$	689.63	954.20	1466.57	1706	2415.20	3040	0.06
	$100 \times 50 \times 20$	273.65	1402.61	2272.41	746	2535.60	4096	0.12
Glob	*Avrg*$(R = 3)$	160.24	337.29	527.91	337.29	626.91	1515.09	0.20
5	$5 \times 3 \times 3$	1.84	3.94	6.70	10	26.40	46	0.29
	$10 \times 5 \times 3$	5.65	7.89	14.33	58	84.40	154	0.40
	$20 \times 10 \times 5$	7.60	11.75	20.91	152	183.20	238	0.27
	$30 \times 15 \times 5$	22.31	27.64	33.54	302	361.20	450	0.13
	$40 \times 20 \times 5$	36.35	72.61	118.72	384	648.80	992	0.08
	$50 \times 25 \times 10$	101.96	143.70	213.36	826	1036.80	1434	0.08
	$60 \times 30 \times 10$	154.29	267.86	360.80	1048	1419.60	1816	0.07
	$70 \times 35 \times 10$	244.93	350.09	471.33	1324	1523.20	1804	0.05
	$80 \times 40 \times 20$	545.48	812.29	993.73	1860	2357.20	2882	0.04
	$90 \times 45 \times 20$	913.20	1033.18	1240.23	1944	2503.60	2966	0.04
	$100 \times 50 \times 20$	873.47	1282.77	1674.22	1744	2548.80	3398	0.07
Glob	*Avrg*$(R = 5)$	264.28	364.88	467.98	877.45	1153.93	1470.91	0.14
10	$5 \times 3 \times 3$	1.40	5.12	8.36	8	21.20	34	0.33
	$10 \times 5 \times 3$	3.44	9.40	16.91	18	56	104	0.27
	$20 \times 10 \times 5$	8.59	23.96	44.38	68	231.20	434	0.07
	$30 \times 15 \times 5$	18.74	41.13	63.64	172	419.60	552	0.11
	$40 \times 20 \times 5$	67.11	86.19	127.58	598	762.80	994	0.16
	$50 \times 25 \times 10$	68.85	135.97	170.81	570	1040	1288	0.07
	$60 \times 30 \times 10$	167.50	261.94	304.53	904	1348	1844	0.07
	$70 \times 35 \times 10$	290.50	373.66	485.72	1404	1499.20	1752	0.07
	$80 \times 40 \times 20$	427.21	596.96	759.83	1462	1771.20	2274	0.05
	$90 \times 45 \times 20$	621.69	1032.79	1626.56	1934	2383.20	3042	0.06
	$100 \times 50 \times 20$	477.19	1263.33	1814.29	1180	2535.60	3724	0.07
Glob	*Avrg*$(R = 10)$	195.66	348.22	492.96	756.18	1097.09	1458.36	0.12

The elements and cardinality of $\mathcal{X}_E$ are calculated using the algorithm presented in [Ouaïl et al., 2017]. The elements and cardinality of $\mathcal{X}_B$ are calculated by using the present method. In addition, we have reported a global average (Glob Avrg) in the last line for each R scenario case, which represents the total mean for all instances studied in that R case.

The results from Table 4.5 show that average CPU time generally increases with the size of the instance and the number of scenarios. For example, for $R = 2$ the average CPU time increases from 2.47 seconds for the instance $(5 \times 3 \times 3)$ to 1407.96 seconds for the instance $(100 \times 50 \times 20)$. Similarly, for $R = 2$, the average CPU time for the instance $(20 \times 10 \times 5)$ is 8.51 seconds, while for $R = 10$ it rises to 23.96 seconds. However, this trend is not always consistent. For example, the instance $(90 \times 45 \times 20)$ shows a decrease in average CPU time from 1070.44 seconds for $R = 2$ to 954.20 seconds for $R = 3$. This is due to the fact that the average CPU time is mainly influenced by the number of visited efficient solutions. In the same instance $(90 \times 45 \times 20)$, the average number of nodes is higher for $R = 2$ at 2667.60 compared to 2415.20 for $R = 3$, which suggests that more efficient solutions were visited for $R = 2$. Thus, the average CPU time is mainly related to the number of efficient solutions explored, and it is represented in figure 4.2.

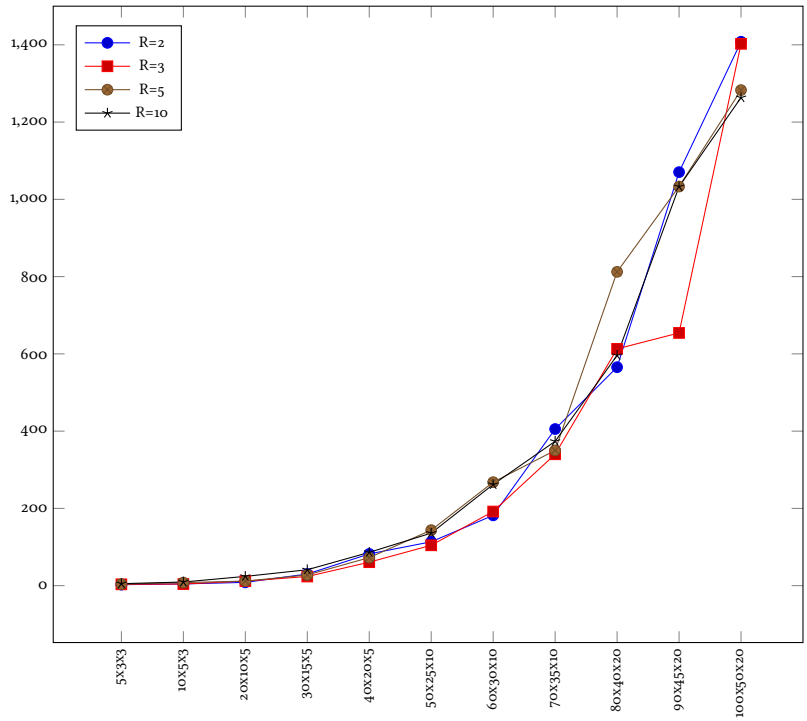


Figure 4.2: Illustration of the CPU time (in seconds) for all studied instances.

Figure 4.3 presents a graphical representation of the average number of nodes required to obtain the optimal set $\mathcal{X}_B$ for all the instances analyzed.

Concerning the last column ρ, we have presented the average ration that exists between the number of elements of the set $\mathcal{X}_B$ calculated by our algorithm and the number of elements of the efficient set $\mathcal{X}_E$

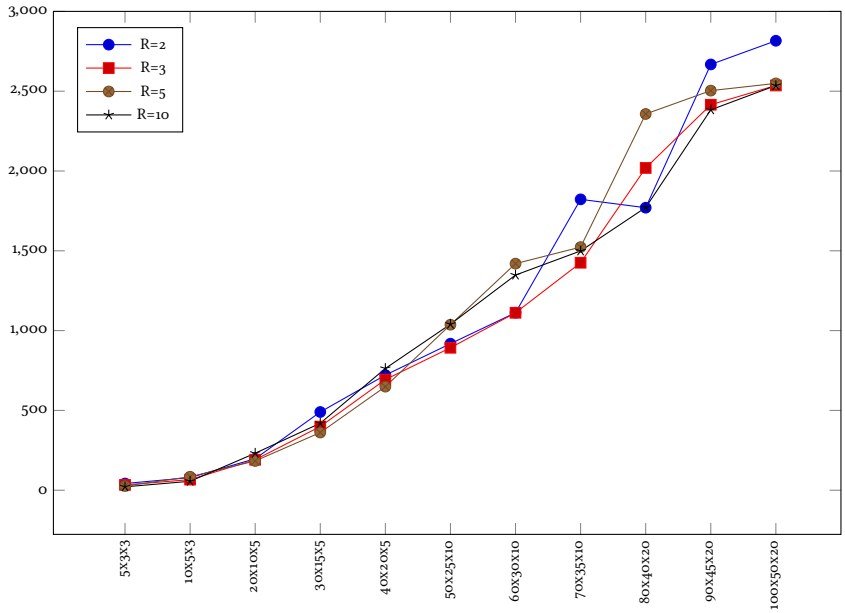


Figure 4.3: Illustration of the mean of the number of nodes required for all studied instances.

of the MOILP problem gotten by the method presented in [Ouaïl et al., 2017]. Where, ρ indicates the average of $|\mathcal{X}_B/\mathcal{X}_E|$. We see that the minimal value of ρ does not exceed 0.04 for all the instances and that the maximal value equal a 0.57 for all the instances. An illustration of the value ρ is presented by the following graphs 4.4.

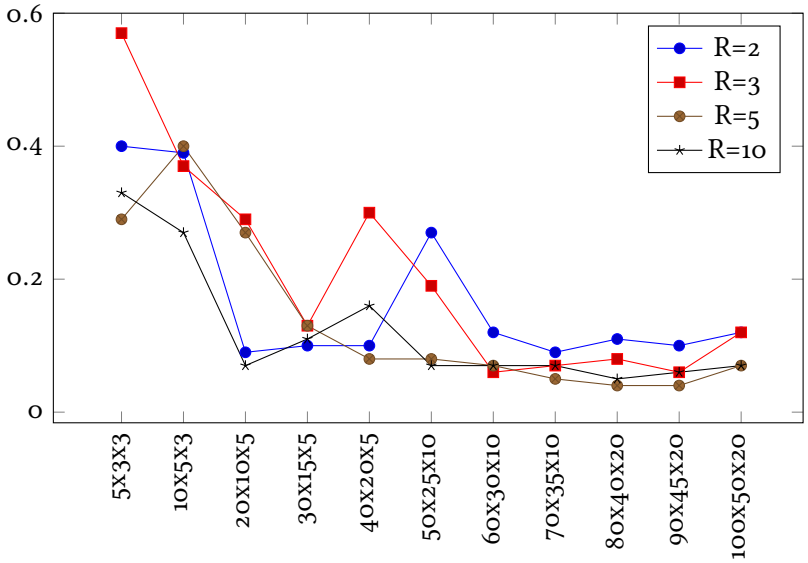


Figure 4.4: Illustration of the value of ρ.

4.6 Conclusion

This chapter presents an exact method for dealing with a new type of stochastic environment problem. The problem considered involves two or more decision-makers, each seeking to optimize their utility functions over the efficient solution set in a MOSILP problem. In addition, the proposed method can be used to identify the intersection between the efficient sets of the two stochastic multi-objective problems. Moreover, by optimizing two functions over the efficient set, decision-makers obtain a set of best compromise solutions in a finite number of iterations. It is also worth noting that the proposed method obtains this set without having to explore the whole efficient set of the MOSILP problem, which reduces the computational complexity compared to solving the MOSILP problem.

The proposed solution approach mainly relies on the combination of two techniques: the first is the L-shaped method, and the second involves an adapted branch-and-bound strategy. This combined approach is further enhanced by incorporating efficient tests and cuts to iteratively reduce the search space of the MOSILP problem. Moreover, the algorithm can be readily extended to accommodate multiple decision-makers by integrating their respective preferences into the efficient cuts. Experimental results showcase the effectiveness of the method, demonstrating excellent performance in terms of the number of iterations required while maintaining reasonable computation times. This contribution has prompted us to contemplate several future avenues of research. Our focus will be directed towards addressing the nonlinear aspect of the problem by considering the adaptation of our method to handle nonlinear multi-objective models, incorporating nonlinear preference functions. We remain committed to continuous improvement of the method in future endeavors.

General conclusion

This thesis was on the topic of 'Bi-objective optimization over efficient set of the multi-objective stochastic integer linear integer problem'. The first part laid the theoretical foundations by introducing the basic concepts of algebra, convex analysis, linear optimization, and stochastic optimization. It also provided an overview of the generalities of multi-objective optimization and stochastic multi-objective optimization and introduced the MOP and MOSP methods relevant to our study.

The second part presented our original contributions to the literature, with research results from the following works:

- **Optimization of a linear function over the efficient set of a multi-objective stochastic integer linear programming (MOSILP) problem:** This work introduces a new method to optimize a stochastic linear function over the efficient set of the MOSILP problem. this work is under review in the international journal RAIRO-Oper
- **Biobjective integer stochastic optimization over the integer stochastic efficient set:** This work, published in the international journal Pesquisa Operacional. (DOI: 10.1590/0101-7438.2023.043.00281853). This work proposes a method for optimising two linear stochastic function over the efficient set of a MOSILP problem,

Looking ahead, we are planning to :

- Improve the efficiency of algorithms by reducing resolution time.
- adaptation of our method to handle nonlinear multi-objective models.
- Explore more complex problems, such as non-convexity.
- Testing methods with mixed integer variables
- Publish the remaining work in international journals.

Bibliography

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

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

[Abdelaziz and Masri, 2010] Abdelaziz, F. B. and Masri, H. (2010). A compromise solution for the multiobjective stochastic linear programming under partial uncertainty. *European Journal of Operational Research*, 202(1):55–59.

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

[Amrouche and Moulaï, 2012] Amrouche, S. and Moulaï, M. (2012). Multi-objective stochastic integer linear programming with fixed recourse. *International Journal of Multicriteria Decision Making 9*, 2(4):355–378.

[Antunes et al., 2016] Antunes, C. H., Alves, M. J., and Clímaco, J. (2016). *Multiobjective linear and integer programming*. Springer.

[Badaoui et al., 2024] Badaoui, I., Moulaï, M., Chaiblaine, Y., and Chaabane, D. (2024). Biobjective integer stochastic optimization over the integer stochastic efficient set. *Pesquisa Operacional*, 44:e281853.

[Beale et al., 1986] Beale, E., Dantzig, G. B., and Watson, R. (1986). A first order approach to a class of multi-time-period stochastic programming problems. *Stochastic Programming 84 Part I*, pages 103–117.

[Belkhiri et al., 2022] Belkhiri, H., Chergui, M. E.-A., and Ouaïl, F. Z. (2022). Optimizing a linear function over an efficient set. *Operational Research*, pages 1–19.

[Bellman, 1958] Bellman, R. (1958). Dynamic programming and stochastic control processes. *Information and control*, 1(3):228–239.

[Ben Abdelaziz and Masmoudi, 2012] Ben Abdelaziz, F. and Masmoudi, M. (2012). A multiobjective stochastic program for hospital bed planning. *Journal of the Operational Research Society*, 63(4):530–538.

[Benayoun et al., 1971] Benayoun, R., De Montgolfier, J., Tergny, J., and Laritchev, O. (1971). Linear programming with multiple objective functions: Step method (stem). *Mathematical programming*, 1(1):366–375.

[Benson, 1984] Benson, H. P. (1984). Optimization over the efficient set. *Journal of Mathematical Analysis and Applications*, 98(2):562–580.

[Birge and Louveaux, 2011] Birge, J. R. and Louveaux, F. (2011). *Introduction to stochastic programming*. Springer Science & Business Media.

[Boland et al., 2015] Boland, N., Charkhgard, H., and Savelsbergh, M. (2015). A criterion space search algorithm for biobjective integer programming: The balanced box method. *INFORMS Journal on Computing*, 27(4):735–754.

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

[Boland et al., 2017] Boland, N., Charkhgard, H., and Savelsbergh, M. (2017). A new method for optimizing a linear function over the efficient set of a multiobjective integer program. *European journal of operational research*, 260(3):904–919.

[Bongo and Sy, 2023] Bongo, M. F. and Sy, C. L. (2023). A bi-objective integer linear optimization model for post-departure aircraft rerouting problem. In *Intelligent and Transformative Production in Pandemic Times: Proceedings of the 26th International Conference on Production Research*, pages 453–462. Springer.

[Bozorgi-Amiri et al., 2013] Bozorgi-Amiri, A., Jabalameli, M. S., and Mirzapour Al-e Hashem, S. (2013). A multi-objective robust stochastic programming model for disaster relief logistics under uncertainty. *OR spectrum*, 35(4):905–933.

[Caballero et al., 2004] Caballero, R., Cerdá, E., del Mar Munoz, M., and Rey, L. (2004). Stochastic approach versus multiobjective approach for obtaining efficient solutions in stochastic multiobjective programming problems. *European Journal of Operational Research*, 158(3):633–648.

[Caballero et al., 2001] Caballero, R., Cerdá, E., Munoz, M., Rey, L., and Stancu-Minasian, I. (2001). Efficient solution concepts and their relations in stochastic multiobjective programming. *Journal of Optimization Theory and Applications*, 110:53–74.

[Cao et al., 2018] Cao, C., Li, C., Yang, Q., Liu, Y., and Qu, T. (2018). A novel multi-objective programming model of relief distribution for sustainable disaster supply chain in large-scale natural disasters. *Journal of Cleaner Production*, 174:1422–1435.

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

[Chaiblaine and Moulaï, 2021] Chaiblaine, Y. and Moulaï, M. (2021). An exact method for optimizing a quadratic function over the efficient set of multiobjective integer linear fractional program. *Optimization Letters*, pages 1–15.

[Chaiblaine et al., 2020] Chaiblaine, Y., Moulaï, M., and Cherfaoui, Y. (2020). An exact method for optimizing two linear fractional functions over the efficient set of a multiobjective integer linear fractional program. *arXiv preprint arXiv:2003.05364*.

[Charnes and Cooper, 1962] Charnes, A. and Cooper, W. (1962). Chance constraints and normal deviates. *Journal of the American statistical association*, 57(297):134–148.

[Cherfaoui and Moulaï, 2021] Cherfaoui, Y. and Moulaï, M. (2021). Biobjective optimization over the efficient set of multiobjective integer programming problem. *Journal of Industrial & Management Optimization*, 17(1):117.

[Chergui et al., 2008] Chergui, M. E.-A., Moulaï, M., and Zohra Ouaïl, F. (2008). Solving the multiple objective integer linear programming problem. In *International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences*, pages 69–76. Springer.

[Cohon, 2013] Cohon, J. L. (2013). *Multiobjective programming and planning*. Courier Corporation.

[Dantzig, 2002] Dantzig, G. B. (2002). Linear programming. *Operations research*, 50(1):42–47.

[Djamal and Marc, 2010] Djamal, C. and Marc, P. (2010). A method for optimizing over the integer efficient set. *Journal of industrial and management optimization*, 6(4):811.

[Dos Santos and Oliveira, 2019] Dos Santos, F. S. P. and Oliveira, F. (2019). An enhanced l-shaped method for optimizing periodic-review inventory control problems modeled via two-stage stochastic programming. *European Journal of Operational Research*, 275(2):677–693.

[Drici et al., 2018] Drici, W., Ouail, F. Z., and Moulaï, M. (2018). Optimizing a linear fractional function over the integer efficient set. *Annals of Operations Research*, 267(1):135–151.

[Ecker and Kouada, 1978] Ecker, J. G. and Kouada, I. (1978). Finding all efficient extreme points for multiple objective linear programs. *Mathematical Programming*, 14(1):249–261.

[Ecker and Song, 1994] Ecker, J. G. and Song, J. H. (1994). Optimizing a linear function over an efficient set. *Journal of Optimization Theory and Applications*, 83(3):541–563.

[Ehrgott, 2012] Ehrgott, M. (2012). Vilfredo pareto and multi-objective optimization. *Doc. math*, 8:447–453.

[Gadegaard et al., 2019] Gadegaard, S. L., Nielsen, L. R., and Ehrgott, M. (2019). Bi-objective branch-and-cut algorithms based on lp relaxation and bound sets. *INFORMS Journal on Computing*, 31(4):790–804.

[Geoffrion et al., 1972] Geoffrion, A. M., Dyer, J. S., and Feinberg, A. (1972). An interactive approach for multi-criterion optimization, with an application to the operation of an academic department. *Management science*, 19(4-part-1):357–368.

[Goicoechea, 1980] Goicoechea, A. (1980). Deterministic equivalents for use in multiobjective, stochastic programming. *IFAC Proceedings Volumes*, 13(3):31–40.

[Goicoechea et al., 1976] Goicoechea, A., Dukstein, L., and Bulfin, R. (1976). Multiobjective stochastic programming the protrade-method. *Operation Research Society of America*.

[Gomory, 2002] Gomory, R. E. (2002). Early integer programming. *Operations Research*, 50(1):78–81.

[Gunantara, 2018] Gunantara, N. (2018). A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, 5(1):1502242.

[Halffmann et al., 2022] Halffmann, P., Schäfer, L. E., Dächert, K., Klamroth, K., and Ruzika, S. (2022). Exact algorithms for multiobjective linear optimization problems with integer variables: A state of the art survey. *Journal of Multi-Criteria Decision Analysis*.

[Haneveld and Van der Vlerk, 2020] Haneveld, K. and Van der Vlerk, M. H. (2020). *Stochastic programming*. Springer.

[Hansotia, 1980] Hansotia, B. J. (1980). Stochastic linear programming with recourse: A tutorial. *Decision Sciences*, 11(1):151–168.

[Higle and Sen, 1991] Higle, J. L. and Sen, S. (1991). Stochastic decomposition: An algorithm for two-stage linear programs with recourse. *Mathematics of operations research*, 16(3):650–669.

[Ibrahim et al., 2018] Ibrahim, H., Aburukba, R. O., and El-Fakih, K. (2018). An integer linear programming model and adaptive genetic algorithm approach to minimize energy consumption of cloud computing data centers. *Computers & Electrical Engineering*, 67:551–565.

[Ignizio and Cavalier, 1994] Ignizio, J. P. and Cavalier, T. M. (1994). *Linear programming*. Prentice-Hall, Inc.

[Irmansyah et al., 2022] Irmansyah, A. Z., Chaerani, D., and Rusyaman, E. (2022). A systematic review on integer multi-objective adjustable robust counterpart optimization model using benders decomposition. *JTAM (Jurnal Teori dan Aplikasi Matematika)*, 6(3):678–698.

[Jahanshahloo et al., 2004] Jahanshahloo, G. R., Lotfi, F. H., Shoja, N., and Tohidi, G. (2004). A method for generating all the efficient solutions of a 0-1 multi-objective linear programming problem. *Asia-Pacific Journal of Operational Research*, 21(01):127–139.

[Jorge, 2009] Jorge, J. M. (2009). An algorithm for optimizing a linear function over an integer efficient set. *European Journal of Operational Research*, 195(1):98–103.

[Kall, 1976] Kall, P. (1976). Stochastic linear programming. econometrics and operations research, vol xxi.

[Kall and Kall, 1976] Kall, P. and Kall, P. (1976). Prerequisites. *Stochastic Linear Programming*, pages 1–10.

[Kall et al., 1994] Kall, P., Wallace, S. W., and Kall, P. (1994). *Stochastic programming*. Springer.

[Karloff, 2008] Karloff, H. (2008). *Linear programming*. Springer Science & Business Media.

[KATAOKA, 1962] KATAOKA, S. (1962). Stochastic programming and its application to production horizon probi. em. *Hitotsubashi journal of arts and sciences*.

[Kataoka, 1963] Kataoka, S. (1963). A stochastic programming model. *Econometrica: Journal of the Econometric Society*, pages 181–196.

[Kim and de Weck, 2006] Kim, I. Y. and de Weck, O. L. (2006). Adaptive weighted sum method for multiobjective optimization: a new method for pareto front generation. *Structural and multidisciplinary optimization*, 31(2):105–116.

[Kolbin, 1977] Kolbin, V. V. (1977). *Stochastic programming*. Number 14. Springer Science & Business Media.

[Li and Grossmann, 2018] Li, C. and Grossmann, I. E. (2018). An improved l-shaped method for two-stage convex 0–1 mixed integer nonlinear stochastic programs. *Computers & Chemical Engineering*, 112:165–179.

[Liao et al., 2017] Liao, S.-H., Hsieh, C.-L., and Ho, W.-C. (2017). Multi-objective evolutionary approach for supply chain network design problem within online customer consideration. *RAIRO-Operations Research*, 51(1):135–155.

[Lokman, 2021] Lokman, B. (2021). Optimizing a linear function over the nondominated set of multi-objective integer programs. *International Transactions in Operational Research*, 28(4):2248–2267.

[Lokman and Köksalan, 2013] Lokman, B. and Köksalan, M. (2013). Finding all nondominated points of multi-objective integer programs. *Journal of Global Optimization*, 57(2):347–365.

[Lucena and Beasley, 1996] Lucena, A. and Beasley, J. E. (1996). Branch and cut algorithms. *Advances in linear and integer programming*, 4:187–221.

[Mahdi and Chaabane, 2015] Mahdi, S. and Chaabane, D. (2015). A linear fractional optimization over an integer efficient set. *RAIRO-Operations Research*, 49(2):265–278.

[Marler and Arora, 2010] Marler, R. T. and Arora, J. S. (2010). The weighted sum method for multi-objective optimization: new insights. *Structural and multidisciplinary optimization*, 41:853–862.

[Mavrotas, 2009] Mavrotas, G. (2009). Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. *Applied mathematics and computation*, 213(2):455–465.

[Miettinen, 1999] Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers.

[Moayedi and Sadeghian, 2023] Moayedi, M. and Sadeghian, R. (2023). A multi-objective stochastic programming approach with untrusted suppliers for green supply chain design by uncertain demand, shortage, and transportation costs. *Journal of Cleaner Production*, 408:137007.

[Morrison et al., 2016] Morrison, D. R., Jacobson, S. H., Sauppe, J. J., and Sewell, E. C. (2016). Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization*, 19:79–102.

[Motahari et al., 2023] Motahari, R., Alavifar, Z., Andaryan, A. Z., Chipulu, M., and Saberi, M. (2023). A multi-objective linear programming model for scheduling part families and designing a group layout in cellular manufacturing systems. *Computers & Operations Research*, 151:106090.

[Moulaï and Drici, 2018] Moulaï, M. and Drici, W. (2018). An indefinite quadratic optimization over an integer efficient set. *Optimization*, 67(8):1143–1156.

[Nabli, 2009] Nabli, H. (2009). An overview on the simplex algorithm. *Applied Mathematics and Computation*, 210(2):479–489.

[Nakayama and Sawaragi, 1984] Nakayama, H. and Sawaragi, Y. (1984). Satisficing trade-off method for multiobjective programming. In *Interactive Decision Analysis: Proceedings of an International Workshop on Interactive Decision Analysis and Interpretative Computer Intelligence Held at the International Institute for Applied Systems Analysis (IIASA), Laxenburg, Austria September 20–23, 1983*, pages 113–122. Springer.

[Obal et al., 2013] Obal, T. M., Volpi, N. M. P., and Miloca, S. A. (2013). Multiobjective approach in plans for treatment of cancer by radiotherapy. *Pesquisa Operacional*, 33:269–282.

[Ouaïl et al., 2017] Ouaïl, F., EA, M, C., and Moulaï, M. (2017). An exact method for optimizing a linear function over an integer efficient set. *WSEAS Transactions on Circuits and Systems*, 16(3):141–148.

[Ouail et al., 2017] Ouail, F. Z., Chergui, M. E.-a., and Moulai, M. (2017). An exact method for optimizing a linear function over an integer efficient set. *International Journal of Mathematical and Computational Methods*, 2.

[Pal and Charkhgard, 2019] Pal, A. and Charkhgard, H. (2019). A feasibility pump and local search based heuristic for bi-objective pure integer linear programming. *INFORMS Journal on Computing*, 31(1):115–133.

[Parragh and Tricoire, 2019] Parragh, S. N. and Tricoire, F. (2019). Branch-and-bound for bi-objective integer programming. *INFORMS Journal on Computing*, 31(4):805–822.

[Philip, 1972] Philip, J. (1972). Algorithms for the vector maximization problem. *Mathematical programming*, 2(1):207–229.

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

[Raith et al., 2024] Raith, A., Lusby, R., and Yousefkhan, A. A. S. (2024). Benders decomposition for bi-objective linear programs. *European Journal of Operational Research*.

[Ramezani et al., 2013] Ramezani, M., Bashiri, M., and Tavakkoli-Moghaddam, R. (2013). A new multi-objective stochastic model for a forward/reverse logistic network design with responsiveness and quality level. *Applied mathematical modelling*, 37(1-2):328–344.

[Rasmi and Türkay, 2019] Rasmi, S. A. B. and Türkay, M. (2019). Gondef: an exact method to generate all non-dominated points of multi-objective mixed-integer linear programs. *Optimization and Engineering*, 20:89–117.

[Ren et al., 2021] Ren, C., Xie, Z., Zhang, Y., Wei, X., Wang, Y., and Sun, D. (2021). An improved interval multi-objective programming model for irrigation water allocation by considering energy consumption under multiple uncertainties. *Journal of Hydrology*, 602:126699.

[Sengupta et al., 2012] Sengupta, J. K., Charnes, A., and Cooper, W. W. (2012). Stochastic programming methods in economic models. *This series aims to report new developments in mathematical economics and operations research and teaching quickly, informally and at a high level. The type of material considered for publication includes: 1. Preliminary drafts of original papers and monographs 2. Lectures on a new field, or presenting a new angle on a classical field*, page 390.

[Shidpour et al., 2013] Shidpour, H., Shahrokhi, M., and Bernard, A. (2013). A multi-objective programming approach, integrated into the topsis method, in order to optimize product design; in three-dimensional concurrent engineering. *Computers & Industrial Engineering*, 64(4):875–885.

[Sierra Altamiranda and Charkhgard, 2019] Sierra Altamiranda, A. and Charkhgard, H. (2019). A new exact algorithm to optimize a linear function over the set of efficient solutions for biobjective mixed integer linear programs. *INFORMS Journal on Computing*, 31(4):823–840.

[Sierra-Altamiranda et al., 2020] Sierra-Altamiranda, A., Charkhgard, H., Eaton, M., Martin, J., Yurek, S., and Udell, B. J. (2020). Spatial conservation planning under uncertainty using modern portfolio theory and nash bargaining solution. *Ecological Modelling*, 423:109016.

[Soylu, 2015] Soylu, B. (2015). Heuristic approaches for biobjective mixed 0–1 integer linear programming problems. *European Journal of Operational Research*, 245(3):690–703.

[Stancu-Minasian, 1990] Stancu-Minasian, I. (1990). Overview of different approaches for solving stochastic programming problems with multiple objective functions. *Stochastic versus fuzzy approaches to multiobjective mathematical programming under uncertainty*, 6:71–101.

[Steuer et al., 1986] Steuer, R., Qi, Y., and Hirschberger, M. (1986). Multiple criteria decision making. *John Whiley*.

[Sylva and Crema, 2004] Sylva, J. and 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.

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

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

[Teghem, 1990] Teghem, J. (1990). "strange": An interactive method for multiobjective stochastic linear programming, and "strange-momix" its extension to integer variables. In *Stochastic Versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty*, pages 103–115. Springer.

[Teghem Jr et al., 1986] Teghem Jr, J., Dufrane, D., Thauvoye, M., and Kunsch, P. (1986). Strange: an interactive method for multi-objective linear programming under uncertainty. *European Journal of Operational Research*, 26(1):65–82.

[Teghem Jr and Kunsch, 1987] Teghem Jr, J. and Kunsch, P. (1987). Momix–an interactive method for mixed integer linear programming. *Submitted for publication*.

[Torres et al., 2022] Torres, J. J., Li, C., Apap, R. M., and Grossmann, I. E. (2022). A review on the performance of linear and mixed integer two-stage stochastic programming software. *Algorithms*, 15(4):103.

[Urli and Nadeau, 1990] Urli, B. and Nadeau, R. (1990). Multiobjective stochastic linear programming with incomplete information: a general methodology. In *Stochastic versus fuzzy approaches to multiobjective mathematical programming under uncertainty*, pages 131–161. Springer.

[Urli and Nadeau, 2004] Urli, B. and Nadeau, R. (2004). Promise/scenarios: An interactive method for multiobjective stochastic linear programming under partial uncertainty. *European journal of operational research*, 155(2):361–372.

[Ušpurienė et al., 2018] Uspurienė, A., Sakalauskas, L., and Gricius, G. (2018). Modified l-shaped decomposition method with scenario aggregation for a two-stage stochastic programming problem. *Information Technology and Control*, 47(4):728–738.

[Vafadarnikjoo et al., 2023] Vafadarnikjoo, A., Moktadir, M. A., Paul, S. K., and Ali, S. M. (2023). A novel grey multi-objective binary linear programming model for risk assessment in supply chain management. *Supply Chain Analytics*, 2:100012.

[Van Slyke and Wets, 1969] Van Slyke, R. M. and Wets, R. (1969). L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4):638–663.

[Verma et al., 2021] Verma, S., Pant, M., and Snasel, V. (2021). A comprehensive review on nsga-ii for multi-objective combinatorial optimization problems. *IEEE access*, 9:57757–57791.

[Walkup and Wets, 1967] Walkup, D. W. and Wets, R. J.-B. (1967). Stochastic programs with recourse. *SIAM Journal on Applied Mathematics*, 15(5):1299–1314.

[White, 1974] White, D. J. (1974). Dynamic programming and probabilistic constraints. *Operations Research*, 22(3):654–664.

[Wierzbicki, 1980] Wierzbicki, A. P. (1980). The use of reference objectives in multiobjective optimization. In *Multiple criteria decision making theory and application: Proceedings of the third conference Hagen/Königswinter, West Germany, August 20–24, 1979*, pages 468–486. Springer.

[Zeleny, 2012] Zeleny, M. (2012). *Linear multiobjective programming*, volume 95. Springer Science & Business Media.

[Zerdani and Moulai, 2011] Zerdani, O. and Moulai, M. (2011). Optimization over an integer efficient set of a multiple objective linear fractional problem.

[Zhang et al., 2021] Zhang, Y., Fu, Z., Xie, Y., Li, Z., Liu, Y., Hu, Q., and Guo, H. (2021). Multi-objective programming for energy system based on the decomposition of carbon emission driving forces: A case study of guangdong, china. *Journal of Cleaner Production*, 309:127410.