

MULTI-OBJECTIVE PORTFOLIO OPTIMIZATION USING AN AUGMENTED LAGRANGIAN ALGORITHM

SOMÉ YIRNIBA BENOIT¹, TOUGMA APPOLINAIRE^{2,3,*}, SOMÉ KOUNHINIR^{1,3}

¹Département de Mathématiques, Laboratoire d'Analyse Numérique d'Informatique et de BIOmathématique, Université Joseph KI-ZERBO, Ouagadougou, Burkina Faso

²Faculté des sciences et technologies, Université Lédéa Bernard Ouédraogo, Ouahigouya, Burkina Faso

³Département de Mathématiques, Laboratoire de Mathématiques, Informatique et Applications (L@MIA), Université Norbert Zongo, Koudougou, BP 376, Burkina Faso.

*Corresponding author: appolinaire.tougma19@gmail.com

Received Aug. 19, 2025

ABSTRACT. This paper proposes a multiobjective optimization model for asset portfolio selection, incorporating investor preferences through inequality and equality constraints. A solution method based on the augmented Lagrangian approach is developed, thus avoiding the reduction of the multiobjective problem into a parametric single-objective framework. The proposed algorithm can handle portfolio optimization problems involving both linear and nonlinear constraints. To validate its effectiveness, a numerical simulation was conducted, comparing its performance to that of the NSGA-II algorithm. The results demonstrate that our approach outperforms NSGA-II in terms of performance, confirming the robustness of the proposed algorithm.

2020 Mathematics Subject Classification. 90C29; 90C30; 91G10; 65K05.

Key words and phrases. Multiobjective Portfolio, Augmented Lagrangian, Portfolio model, Pareto front.

1. INTRODUCTION

Mathematical models for multiobjective optimization problems are models that consider multiple objectives of the decision-maker, as opposed to single-objective models. Solving these multiobjective models provides multiple solutions to the decision-maker, known as Pareto optimal solutions or good compromise solutions. This is justified by the fact that objectives are often conflicting, meaning improving one objective may lead to the deterioration of others. Numerous methods have been proposed in the literature for solving multiobjective optimization problems, originating from various domains such as the environment, electricity, and finance, with a particular focus on portfolio selection [23, 24].

The aim of portfolio optimization is to find an optimal set of assets to invest in, as well as the optimal investment allocation for each asset. The selection and optimal weighting of assets to maximize the total investment return, minimize total investment risk, and achieve other goals are common objectives in the literature [19,23]. Some researchers have worked on single-objective portfolio models; examples include studies such as [20,26]. Given that most problems involve conflicting objectives or criteria, other researchers have developed multiobjective models to account for the diverse preferences of decision-makers [13,19,23], aiming to provide multiple solutions from which decision-makers can choose based on their preferences. Among portfolio optimization models, whether single-objective or multiobjective, some involve linear constraints while others involve nonlinear constraints, which make solving these problems more challenging. Constraints are conditions associated with objectives to make solutions more acceptable. For the same objectives in a given problem, constraints can always be modified, i.e., removed or added, to improve the problem's solutions. The desire to improve portfolio models to achieve better results remains a current research topic.

There are several approaches for solving constrained optimization problems [17,28]. Among these approaches, the augmented Lagrangian method has proven effective for solving constrained optimization problems. Researchers have applied it to portfolio problems, including [3], who developed an algorithm for portfolio optimization problems with value-at-risk (VaR) constraints and transaction costs. They demonstrated that the augmented Lagrangian approach is a suitable tool for handling problems involving a high number of variables and constraints. [22] proposed an augmented Lagrangian algorithm for solving a portfolio problem with lower-level and cardinality constraints. Their experimental results showed that their algorithm achieves more feasible solutions with fewer iterations than [18] under the same parameters. [7] based on a class of optimization problems with constraints where the objective function is a sum of a smooth function and a nonconvex, non-Lipschitz function. Their work employed a nonmonotone proximal gradient method for solving the augmented Lagrangian subproblems and applied this method to portfolio selection models. [14] worked on stochastic optimization problems with risk aversion, incorporating a risk-shaping constraint in the form of a stochastic order relation for solving portfolio problems using the augmented Lagrangian method.

A specific augmented Lagrangian algorithm, called Algencan (a Fortran code for minimizing a smooth function), was described by [3,5]. It is based on the augmented Lagrangian approach by [11,25] for portfolio optimization to address market fluctuations. Most of his previously mentioned works addressed single-objective portfolio problems. Other studies have investigated the multiobjective case, including [2,8,21], some of which transform the original problem into a parametric single-objective problem [2,6] or use stochastic algorithms [1,8,15] for solving these problems.

In this work, we propose a multiobjective portfolio model that takes into account constraints related to investments in assets with a given minimal threshold. The main contributions and highlights of this paper are summarized as follows:

- Proposal of a new multiobjective portfolio optimization model;
- Proposal of an algorithm for solving multiobjective portfolio optimization problems;
- Application to a test case accompanied by a comparison with other resolution methods existing in the literature.

In the remainder of this paper, we present some preliminaries on multiobjective optimization in Section 2. In Section 3, we describe the proposed multiobjective portfolio optimization model. In section 4, we will apply the proposed model to a test case using data from Yahoo Finance, along with a comparison with other methods in the literature. We conclude with a summary and future research directions in Section 5.

2. PRELIMINARIES

In this work, we consider the formulation of a multiobjective optimization problem as follows:

$$\begin{aligned} \min F(x) &= \left(f_1(x), f_2(x), \dots, f_p(x) \right)^T \\ \text{s.t. } &\begin{cases} g_i(x) \leq 0, & i = 1, \dots, m, \\ h_j = 0, & j = 1, \dots, q, \\ x \in \mathbb{R}^n, \end{cases} \end{aligned} \quad (1)$$

where $F : \mathbb{R}^n \rightarrow \mathbb{R}^p$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $h : \mathbb{R}^n \rightarrow \mathbb{R}^q$.

Let $\mathbf{D} = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i = 1, \dots, m, h_j(x) = 0 j = 1, \dots, q\}$ denote the feasible solution set of problem (1). To characterize a compromise solution of problem (1), we provide the following definitions.

Definition 1 ([16]). A point $x^* \in \mathbf{D}$ is called Pareto optimal if there is no other point $x \in \mathbf{D}$ such that $f_i(x) \leq f_i(x^*)$ for all $i = 1; \dots, p$ with at least one $f_i(x) < f_i(x^*)$ for some $i = 1; \dots, p$.

Definition 2 ([16,27]). A vector $x^* \in \mathbf{D}$ is said to be locally Pareto optimal if there exists $r > 0$ such that x^* is Pareto optimal on $\mathbf{D} \cap B(x^*, r)$, where $B(x^*, r)$ is the ball centered at x^* with radius r .

Definition 3 ([16,27]). A vector $x^* \in \mathbf{D}$ is said to be weakly Pareto optimal if there is no other vector $x \in \mathbf{D}$ such that $f_i(x) \leq f_i(x^*)$ for all $i = 1, \dots, p$.

Using Definitions 2 and 3, we characterize an optimal and a weakly optimal Pareto solution through the following lemma.

Lemma 1 ([27]).

- (1) A point $x^* \in \mathbf{D}$ is a Pareto optimal solution of problem (1) if and only if, for all $x \in \mathbf{D}$, at least one of the following conditions holds: (i) $\max_{i=1,\dots,m} \{f_i(x) - f_i(x^*)\} > 0$; (ii) $\min_{i=1,\dots,m} \{f_i(x) - f_i(x^*)\} \geq 0$;
- (2) A point $x^* \in \mathbf{D}$ is a weakly Pareto optimal solution of problem (1) if and only if, for all $x \in \mathbf{D}$, we have $\max_{i=1,\dots,m} \{f_i(x) - f_i(x^*)\} \geq 0$.

Definition 4 ([16,27]). The Pareto front, or trade-off surface, is defined as the image of the efficient or Pareto optimal solutions under the objective functions.

3. A MULTI-OBJECTIVE PORTFOLIO PROBLEM

3.1. Multiobjective Augmented Lagrangian algorithm. Considering problem (1), the Augmented Lagrangian function transforms it into a parametric unconstrained multiobjective problem, as described in [16]:

$$\mathcal{L}\left(x, \lambda^k, \mu^k, \eta_k\right) = F(x) + \frac{\eta_k}{2} \left[\sum_{j=1}^q \left[h_j(x) + \frac{\lambda_j^k}{\eta_k} \right]^2 + \sum_{i=1}^m \left[\max \left(0, g_i(x) + \frac{\mu_i^k}{\eta_k} \right) \right]^2 \right] \cdot e, \quad (2)$$

where $\eta_k > 0$ is a penalty parameter, $\lambda \in \mathbb{R}_+^q$ and $\mu \in \mathbb{R}_+^m$ are the approximate Lagrange multiplier vectors associated with equality and inequality constraints, respectively, and $e = (1, 1, \dots, 1)$ is the unit vector in $\mathbb{R}^p$.

The update processes at each iteration for the penalty parameters and Lagrange multipliers are defined as follows.

For the Lagrange multipliers:

$$\lambda^{k+1} = \lambda^k + \eta_k h(x^k); \text{ and } \mu^{k+1} = \mu^k + \eta_k g(x^k).$$

To update the penalty parameter, we first compute, as presented in [4,9]:

$$V_i^k = \min \left\{ -g_i(x^k), \frac{\mu_i^k}{\eta_k} \right\}, \quad \text{for } i = 1, \dots, m$$

and:

$$\begin{cases} \text{If } \max \{ \|h(x^k)\|, \|V^k\| \} \leq \sigma \max \{ \|h(x^{k-1})\|, \|V^{k-1}\| \}, & \eta_{k+1} = \eta_k, \\ \text{Else} & \eta_{k+1} = \alpha \eta_k. \end{cases} \quad (3)$$

The resolution algorithm is presented as follows:

Algorithm 1: Multiobjective Augmented Lagrangian Algorithm

Input: $\mu^0 \in \mathbb{R}^p$, $\lambda^0 \geq 0$, $\rho > 1$, $\sigma \in (0, 1)$, $\eta_0 > 0$, $\lambda^{inf} \in \mathbb{R}^+$, $\lambda^{sup} \in \mathbb{R}^+$, $\bar{\mu} \in \mathbb{R}^+$, $\mathcal{X}^0 \in \mathbb{R}^n$ (a list of feasible, non-dominated points for the original problem), $\{\epsilon_k\} \in \mathbb{R}$ (a decreasing sequence).

1 **for** $k = 1, \dots$ **do**

2 Define $\mathcal{L}$, the current Augmented Lagrangian function, as:

$$\mathcal{L}(x, \lambda^k, \mu^k, \eta_k) = F(x) + \frac{\eta_k}{2} \left[\sum_{j=1}^q \left[h_j(x) + \frac{\lambda_j^k}{\eta_k} \right]^2 + \sum_{i=1}^m \left[\max \left(0, g_i(x) + \frac{\mu_i^k}{\eta_k} \right) \right]^2 \right] \cdot e$$

3 $\mathcal{X}^k = \{x \in \mathcal{X} \mid \exists y \in \mathcal{X} \text{ such that } \mathcal{L}(y, \lambda^k, \mu^k, \eta_k) \leq \mathcal{L}(x, \lambda^k, \mu^k, \eta_k)\};$

4 Set $\mathcal{X}^{k+1}$ as a set of non-dominated solutions for the problem:

$$\min_{x \in D} \mathcal{L}(x, \lambda^k, \mu^k, \eta_k) \quad (4)$$

5 $\mathcal{X}^{k+1} = \mathcal{X}_{\text{tmp}};$

6 **for** $j = 1, \dots, m$ **do**

7 Set $\lambda_j^{k+1} = \max \left\{ \lambda_j^{inf}, \min \left\{ \lambda_j^k + \eta_k \max_{x \in X^{k+1}} \{ |h(x)| \}, \lambda_j^{sup} \right\} \right\};$

8 **for** $i = 1, \dots, q$ **do**

9 Set $V_i^{k+1} = \min \left\{ \min_{x \in X^{k+1}} \{-g(x)\}, \frac{\mu_i^k}{\eta_k} \right\};$

10 Set $\mu_i^{k+1} = \max \left\{ 0, \min \left\{ \mu_i^k + \eta_k \max_{x \in X^{k+1}} \{-g(x)\}, \bar{\mu} \right\} \right\};$

11 **if** $\max \{ \|h(x^{k+1})\|, \|V^{k+1}\| \} \leq \sigma \max \{ \|h(x^k)\|, \|V^k\| \}$ **or**
 $\left(\exists x^{k+1} \in \mathcal{X}^{k+1} \text{ s.t. } g_i(x^{k+1}) < 0, h_j(x^{k+1}) = 0 \right)$ **for some** $i \in \{1, \dots, m\}, j \in \{1, \dots, q\}$ **then**

12 Set $\eta_{k+1} = \eta_k;$

13 **else**

14 Set $\eta_{k+1} = \alpha \eta_k;$

The algorithm begins by initializing a set of non-dominated points X^0 and the initial parameters at step 0. From step 2 to step 5 of the algorithm, the goal is to determine a set of non-dominated points X^k . This set is derived by considering the points in X^{k-1} for all $k > 1$. Specifically, for each point in X^{k-1} , a new ε_k -Pareto point x^k is sought by solving the function $\mathcal{L}(x, \lambda^k, \mu^k, \eta_k)$. After determining the set of non-dominated points X^k , the update of the parameters λ and μ , which are the Lagrange multipliers associated with the equality constraints h_j and the inequality constraints g_i , respectively, begins from step 6 to step 10.

Thus, the multipliers $(\lambda^{k+1}, \mu^{k+1})$ are chosen as the projection of the multipliers (λ^k, μ^k) onto the intervals defined in steps 7 and 10 of Algorithm 1.

For example, the projection rule for λ^k can be described as follows:

- If $\lambda^{k+1} > \lambda^{\sup}$, then $\bar{\lambda}^{k+1} = \lambda^{\sup}$.
- If $\lambda^{k+1} < \lambda^{\inf}$, then $\bar{\lambda}^{k+1} = \lambda^{\inf}$.
- If $\lambda^{k+1} \in [\lambda^{\inf}, \lambda^{\sup}]$, then $\bar{\lambda}^{k+1} = \lambda^{k+1}$.

The penalty parameter update starts from step 8, according to system (3). Furthermore, the proper definition of Algorithm 1 relies on step 5, which we will discuss in detail in the following sections. We begin the convergence analysis by stating the following assumptions:

Assumption 1. *The set*

$$\mathbf{D} = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i = 1, \dots, m, h_j(x) = 0, j = 1, \dots, q\}$$

is non-empty and compact.

Regarding the feasibility and optimality results of the generated solutions, we obtain the following results. These results are similar to those proposed in the works of [9,10]. In particular, Proposition 3 establishes that all sequences generated by Algorithm 1 are feasible, and Proposition 4 demonstrates that all sequences generated by this algorithm converge to a Pareto optimal point.

Proposition 2. *Let $\{X^{k+1}\}$ be a sequence of point sets generated by Algorithm 1. Then, for each k and for each $x^{k+1} \in X^{k+1}$, we have x^{k+1} that is not dominated by any other point in X^{k+1} with respect to $\mathcal{L}(x, \lambda^k, \mu^k, \eta_k)$, that is, there does not exist $y \in X^{k+1}$ such that $\mathcal{L}(y, \lambda^k, \mu^k, \eta_k) \prec \mathcal{L}(x^{k+1}, \lambda^k, \mu^k, \eta_k)$;*

Proof. We have X^0 which is a set of non-dominated points and from step 4 of algorithm 1, for all $k > 0$, $k \in K$, X^k is formed of non-dominated points. $\square$

Proposition 3. *Let X^k be a set of point sequences generated by algorithm 1 and x^* be a limit point of a point sequence $\{x^k\} \in X^k$. Then, x^* is a feasible point of problem (1).*

Proof. Let $K \subseteq \mathbb{N}$, $\{x^k\}$ be a sequence of points generated by Algorithm 1 with x^* as its limit point. Since $\mathbf{D}$ is a compact set and $x^k \in \mathbf{D}$, it follows that $x^* \in \mathbf{D}$. According to step 10 of algorithm 1, we have $\mu_i^k \geq 0$ for all k . Thus, two cases can be considered: $\{\eta_k\}$ bounded and $\{\eta_k\}$ unbounded ($\eta_k \rightarrow \infty$).

- Case 1: $\{\eta_k\}$ is bounded.

In this case, there exists k_0 such that for all $k \geq k_0$, the condition

$$\max \left\{ \|h(x^{k+1})\|, \|V^{k+1}\| \right\} \leq \sigma \max \left\{ \|h(x^k)\|, \|V^k\| \right\} \text{ is satisfied.} \quad \text{This means}$$

$$\lim_{\substack{k \rightarrow \infty \\ k \in K}} \|V^k\| = 0 \text{ and } \lim_{\substack{k \rightarrow \infty \\ k \in K}} \|h_j(x^k)\| = 0,$$

$$\text{therefore } \lim_{k \rightarrow \infty} V_i^{k+1} = \lim_{k \rightarrow \infty} \min \left\{ \min_{x \in X^{k+1}} \left\{ -g_i(x^{k+1}) \right\}, \frac{\mu_i^k}{\eta_k} \right\} = 0 \text{ and } \lim_{k \rightarrow \infty} h(x^{k+1}) = 0.$$

Since $\mu_i^k \geq 0$ for all k , implies that $g_i(x^*) = \lim_{\substack{k \rightarrow \infty \\ k \in K}} g_i(x^{k+1}) \leq 0$, moreover, if $g_i(x^*) < 0$, then

$\mu_i^k = 0$ and $h_j(x^*) = \lim_{\substack{k \rightarrow \infty \\ k \in K}} h_j(x^{k+1}) = 0$, consequently, we have $h_j(x^*) = 0$ and $g_i(x^*) \leq 0$, therefore x^* is a feasible point of problem (2.1).

- Case 2: η_k unbounded ($\eta_k \rightarrow \infty$).

According to the assumptions, a feasible point y of problem (1) exists, which means $h(y) = 0$ and $g(y) \leq 0$. Following the indications of Algorithm 1 and using Lemma 1, for all $k \in K$, we have:

$$\min_{\substack{l=1,\dots,p \\ \forall y \in D}} \left\{ f_l(y) + \frac{\eta_k}{2} \left[\sum_{j=1}^q \left[h_j(y) + \frac{\lambda_j^k}{\eta_k} \right]^2 + \sum_{i=1}^m \left[\max \left(0, g_i(y) + \frac{\mu_i^k}{\eta_k} \right) \right]^2 \right. \right. \\ \left. \left. - f_l(x^{k+1}) - \frac{\eta_k}{2} \left[\sum_{j=1}^q \left[h_j(x^{k+1}) + \frac{\lambda_j^k}{\eta_k} \right]^2 + \sum_{i=1}^m \left[\max \left(0, g_i(x^{k+1}) + \frac{\mu_i^k}{\eta_k} \right) \right]^2 \right] \right\} \geq 0;$$

$$\Rightarrow \min_{\substack{l=1,\dots,p \\ \forall y \in D}} \left\{ f_l(x^{k+1}) + \frac{\eta_k}{2} \left[\sum_{j=1}^q \left(h_j(x^{k+1}) + \frac{\lambda_j^k}{\eta_k} \right)^2 + \sum_{i=1}^m \left(\max \left(0, g_i(x^{k+1}) + \frac{\mu_i^k}{\eta_k} \right) \right)^2 \right] \right. \\ \left. - f_l(y) - \frac{\eta_k}{2} \left[\sum_{j=1}^q \left(h_j(y) + \frac{\lambda_j^k}{\eta_k} \right)^2 + \sum_{i=1}^m \left(\max \left(0, g_i(y) + \frac{\mu_i^k}{\eta_k} \right) \right)^2 \right] \right\} \leq 0.$$

Since the sequences $\{\mu_i^k\}$ and $\{\lambda_j^k\}$ are bounded by definition, which implies that $\lim_{\substack{k \rightarrow \infty \\ k \in K}} \frac{\mu_i^k}{\eta_k} = 0$ and $g_i(x^*) < 0$ then $\lim_{\substack{k \rightarrow \infty \\ k \in K}} \frac{\lambda_j^k}{\eta_k} = 0$ and $h_j(x^*) = 0$, F continuous and $\eta_k \rightarrow \infty$ then taking the limit for k large enough ($k \rightarrow \infty$), $k \in K$ and dividing each side of the inequality by η_k , we obtain:

$$\lim_{\substack{k \rightarrow \infty \\ k \in K}} \min_{\substack{l=1,\dots,p \\ \forall y \in D}} \left\{ \frac{f_l(x^{k+1}) - f_l(y)}{\eta_k} + \frac{1}{2} \left[\sum_{j=1}^q \left(h_j(x^{k+1}) + \frac{\lambda_j^k}{\eta_k} \right)^2 + \sum_{i=1}^m \left(\max \left(0, g_i(x^{k+1}) + \frac{\mu_i^k}{\eta_k} \right) \right)^2 \right] \right. \\ \left. - \frac{1}{2} \left[\sum_{j=1}^q \left(h_j(y) + \frac{\lambda_j^k}{\eta_k} \right)^2 + \sum_{i=1}^m \left(\max \left(0, g_i(y) + \frac{\mu_i^k}{\eta_k} \right) \right)^2 \right] \right\} \leq 0$$

$$\Rightarrow \min_{\substack{l=1,\dots,p \\ \forall y \in D}} \frac{1}{2} [(\|h(x^*)\| + \|g_+(x^*)\|) - (\|h(y)\| + \|g_+(y)\|)] \leq 0, \text{ because}$$

$$\lim_{\substack{k \rightarrow \infty \\ k \in K}} \frac{f_l(x^{k+1}) - f_l(y)}{\eta_k} = 0 \text{ since } \eta_k \rightarrow \infty$$

$$\min_{\substack{l=1,\dots,p \\ \forall y \in D}} \frac{1}{2} [(\|h(x^*)\| + \|g_+(x^*)\|) - (\|h(y)\| + \|g_+(y)\|)] \leq 0$$

$$\Rightarrow \frac{1}{2} [(\|h(x^*)\| + \|g_+(x^*)\|) - (\|h(y)\| + \|g_+(y)\|)] \leq 0$$

$$\implies \|h(x^*)\| + \|g_+(x^*)\| \leq \|h(y)\| + \|g_+(y)\|, \text{ where } g_+(x) = \max\{0, g(x)\}$$

Since y is a feasible point by definition, therefore $h(y) = 0$ and $g_+(y) = 0$ which implies that

$$\|h(x^*)\| + \|g_+(x^*)\| \leq 0 \implies \|g_+(x^*)\| \leq -\|h(x^*)\| \text{ but } \|g_+(x^*)\| \geq 0,$$

thus $0 \leq \|g_+(x^*)\| \leq -\|h(x^*)\| = 0$, because this inequality exists if and only if $\|h(x^*)\| = 0$.

Therefore, we have $h(x^*) = 0$ and $g(x^*) \leq 0$, consequently x^* is a feasible point. $\square$

Proposition 4. Let X^k be a set of point sequences generated by algorithm 1 and x^* be a limit point of a point sequence $\{x^k\} \in X^k$. Then, x^* is a weakly Pareto optimal point of problem (1).

Proof. Let $K \subseteq \mathbb{N}$, and let $\{x^k\}$ be a sequence of points generated by the Algorithm 1 with x^* as its limit point.

Referring to Proposition 3, we have: $h(x^*) = 0$ and $g(x^*) \leq 0$. The continuation of the proof will proceed by contradiction, assuming that x^* is not a weakly Pareto optimal point for the problem (1).

In this case, there exists $y \in \mathbf{D}$ such that

$$f_l(y) < f_l(x^*) \quad \forall l = 1, \dots, p. \quad (5)$$

Considering Lemma 1 and in accordance with the guidelines of Algorithm 1, we have:

$$\min_{\substack{l=1,\dots,p \\ \forall y \in D}} \left\{ f_l(x^{k+1}) + \frac{\eta_k}{2} \left[\sum_{j=1}^q \left(h_j(x^{k+1}) + \frac{\lambda_j^k}{\eta_k} \right)^2 + \sum_{i=1}^m \left(\max \left(0, g_i(x^{k+1}) + \frac{\mu_i^k}{\eta_k} \right) \right)^2 \right] - f_l(y) \right. \\ \left. - \frac{\eta_k}{2} \left[\sum_{j=1}^q \left(h_j(y) + \frac{\lambda_j^k}{\eta_k} \right)^2 + \sum_{i=1}^m \left(\max \left(0, g_i(y) + \frac{\mu_i^k}{\eta_k} \right) \right)^2 \right] \right\} \leq 0.$$

Each f_l is subject to the constraints g_i for $i = 1, \dots, m$ and h_j for $j = 1, \dots, q$, thus we can rewrite this inequality as follows:

$$\min_{\substack{l=1,\dots,p \\ \forall y \in D}} \left\{ f_l(x^{k+1}) - f_l(y) \right\} \leq -\frac{\eta_k}{2} \left[\sum_{j=1}^q \left[h_j(x^{k+1}) + \frac{\lambda_j^k}{\eta_k} \right]^2 + \sum_{i=1}^m \left[\max \left(0, g_i(x^{k+1}) + \frac{\mu_i^k}{\eta_k} \right) \right]^2 \right] \\ + \frac{\eta_k}{2} \left[\sum_{j=1}^q \left[h_j(y) + \frac{\lambda_j^k}{\eta_k} \right]^2 + \sum_{i=1}^m \left[\max \left(0, g_i(y) + \frac{\mu_i^k}{\eta_k} \right) \right]^2 \right].$$

After simplification, we have

$$\min_{\substack{l=1,\dots,p \\ \forall y \in D}} \left\{ f_l(x^{k+1}) - f_l(y) \right\} \leq -\frac{1}{2\eta_k} \left[\sum_{j=1}^q \left[\eta_k h_j(x^{k+1}) + \lambda_j^k \right]^2 + \sum_{i=1}^m \left[\max \left(0, \eta_k g_i(x^{k+1}) + \mu_i^k \right) \right]^2 \right] \\ + \frac{1}{2\eta_k} \left[\sum_{j=1}^q \left[\eta_k h_j(y) + \lambda_j^k \right]^2 + \sum_{i=1}^m \left[\max \left(0, \eta_k g_i(y) + \mu_i^k \right) \right]^2 \right].$$

We now have two cases to consider: $\eta_k \rightarrow \infty$ and $\eta_k \rightarrow \bar{\eta}$. First, let us consider the case $\eta_k \rightarrow \infty$, we have:

$$\begin{aligned} \min_{\substack{l=1,\dots,p \\ \forall y \in D}} \left\{ f_l(x^{k+1}) - f_l(y) \right\} &\leq -\frac{1}{2\eta_k} \left[\sum_{j=1}^q \left[\eta_k h_j(x^{k+1}) + \lambda_j^k \right]^2 + \sum_{i=1}^m \left[\max \left(0, \eta_k g_i(x^{k+1}) + \mu_i^k \right) \right]^2 \right] \\ &\quad + \frac{1}{2\eta_k} \left[\sum_{j=1}^q \left[\eta_k h_j(y) + \lambda_j^k \right]^2 + \sum_{i=1}^m \left[\max \left(0, \eta_k g_i(y) + \mu_i^k \right) \right]^2 \right] \\ \Rightarrow \min_{\substack{l=1,\dots,p \\ \forall y \in D}} \left\{ f_l(x^{k+1}) - f_l(y) \right\} &\leq \frac{1}{2\eta_k} \left[\sum_{j=1}^q \left[\eta_k h_j(y) + \lambda_j^k \right]^2 + \sum_{i=1}^m \left[\max \left(0, \eta_k g_i(y) + \mu_i^k \right) \right]^2 \right]. \end{aligned}$$

Indeed, $\eta_k h_j(y) + \lambda_j^k$ and $\max(0, \eta_k g_i(y) + \mu_i^k)$ can respectively be equal to 0 when k is sufficiently large (in the case where $h_j(y) = 0$, $g_i(y) < 0$, given that $\{\mu_i^k\}$ and $\{\lambda_j^k\}$ are bounded), or they can respectively be equal to μ_i^k and λ_j^k for any value of k (if $h_j(y) = 0$ and $g_i(y) = 0$, since μ_i^k and λ_j^k are non-negative, ensuring that these expressions can never be less than zero). Therefore, taking the limit as k tends to infinity, with $k \in K$, and keeping in mind that $\{\mu_i^k\}$ and $\{\lambda_j^k\}$ are bounded, we obtain:

$$\min_{\substack{l=1,\dots,p \\ \forall y \in D}} \left\{ f_l(x^*) - f_l(y) \right\} \leq 0,$$

which contradicts inequality (5).

Now consider the case where η_k converges to η^* . According to the instructions of Algorithm 1, V^{k+1} tends to 0 and $h_j(x^{k+1}) = 0$, meaning that $\frac{\mu_i^k}{\eta_k}$ tends to 0 as $k \rightarrow +\infty$ with $k \in K$. This implies that for all indices i and j such that $g_i(x^*) < 0$ and $h_j(x^*) = 0$, the values of V^k become negligible as k increases and $h(x^k) = 0$. In particular, for these indices i , we have $\mu_i^k \rightarrow 0$, as $k \rightarrow +\infty$, with $k \in K$. Furthermore, since by definition we have $h_j(y) = 0$ and $g_i(y) \leq 0$. Consequently, we obtain,

$$\begin{aligned} &-\frac{1}{2\eta_k} \left[\sum_{h_j(x^*)=0} \left[\eta_k h_j(x^{k+1}) + \lambda_j^k \right]^2 + \sum_{g_i(x^*)<0} \left[\max \left(0, \eta_k g_i(x^{k+1}) + \mu_i^k \right) \right]^2 \right] \\ &\quad + \frac{1}{2\eta_k} \left[\sum_{h_j(x^*)=0} \left[\eta_k h_j(y) + \lambda_j^k \right]^2 + \sum_{g_i(x^*)<0} \left[\max \left(0, \eta_k g_i(y) + \mu_i^k \right) \right]^2 \right] \\ &\leq -\frac{1}{2\eta_k} \left[\sum_{h_j(x^*)=0} \left[\lambda_j^k \right]^2 \right] + \frac{1}{2\eta_k} \left[\sum_{h_j(x^*)=0} \left[\eta_k h_j(y) + \lambda_j^k \right]^2 + \sum_{g_i(x^*)<0} \left[\max \left(0, \eta_k g_i(y) + \mu_i^k \right) \right]^2 \right] \\ &\leq \lim_{\substack{k \rightarrow \infty \\ k \in K}} \left(-\frac{1}{2\eta^*} \left[\sum_{h_j(x^*)=0} \left[\lambda_j^k \right]^2 \right] + \frac{1}{2\eta^*} \left[\sum_{h_j(x^*)=0} \left[\lambda_j^k \right]^2 \right] + \frac{1}{2\eta^*} \left[\sum_{g_i(x^*)<0} \left[\max \left(0, \eta^* g_i(y) \right) \right]^2 \right] \right) \\ &\leq \lim_{\substack{k \rightarrow \infty \\ k \in K}} \frac{1}{2\eta^*} \left[\sum_{g_i(x^*)<0} \left[\max \left(0, \eta_k g_i(y) \right) \right]^2 \right] = 0. \end{aligned}$$

Now consider $\{i/g_i(x^*) = 0\}$. We have

$$-\frac{1}{2\eta_k} \left[\sum_{h_j(x^*)=0} [\eta_k h_j(x^{k+1}) + \lambda_j^k]^2 + \sum_{g_i(x^*)=0} [\max(0, \eta_k g_i(x^{k+1}) + \mu_i^k)]^2 \right] \\ + \frac{1}{2\eta_k} \left[\sum_{h_j(x^*)=0} [\eta_k h_j(y) + \lambda_j^k]^2 + \sum_{g_i(x^*)=0} [\max(0, \eta_k g_i(y) + \mu_i^k)]^2 \right]$$

Using $g_i(y) \leq 0$, we obtain

$$\lim_{\substack{k \rightarrow \infty \\ k \in K}} \left(-\frac{1}{2\eta^*} \left[\sum_{g_i(x^*)=0} [\max(0, \mu_i^k)]^2 \right] + \frac{1}{2\eta^*} \left[\sum_{g_i(x^*)=0} [\max(0, \eta^* g_i(y) + \mu_i^k)]^2 \right] \right) \\ \leq \lim_{\substack{k \rightarrow \infty \\ k \in K}} \left(-\frac{1}{2\eta^*} \left[\sum_{g_i(x^*)=0} [\max(0, \mu_i^k)]^2 \right] + \frac{1}{2\eta^*} \left[\sum_{g_i(x^*)=0} [\max(0, \mu_i^k)]^2 \right] \right) = 0.$$

Putting all of this together, we again obtain

$$\min_{\substack{l=1, \dots, p \\ \forall y \in D}} \{f_l(x^*) - f_l(y)\} \leq 0$$

which contradicts inequality (5). Therefore, x^* is a weakly Pareto optimal point of the problem (1). $\square$

3.2. Solving the subproblem of Algorithm 1. In this section, we present a stochastic algorithm that we use to solve the subproblem of the Algorithm 1. This is a multi-objective algorithm of the Black Widow Optimization (BWO) algorithm called *MOBWO*. It is an extension to the multi-objective case of the *BWO* algorithm. This optimization algorithm is inspired by social behaviors (such as group swimming, the dynamics of beluga whale life, communication, and information sharing) and the hunting strategies of beluga whales. The optimization process of BWO consists of three main phases: the exploration phase, which simulates swimming behavior; the exploitation phase, which imitates predation behavior; and the whale's fall phase inspired by beluga whale's fall. The principle of BWO is as follows:

Exploration: The whales explore the search space to discover new solutions, mimicking their group swimming behavior, which can be mathematically modeled as follows.

$$\begin{cases} X_{i,j}^{t+1} = X_{i,p_j}^t + (X_{r,p_1}^t - X_{i,p_j}^t) (1 + r_1) \sin(2\pi r_2), & j \text{ odd} \\ X_{i,j}^{t+1} = X_{i,p_j}^t + (X_{r,p_1}^t - X_{i,p_j}^t) (1 + r_1) \sin(2\pi r_2), & j \text{ even} \end{cases} \quad (6)$$

where t represents the current iteration number or candidate in the optimization process, $X_{i,j}^{t+1}$ indicates the new or updated position of the i^{th} beluga whale in the search space, specifically in the j^{th} dimension. $p_j (j = 1, 2, \dots, d)$ is a random number selected from d dimensions and represents a specific dimension in the search space. X_{i,p_j}^t represents the current position of the i^{th} beluga whale in the dimension defined by p_j at iteration t . The algorithm uses both the current position of the i^{th} whale

and that of another whale (the r^{th}) to update the position of the i^{th} whale. The r^{th} whale is randomly selected from within the population. Here, r_1, r_2 are random numbers between $(0,1)$, and $\sin(2\pi r_2)$ and $\cos(2\pi r_2)$ indicate that the fins of beluga whales are oriented towards the water's surface.

Exploitation: Once a food source (optimal solution) is identified, the whales focus on that area to refine their search, mathematically modeled by the following equation.

$$X_i^{t+1} = r_3 X_{best}^t - r_4 X_i^t + C_1 \cdot L_F \cdot (X_r^t - X_i^t) \quad (7)$$

where X_i^t and X_r^t are the current positions of the i^{th} beluga and a randomly chosen beluga, allowing for diversity in the search process. X_i^{t+1} is the new position of the i^{th} whale, X_{best}^t is the best position among the whales, r_3, r_4 are random numbers between $(0,1)$, C_1 is the strength of the random jump that measures the intensity of the Lévy flight performed by the whale, and L_F is the Lévy flight function given by

$$L_F = 0.05 \times \frac{u \times \sigma}{|\nu|^{1/\beta}}.$$

Whale fall: This mechanism allows for escaping local optima by simulating the death of certain whales, thereby introducing diversity into the population. The algorithm also uses balance factors and adaptive fall probabilities, allowing for dynamic control between exploration and exploitation, mathematically represented as follows.

$$X_i^{t+1} = r_5 X_i^t - r_6 X_r^t + r_7 X_{step}. \quad (8)$$

where r_5, r_6 , and r_7 are random numbers between $(0,1)$, and X_{step} is the step size of the whale that determines how far this new position will be from the current position of the whale. For more details, see [29].

The *BWO* algorithm can transition from exploration to exploitation based on the balance factor $B_f = B_0(1 - t/2t_{max})$, where t is the current iteration, t_{max} is the maximum number of iterations, and $B_0 \in (0,1)$ varies randomly at each iteration.

The probability of whale diving (W_f) is determined by:

$$W_f = 0.1 - 0.05 \frac{t}{t_{max}}.$$

The pseudo-code of the BWO algorithm (see [29]) is as follows:

Algorithm 2: BWO Algorithm

-
- 1 **Step 1: Initialization: Input: ;**
 - 2 • Determine the algorithm parameters: population size (n), maximum number of iterations (t_{max}), balance factor (B_f), drop probability (W_f), position update parameters (C_1, C_2);
 - 3 • Randomly generate the initial positions X^0 of all whales in the search space;
 - 4 • Evaluate the fitness values for these positions;
 - 5 **Step 2: Update during the Exploration and Exploitation Phases;;**
 - 6 • For each whale, determine whether it enters the exploration or exploitation phase based on a balance factor B_f ;
 - 7 • If $B_f > 0.5$, the whale enters the exploration phase, and its position is updated using equation (6);
 - 8 otherwise, if $B_f < 0.5$, it enters the exploitation phase, and its position is updated using equation (7);
 - 9 • The fitness values of the new positions are calculated and sorted to find the optimal result for the current iteration;
 - 10 **Step 3: Update during the Beluga Drop Phase;;**
 - 11 • Calculate the drop probability W_f at each iteration;
 - 12 • Update the positions of the whales based on this probability using equation (8);
 - 13 **Step 4: Check the Stopping Condition: ;**
 - 14 • If the current iteration exceeds the maximum number of iterations, the BWO algorithm stops;
 - 15 • Otherwise, repeat from step 2.
-

The *MOBWO* algorithm integrates elitism mechanisms based on Non-Dominated Solutions (NDR) and Crowding Distance (CD) to maintain diversity among solutions on the Pareto front. The NDR process consists of three steps: identification of non-dominated solutions, application of the NDR approach, and calculation of the non-dominated rank for each non-dominated solution. The NDR divides the Pareto front into two sets: the first includes non-dominated solutions that are not dominated by any other solution, while the second contains solutions that are dominated by at least one other. Crowding Distance (CD) is used to preserve diversity among non-dominated solutions on the Pareto front, following a principle similar to the NSGA-II algorithm, and can be calculated as follows:

$$CD_j^i = \frac{obj_j^{i+1} - obj_j^{i-1}}{obj_j^{max} - obj_j^{min}}$$

where obj_j is the j^{th} objective function, and $obj_j^{\max}$ and $obj_j^{\min}$ represent the maximum and minimum values of the objective function obj_j .

The first step of MOBWO is to determine the parameters of BWO, the maximum number of iterations, the stopping criteria, and the population size. An initial parent population X^0 is randomly created in the feasible solution space, and each objective function in the vector F is evaluated for the members of X^0 . The Non-Dominance and CD strategies are then applied to the population X^0 . These techniques generate a new population X^j using the MOBWO Algorithm, which is combined with X^0 to create an intermediate population X^i . The population X^i is sorted based on an elitism criterion that considers the Non-Dominance of solutions, taking into account information related to NDR and CD .

The best solutions among the best are then adjusted to form a new parent population. The algorithm repeats this procedure until the stopping condition is met. The pseudo-code of the *MOBWO* algorithm is as follows:

Algorithm 3: Multi-Objective BWO (MOBWO)

Input : BWO Parameters: α, β , etc.

Input : Initialization of solution positions $X^0 : i = 1, \dots, N$

- 1 Evaluate the population in the objective space;
 - 2 Sort the population based on the elitist non-dominated sort method and find the Non-Dominated Rank (NDR) and fronts;
 - 3 For each front, calculate the Crowding Distance (CD);
// Update the population X^j
 - 4 Update the solutions X^j using Algorithm 2;
 - 5 Combine X^0 and X^j to generate X^i , such that $X^i = X^0 \cup X^j$;
 - 6 Perform **Step 1** on X^i ;
 - 7 Sort X^i based on NDR and CD;
 - 8 Replace the initial population X^0 with the top members of the new population X^i ;
-

Since the solution of each subproblem (4) is obtained using a stochastic method that generates a population of points at each iteration, the position of each point in the population is considered a stochastic vector. The convergence analysis of the algorithm's properties can be based on probability theory.

The convergence analysis presented here is similar to that of [12], applied to a specific class of evolutionary strategies for unconstrained problems. Probability theory from the perspective of measure theory is used.

Let

$$\mathcal{L}^k(x) = \mathcal{L}(x, \lambda^k, \mu^k, \eta_k) \quad (9)$$

We make the following assumptions:

- (i) The functions $f_l, l = 1, \dots, p, h_j, j = 1, \dots, q$, and $g_i, i = 1, \dots, m$, are continuous.
- (ii) The search space $\mathbf{D}$ is a Lebesgue measurable space.
- (iii) $\mathcal{L}^k: \mathbf{D} \rightarrow \mathbb{R}^p$ is a measurable function.
- (iv) There exists at least one Pareto-optimal solution with respect to $\mathcal{L}^k$ in the search space $\mathbf{D}$.
- (v) If $m(S)$ denotes the Lebesgue measure of a set S , then for any positive number $a > 0, m(S) > 0$, where $S = \{x : \max_{j=1, \dots, p} (\mathcal{L}_j^k(y) - \mathcal{L}_j^k(x)) + a \geq 0, \forall y \in \mathbf{D}\}$.

Definition 5. Let x and y be two points in $\mathbf{D}$. The point y is said to improve with respect to x if, for $\epsilon > 0$, the following condition is satisfied:

$$\max_{l=1, \dots, p} \left\{ \mathcal{L}_l^k(x) - \mathcal{L}_l^k(y) \right\} \leq \max_{l=1, \dots, p} \left\{ \mathcal{L}_l^k(x) - \mathcal{L}_l^k(z) \right\} - \epsilon, \quad \forall z \in \mathbf{D}.$$

Remark 1. Two random numbers r_1 and r_2 in $(0, 1)$ are used to enhance the random operators during the exploration phase, and r is a randomly selected beluga whale.

During the exploitation phase of the BWO, the Levy flight function L_F is introduced to improve convergence, where ν and v are normally distributed random numbers.

To model the behavior of whales falling at each iteration during the falling phase, a falling probability W_f is applied among the individuals of the population to simulate small changes in the groups. r_5, r_6 , and r_7 are random numbers in $(0, 1)$. In the falling phase of the whales, the falling probability W_f starts at 0.1 and decreases linearly to 0.05 by the last iteration, remaining positive as long as $t \leq t_{max}$. A positive W_f indicates a non-zero probability of whales falling at each iteration, enabling changes in their positions and contributing to the dynamics of the MOBWO algorithm. A falling probability implies that individuals are updated during the falling phase, creating an unconstrained distribution of positions that allows whales to change their positions randomly. The algorithm is less likely to get stuck in local minima, thus balancing exploration (searching for new solutions) and exploitation (refining existing solutions). This allows the algorithm to adapt and evolve over iterations, increasing the chances of finding Pareto-optimal solutions. All of the above indicates a non-zero density in the search space.

The position update equations for the whales, which take into account the best solution found and the positions of other whales, promote convergence by allowing the whales to move closer to the Pareto-optimal solutions over iterations. Therefore, Pareto-optimal solutions represent the best possible trade-offs, and the Lagrangian of these solutions is generally better (lower) compared to non-Pareto-optimal solutions in the trade-off space. Hence, for any $x^* \in \mathbf{D}$, it is an ϵ -Pareto-optimal solution of MOBWO, and for any feasible solution $x \in \mathbf{D}$, the Lagrangian satisfies: $\mathcal{L}(x^*, \lambda^k, \mu^k, \eta_k) \leq \mathcal{L}(x, \lambda^k, \mu^k, \eta_k)$.

Suppose that z is an ϵ -Pareto-optimal solution of problem (1) generated by MOBWO. Then the point z is a solution to the subproblem (4) of minimization from Algorithm 3 if z is also an ϵ -Pareto-optimal solution generated by MOBWO.

Theorem 5. Let $\mathcal{L}^k(x) = (\mathcal{L}_1^k(x), \mathcal{L}_2^k(x), \dots, \mathcal{L}_p^k(x))$ be a vector of objective functions defined over the feasible space $\mathbf{D}$. If the pair $(\mathcal{L}^k, \mathbf{D})$ satisfies assumptions (i)-(iv), then for $\epsilon > 0$, the MOBWO algorithm converges to an ϵ -Pareto-optimal solution for the subproblem (4) with a probability of one.

Proof. Let z be a solution to the subproblem (4). If z is at least one of the ϵ -Pareto optimal solutions of MOBWO and using Lemma 1, we have $y \in \mathbf{D}$ such that

$$\max_{\substack{l=1,\dots,p \\ y \in \mathbf{D}}} \{f_l(y) - f_l(z)\} > 0 \implies \max_{\substack{j=1,\dots,p \\ y \in \mathbf{D}}} \{f_l(y) - f_l(z)\} + \epsilon > 0$$

with $\epsilon > 0$.

Thus,

$$\max_{\substack{l=1,\dots,p \\ y \in \mathbf{D}}} \{\mathcal{L}_l^k(y) - \mathcal{L}_l^k(z)\} > 0 \implies \max_{\substack{l=1,\dots,p \\ y \in \mathbf{D}}} \{\mathcal{L}_l^k(y) - \mathcal{L}_l^k(z)\} + \epsilon > 0.$$

We begin by assuming that the population consists of a single point ($n = 1$) and that ϵ represents a precision limit error. In this case, z is a solution of MOBWO if

$$\max_{\substack{l=1,\dots,p \\ y \in \mathbf{D}}} \{\mathcal{L}_l^k(y) - \mathcal{L}_l^k(z)\} + \epsilon > 0.$$

It is crucial to note that when the algorithm proceeds to the next iteration and the point z , as a candidate solution, improves upon the current solution x , this implies that

$$\max_{\substack{l=1,\dots,p \\ \forall y \in \mathbf{D}}} \{\mathcal{L}_l^k(y) - \mathcal{L}_l^k(z)\} \leq \max_{\substack{l=1,\dots,p \\ \forall y \in \mathbf{D}}} \{\mathcal{L}_l^k(y) - \mathcal{L}_l^k(x)\} + \epsilon.$$

Each update contributes to the improvement of the solution. Thus, an ϵ -Pareto solution can be achieved from any initial point $x \in \mathbf{D}$, after a finite number of iterations (updates) where z , the candidate solution for the next iteration, improves upon x from the current iteration. At each iteration of MOBWO, a parent $z \in X^k$ generates an offspring $y \in X^{k+1}$, where y represents the candidate solution that improves upon z , justifying the decrease of the following by ϵ :

$$\max_{\substack{l=1,\dots,p \\ \forall x \in \mathbf{D}}} \{\mathcal{L}_l^k(x) - \mathcal{L}_l^k(y)\} \leq \max_{\substack{l=1,\dots,p \\ \forall x \in \mathbf{D}}} \{\mathcal{L}_l^k(x) - \mathcal{L}_l^k(z)\} - \epsilon.$$

We replace or update the parent with its offspring. Thus, y becomes an ϵ -Pareto optimal solution.

For the rest of the proof, we will show that z is not an ϵ -Pareto optimal solution, which implies that, $\forall y \in \mathbf{D}$,

$$\max_{\substack{l=1,\dots,p \\ \forall y \in \mathbf{D}}} \{\mathcal{L}_l^k(y) - \mathcal{L}_l^k(z)\} > \max_{\substack{l=1,\dots,p \\ \forall y \in \mathbf{D}}} \{\mathcal{L}_l^k(y) - \mathcal{L}_l^k(x)\} + \epsilon,$$

with a probability of one. The candidate solution z does not improve upon the current solution y^* after a finite number of iterations.

$$\max_{\substack{l=1,\dots,p \\ \forall x \in \mathbf{D}}} \left\{ \mathcal{L}_l^k(x) - \mathcal{L}_l^k(y^*) \right\} > \max_{\substack{l=1,\dots,p \\ \forall x \in \mathbf{D}}} \left\{ \mathcal{L}_l^k(x) - \mathcal{L}_l^k(z) \right\} + \epsilon.$$

let's consider

$$\begin{aligned} a &= \frac{1}{2} \left[\max_{\substack{l=1,\dots,p \\ \forall x \in \mathbf{D}}} \left\{ \mathcal{L}_l^k(x) - \mathcal{L}_l^k(y^*) \right\} - \max_{\substack{l=1,\dots,p \\ \forall x \in \mathbf{D}}} \left\{ \mathcal{L}_l^k(x) - \mathcal{L}_l^k(z) \right\} - \epsilon \right] > 0 \\ \implies \max_{\substack{l=1,\dots,p \\ \forall x \in \mathbf{D}}} \left\{ \mathcal{L}_l^k(x) - \mathcal{L}_l^k(y^*) \right\} - \epsilon &= \max_{\substack{l=1,\dots,p \\ \forall x \in \mathbf{D}}} \left\{ \mathcal{L}_l^k(x) - \mathcal{L}_l^k(z) \right\} + 2a \\ &> \max_{\substack{l=1,\dots,p \\ \forall x \in \mathbf{D}}} \left\{ \mathcal{L}_j^k(x) - \mathcal{L}_l^k(z) \right\} + a, \end{aligned}$$

We can thus define the set S as follows:

$$S = \left\{ \nu \in \mathbf{D} \mid \max_{\substack{l=1,\dots,p \\ \forall z \in \mathbf{D}}} \left(\mathcal{L}_l^k(z) - \mathcal{L}_l^k(\nu) \right) + \epsilon \right\} \subset \mathbf{D}.$$

Condition (v) of the above hypothesis implies that $m(S) > 0$.

To conclude the proof, let us proceed by contradiction. Suppose there exists an iteration where x is retained (i.e., it is not updated) such that the next iteration occurs infinitely many times.

The probability $W_f > 0$ allows whales to change position with non-zero density, as noted, to improve solutions. This is used to generate y , and $m(S) > 0$ implies that $P = \text{prob}[y \in S] > 0$. This is also true for any other distribution whose density function is non-zero on the set $\mathbf{D}$. At the k -th iteration, the probability that y^* is not updated (i.e., retained) in the next iteration (y does not improve upon y^*) is given by

$$q_k = \text{prob} \left[\max_{\substack{l=1,\dots,p \\ \forall x \in \mathbf{D}}} \left\{ \mathcal{L}_l^k(x) - \mathcal{L}_l^k(y) \right\} \geq \max_{\substack{l=1,\dots,p \\ \forall x \in \mathbf{D}}} \left\{ \mathcal{L}_l^k(x) - \mathcal{L}_l^k(y^*) \right\} - \epsilon \right],$$

and

$$\begin{aligned} \max_{\substack{l=1,\dots,p \\ \forall x \in \mathbf{D}}} \left\{ \mathcal{L}_l^k(x) - \mathcal{L}_l^k(z) \right\} + 2a &> \max_{\substack{l=1,\dots,p \\ \forall x \in \mathbf{D}}} \left\{ \mathcal{L}_l^k(x) - \mathcal{L}_l^k(z) \right\} + a, \\ \implies q_k &< \text{prob} \left[\max_{\substack{l=1,\dots,p \\ \forall x \in \mathbf{D}}} \left\{ \mathcal{L}_l^k(x) - \mathcal{L}_l^k(y) \right\} > \max_{\substack{j=1,\dots,p \\ \forall x \in \mathbf{D}}} \left\{ \mathcal{L}_l^k(x) - \mathcal{L}_l^k(z) \right\} + a \right], \\ &= \text{prob}[y \in \mathbf{D} \setminus S] = 1 - \text{prob}[y \in S] = 1 - P < 1. \end{aligned}$$

When x is not updated as assumed, P remains unchanged throughout the iterative process. The probability that no update occurs during k_1 consecutive iterations, with each iteration being independent, is given by:

$$q_1 q_2 \dots q_{k_1} = \prod_{k=1}^{k_1} q_k \leq (1 - P)^{k_1},$$

thus

$$\lim_{k_1 \rightarrow +\infty} \prod_{k=1}^{k_1} q_k = 0, \text{ since } 1 - P < 1.$$

Consequently, x is not an ϵ -Pareto optimal solution, ensuring that an update will occur within a finite number of iterations. Moreover, it is guaranteed that a solution y improving upon the current solution y^* will be generated. Thus, an ϵ -Pareto optimal solution is reached at some point.

Let us now address the case where $n > 1$. When a population-based method of size n is used, with solutions X in the search space, we have $\mathcal{X} \in \mathbf{D}^n$. In this context, let us redefine S as the set of all populations $\mathcal{Y} \in \mathbf{D}^n$ whose best point reflects the optimal position of the whales, with a bounded error relative to the ϵ -Pareto optimal solution. Each function $\mathcal{L}_l^k$, for $l = 1, \dots, p$, has a fitness value with a bounded error ϵ relative to the solution, which allows us to define the function $\mathcal{L}^k$ with an ϵ -Pareto optimal solution.

$$S = \left\{ \mathcal{Y} \in \mathbf{D}^n \mid \max_{\substack{l=1, \dots, p \\ \forall W \in \mathbf{D}}} (\mathcal{L}_l^{k,k}(W) - \mathcal{L}_l^{k,k}(Y)) + \epsilon \right\} \subset \mathbf{D}^n,$$

Let $P' = \text{prob} [X^{k+1} \in S] > 0$, where X^{k+1} is the population of points at the next iteration $k + 1$.

We will now show that if X^k represents the best position of the whale population at step k , then X^k is not an ϵ -Pareto optimal solution with probability one. X^{k+1} is the candidate solution of the next population, and X^{k+1} improves upon X^k in a finite number of iterations. Let us proceed by contradiction. Suppose that

$$\max_{\substack{l=1, \dots, p \\ \forall X' \in \mathbf{D}}} (\mathcal{L}_l^k(X') - \mathcal{L}_l^k(X^{k+1})) > \max_{\substack{l=1, \dots, p \\ \forall X' \in \mathbf{D}}} (\mathcal{L}_l^k(X') - \mathcal{L}_l^k(X^k)) - \epsilon$$

is true for infinitely many iterations.

Let

$$q_k = \text{prob} \left[\max_{\substack{l=1, \dots, p \\ \forall X' \in \mathbf{D}}} (\mathcal{L}_l^k(X') - \mathcal{L}_l^k(X^{k+1})) \geq \max_{\substack{j=1, \dots, p \\ \forall X' \in \mathbf{D}}} (\mathcal{L}_l^k(X') - \mathcal{L}_j^k(X^k)) - \epsilon \right].$$

Thus,

$$\begin{aligned} q_k &< \text{prob} \left[\max_{\substack{l=1, \dots, p \\ \forall X' \in \mathbf{D}}} (\mathcal{L}_l^k(X') - \mathcal{L}_l^k(X^{k+1})) > \max_{\substack{l=1, \dots, p \\ \forall X' \in \mathbf{D}}} (\mathcal{L}_l^k(X') - \mathcal{L}_l^k(X^k)) + a \right] \\ &= \text{prob} [X^{k+1} \in \mathbf{D}^n \setminus S] \\ &= 1 - \text{prob} [X^{k+1} \in S] \\ &= 1 - P' < 1. \end{aligned}$$

Using a reasoning similar to the case $n = 1$, we have:

$$q_1 q_2 \dots q_n = \prod_{k=1}^n q_k \leq (1 - P')^n \text{ and } \lim_{n \rightarrow +\infty} \prod_{k=1}^n q_k = 0.$$

We can conclude that if X^k is not a solution, then a better point X^{k+1} , which improves upon the best position of the whales X^k , is guaranteed to be generated in a finite number of iterations. Therefore, after a finite number of iterations, a population X as an ϵ -Pareto optimal solution is reached. $\square$

3.3. Multiobjective model of a portfolio problem. We consider the multiobjective model of a portfolio problem as proposed in [23], which is given by:

$$\left\{ \begin{array}{l} \min \sum_{i=1}^n \sum_{j=1}^n x_i x_j \sigma_{ij}, \\ \max \sum_{j=1}^n \mu_j x_j, \\ \text{subject to} \\ \sum_{j=1}^n x_j = 1, \\ 0 \leq x_j \leq 1, \quad j = 1, \dots, n. \end{array} \right. \quad (10)$$

In practice, to ensure that investments below a given threshold ξ_j , deemed too low and insignificant, we add nonlinear constraints as defined by [5]:

$$\xi_j x_j - x_j^2 \leq 0, \quad \forall j = 1, \dots, n. \quad (11)$$

This constraint ensures that the investment x_j in asset j does not fall within the interval $(0, \xi_j)$, implicitly acting as a cardinality constraint.

The multiobjective model proposed in this work can be summarized as follows:

$$\left\{ \begin{array}{l} \min \sum_{i=1}^n \sum_{j=1}^n x_i x_j \sigma_{ij}, \\ \max \sum_{j=1}^n \mu_j x_j, \\ \text{subject to} \\ \sum_{j=1}^n x_j = 1, \\ \xi_j x_j - x_j^2 \leq 0, \quad j = 1, \dots, n, \\ 0 \leq x_j \leq 1, \quad j = 1, \dots, n. \end{array} \right. \quad (12)$$

Here: - n is the total number of risky assets available to the investor; - x_j represents the fraction of capital invested in asset j ; - μ_j is the expected average return of asset j ; - σ_{ij} is the covariance of the returns of assets i and j .

The constraint $\sum_{j=1}^n x_j = 1$ is crucial, as it ensures that the investor allocates the entire budget across the selected assets, i.e., 100% of the capital is invested.

The purpose of the constraint $\xi_j x_j - x_j^2 \leq 0$ is to ensure that for each asset j , the investment x_j is not too low by imposing a minimum investment threshold. This implies that if one invests in an asset, at least a certain amount ξ_j must be allocated to avoid insignificant investments in assets. This helps concentrate resources on assets with higher return potential, enhancing the overall portfolio performance and reducing the risks associated with overly fragmented investments. The choice of ξ_j can be:

- uniformly random values within a specified range;
- values derived from historical statistics (e.g., using the mean or standard deviation to define ξ_j);
- values based on a normal distribution (useful if most values should be near a certain mean with a few extremes);
- specific criteria for each asset (based on past performance or risk levels).

In this model, we address a bi-objective problem where the first objective is to minimize the portfolio variance, and the second is to maximize the portfolio return. Note that the expected portfolio return, denoted μ_P , is defined as:

$$\mu_P = \sum_{j=1}^n x_j \mu_j.$$

The return measures the relative appreciation or depreciation of an asset or portfolio over a given period, while the risk, representing the uncertainty about the future value of the asset, must be minimized. The first objective of model (12) represents the portfolio's overall risk. It is essential to note that portfolio optimization considers portfolio-wide risk rather than individual asset risks.

The covariance σ_{ij} and the mean returns are derived from historical data. The daily return R of each asset is calculated using the formula:

$$R_{t_1} = \frac{P_{t_1} - P_{t_0}}{P_{t_0}},$$

where P_{t_0} is the asset price at time t_0 (e.g., yesterday) and P_{t_1} is the asset price at time t_1 (e.g., today).

The average return of asset i , $\bar{R}_i$, is given by:

$$\bar{R}_i = \frac{1}{T} \sum_{t=1}^T R_{i,t},$$

where T is the number of periods, and $R_{i,t}$ is the return of asset i at time t . The covariance between the returns R_i and R_j of assets is calculated as:

$$\sigma_{ij} = \text{cov}(R_i, R_j) = \frac{1}{T} \sum_{t=1}^T (R_{i,t} - \bar{R}_i) (R_{j,t} - \bar{R}_j).$$

The expression for μ_j is given by:

$$\mu_j = \frac{1}{T} \sum_{t=1}^T R_{j,t}.$$

4. APPLICATION AND NUMERICAL SIMULATION

In this section, we apply the portfolio model (12) presented above to a set of 10 assets, 50 assets, and 100 assets. We took the daily price of the adjusted closing value over 2940 business days. The data we used corresponds to the date range from January 14, 2013, to September 19, 2024. We first selected 100 companies that have data during this period from the New York Stock Exchange list available at: <http://www.nasdaq.com/screening/company-list.aspx>; then, we extracted the daily prices of the adjusted closing value from Yahoo! Finance historical data for these companies at: <http://finance.jasonstrimpel.com>.

Once the data was collected, we calculated σ_{ij} and μ_i and then proceeded to the numerical simulation. The list of the 100 companies (abbreviated names) used is as follows: AAPL, ABBV, ABT, ACN, ADBE, AMAT, AMD, AMGN, AMZN, ASML, AVGO, AXP, AZN, BAC, BHP, BKNG, BLK, BRK-A, BRK-B, BSX, CAT, CMCSA, COST, CRM, CSCO, CVX, DHR, DIS, ELV, ETN, FMX, GE, GOOG, GOOGL, GS, HD, HDB, HON, IBM, INTU, ISRG, JNJ, JPM, KO, LIN, LLY, LMT, LOW, LYG, MA, MCD, META, MRK, MS, MSFT, MUFG, NEE, NFLX, NKE, NOW, NVDA, NVO, NVS, PEP, PFE, PG, PGR, PLD, PM, QCOM, REGN, RTX, RY, RYAAY, SAP, SCHW, SHEL, SNY, SONY, SPGI, SYK, T, TJX, TM, TMO, TMUS, TSLA, TSM, TTE, TXN, UL, UNH, UNP, V, VRTX, VZ, WFC, WTM, XOM.

We used Matlab software to implement the proposed algorithms considering the following parameters: $\lambda^0 = 0.8147$, $\mu^0 = 1.rand(1, 100)$, $\eta_0 = 1$, $\maxIter = 10000$.

For each category of problems (10 assets, 50 assets and 100 assets), we have n variables and n inequality constraints along with a single equality constraint (excluding non-negativity constraints). For the inequality constraints that prevent investing small percentages of capital in a given asset we take $\xi_j \in \{\dots; 0.01; 0.02; \dots; 0.1\}$.

We compare the results obtained with the NSGA-II method. The parameters used for the NSGA-II method are as follows: number of generations is 10000 with a population size of 100. Both methods were compiled on an HP laptop: Intel(R) Core(TM) i7-7600U CPU @ 2.80GHz 2.90 GHz RAM: 16 G₀.

Figure 1 shows the Pareto fronts obtained with the proposed method and the NSGA-II algorithm for 10 assets while Figure 2 presents the Pareto fronts for 50 assets and Figure 3 for 100 assets. Figure 1 shows that the solutions obtained by Algorithm 1 are better distributed along the Pareto front compared to the solutions from NSGA-II. Thus the Pareto front of Algorithm 1 allows for exploring multiple trade-offs between objectives compared to NSGA-II. The solutions from Algorithm 1 are significantly better than those from NSGA-II regarding portfolio return represented by objective 1 and portfolio risk represented by objective 2.

For Figure 2 and Figure 3, we observe that the risks of the solutions proposed by Algorithm 1 are much lower than those proposed by NSGA-II. Additionally for each solution from Algorithm 1, the return is significantly higher than the risk unlike that of NSGA-II where the risk is greater than or almost equal to the return of each solution.

We generally observe that in each case (10 assets or 50 assets or 100 assets), Algorithm 1 produces results where return exceeds risk unlike NSGA-II method where risk often exceeds return or is almost equal. This shows that Algorithm 1 outperforms NSGA-II method.

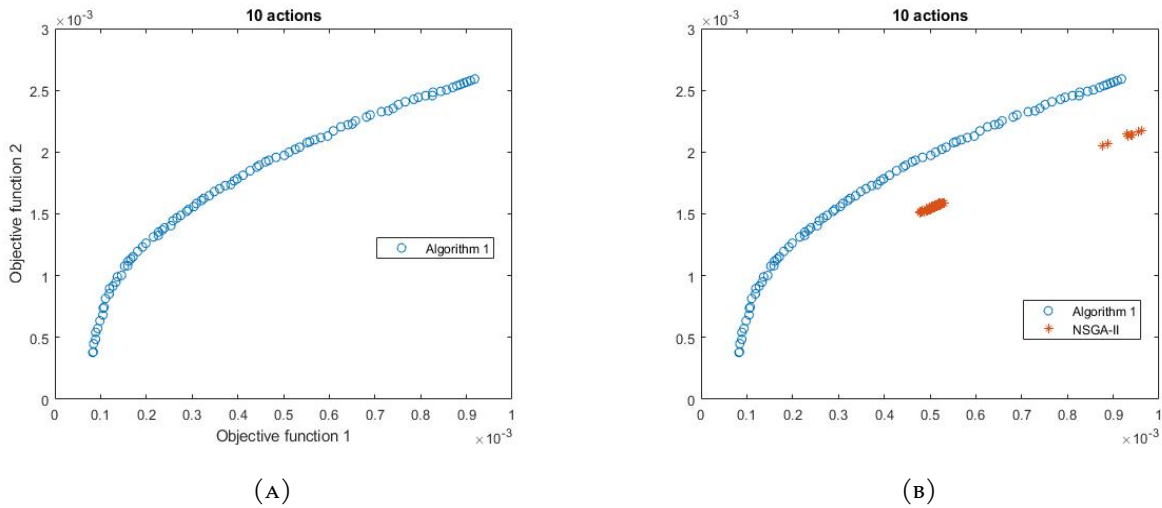


FIGURE 1. Pareto front on the 10 assets of the Algorithm 1 and the NSGA-II method

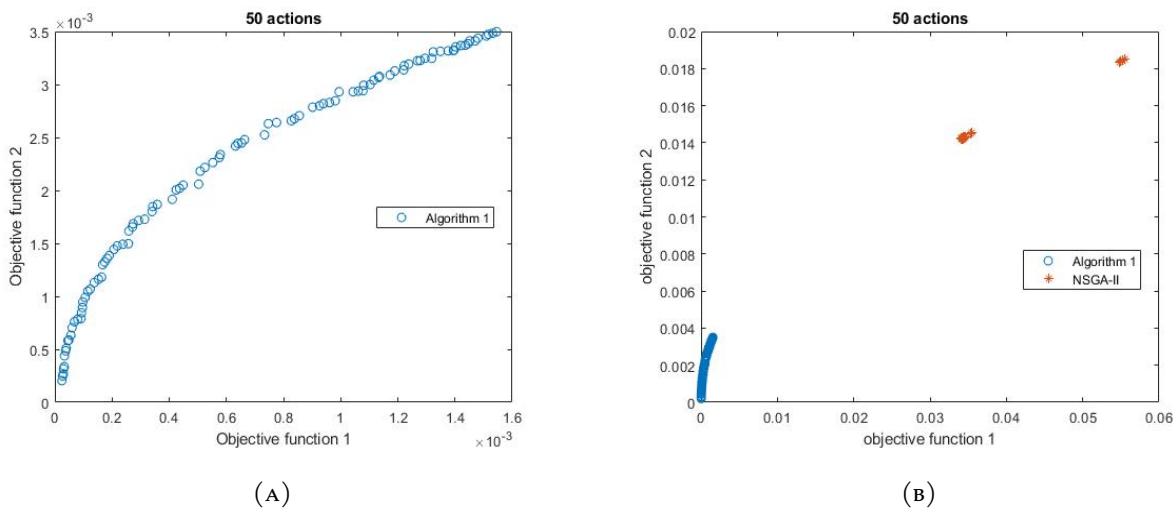


FIGURE 2. Pareto front on the 50 assets of the Algorithm 1 and the NSGA-II method

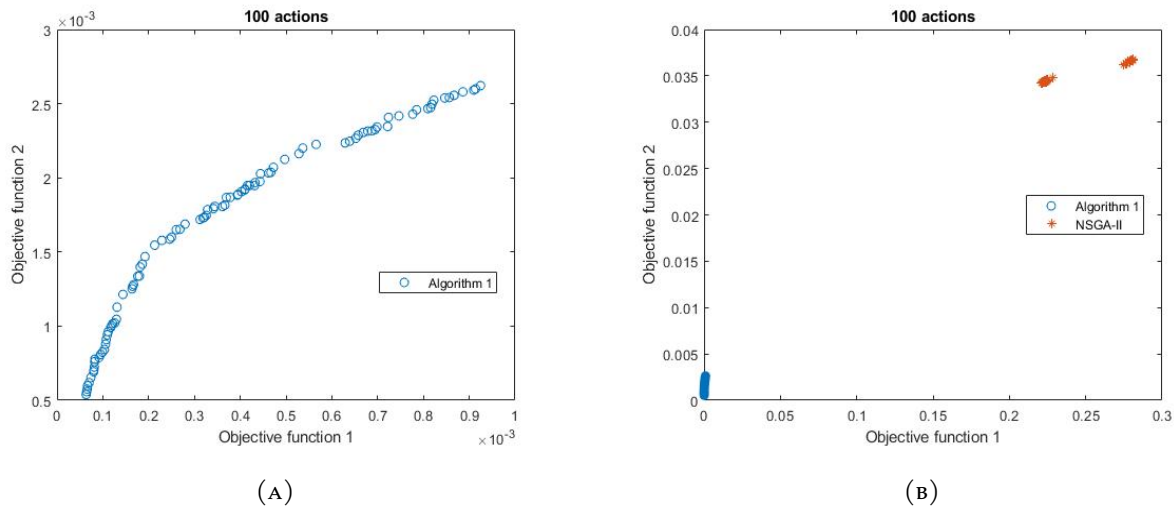


FIGURE 3. Pareto front on the 100 assets of the Algorithm 1 and the NSGA-II method

5. CONCLUSION

In this work, we proposed a multi-objective optimization model for portfolio selection, as well as a solution method based on the augmented Lagrangian. The model takes into account the decision-maker's preferences through inequality and equality constraints, which restrict the choice of certain assets in the portfolio. Unlike classical approaches, our method does not transform the multi-objective problem into a parametric single-objective problem.

We also developed an algorithm to solve portfolio optimization problems, including linear and nonlinear constraints. A numerical simulation was then conducted, showing that our algorithm produces more effective results than those obtained by the NSGA-II algorithm, as evidenced by the comparison of results in Figures 1, 2, and 3.

For future research, we plan to address portfolio problems with cardinality constraints, which introduce discontinuities, and to further improve our solution method to meet the requirements of this type of constraint.

Authors' Contributions. All authors have read and approved the final version of the manuscript. The authors contributed equally to this work.

Conflicts of Interest. The authors declare that there are no conflicts of interest regarding the publication of this paper.

REFERENCES

- [1] E. Aguilera, J. de Jong, F. Phillipson, S. Taamallah, M. Vos, Multi-objective Portfolio Optimization Using a Quantum Annealer, *Mathematics* 12 (2024), 1291. <https://doi.org/10.3390/math12091291>.

- [2] M. Bezoui, Contribution de la Programmation Multiobjectif dans l'Optimisation des Portefeuilles, Ph.D. Dissertation, Université des Sciences et de la Technologie Houari Boumediène, 2019.
- [3] E.G. Birgin, L.F. Bueno, N. Krejić, J.M. Martínez, Low Order-Value Approach for Solving Var-Constrained Optimization Problems, *J. Glob. Optim.* 51 (2011), 715–742. <https://doi.org/10.1007/s10898-011-9656-7>.
- [4] E.G. Birgin, J.M. Martínez, L.F. Prudente, Augmented Lagrangians with Possible Infeasibility and Finite Termination for Global Nonlinear Programming, *J. Glob. Optim.* 58 (2013), 207–242. <https://doi.org/10.1007/s10898-013-0039-0>.
- [5] E. Birgin, J. Martínez, On the Application of an Augmented Lagrangian Algorithm to Some Portfolio Problems, *EURO J. Comput. Optim.* 4 (2016), 79–92. <https://doi.org/10.1007/s13675-015-0052-9>.
- [6] S. Bullah, T.L. van Zyl, A Learnheuristic Approach to a Constrained Multi-Objective Portfolio Optimisation Problem, in: *Proceedings of the 2023 7th International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence*, ACM, New York, 2023, pp. 58–65. <https://doi.org/10.1145/3596947.3596965>.
- [7] X. Chen, L. Guo, Z. Lu, J.J. Ye, An Augmented Lagrangian Method for Non-Lipschitz Nonconvex Programming, *SIAM J. Numer. Anal.* 55 (2017), 168–193. <https://doi.org/10.1137/15m1052834>.
- [8] Y. Chen, A. Zhou, Multiobjective Portfolio Optimization via Pareto Front Evolution, *Complex Intell. Syst.* 8 (2022), 4301–4317. <https://doi.org/10.1007/s40747-022-00715-8>.
- [9] G. Cocchi, M. Lapucci, An Augmented Lagrangian Algorithm for Multi-Objective Optimization, *Comput. Optim. Appl.* 77 (2020), 29–56. <https://doi.org/10.1007/s10589-020-00204-z>.
- [10] G. Cocchi, M. Lapucci, P. Mansueto, Pareto Front Approximation Through a Multi-Objective Augmented Lagrangian Method, *EURO J. Comput. Optim.* 9 (2021), 100008. <https://doi.org/10.1016/j.ejco.2021.100008>.
- [11] A. Conn, G. Gould, P. Toint, *Lancelot: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*, Springer, 2013.
- [12] M.F.P. Costa, A.M.A. Rocha, E.M. Fernandes, An Artificial Fish Swarm Algorithm Based Hyperbolic Augmented Lagrangian Method, *J. Comput. Appl. Math.* 259 (2014), 868–876. <https://doi.org/10.1016/j.cam.2013.08.017>.
- [13] K. Deb, Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction, in: *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*, Springer, 2011: pp. 3–34. https://doi.org/10.1007/978-0-85729-652-8_1.
- [14] D. Dentcheva, G. Martinez, E. Wolfhagen, Augmented Lagrangian Methods for Solving Optimization Problems with Stochastic-Order Constraints, *Oper. Res.* 64 (2016), 1451–1465. <https://doi.org/10.1287/opre.2016.1521>.
- [15] A. Estrada-Padilla, D. Lopez-Garcia, C. Gómez-Santillán, H.J. Fraire-Huacuja, L. Cruz-Reyes, et al., Modeling and Optimizing the Multi-Objective Portfolio Optimization Problem with Trapezoidal Fuzzy Parameters, *Math. Comput. Appl.* 26 (2021), 36. <https://doi.org/10.3390/mca26020036>.
- [16] N. Fazzio, *Teoría y Métodos Para Problemas de Optimización Multiobjetivo*, Ph.D. Dissertation, Universidad Nacional de La Plata, 2018.
- [17] F.V. Jezeie, S.J. Sadjadi, A. Makui, Constrained Portfolio Optimization with Discrete Variables: An Algorithmic Method Based on Dynamic Programming, *PLOS ONE* 17 (2022), e0271811. <https://doi.org/10.1371/journal.pone.0271811>.
- [18] C. Kanzow, A.B. Raharja, A. Schwartz, An Augmented Lagrangian Method for Cardinality-Constrained Optimization Problems, *J. Optim. Theory Appl.* 189 (2021), 793–813. <https://doi.org/10.1007/s10957-021-01854-7>.
- [19] T. Krink, S. Paterlini, Multiobjective Optimization Using Differential Evolution for Real-World Portfolio Optimization, *Comput. Manag. Sci.* 8 (2009), 157–179. <https://doi.org/10.1007/s10287-009-0107-6>.
- [20] H.M. Markowitz, Foundations of Portfolio Theory, *J. Financ.* 46 (1991), 469–477. <https://doi.org/10.2307/2328831>.

- [21] M.K. de Melo, R.T.N. Cardoso, T.A. Jesus, Multiobjective Model Predictive Control for Portfolio Optimization with Cardinality Constraint, *Expert Syst. Appl.* 205 (2022), 117639. <https://doi.org/10.1016/j.eswa.2022.117639>.
- [22] M. Xue, L. Pang, S. Li, N. Xu, Convergence of the Augmented Lagrangian Algorithm with Lower-level Constraints for Cardinality-constrained Optimization Problems, Preprint, 2023. <https://www.researchgate.net/publication/375895224>.
- [23] S. Mishra, G. Panda, S. Meher, R. Majhi, M. Singh, Portfolio Management Assessment by Four Multiobjective Optimization Algorithm, in: 2011 IEEE Recent Advances in Intelligent Computational Systems, IEEE, 2011, pp. 326-331. <https://doi.org/https://doi.org/10.1109/RAICS.2011.6069328>.
- [24] V. Mohagheghi, S. Mousavi, B. Vahdani, A New Multi-objective Optimization Approach for Sustainable Project Portfolio Selection: a Real-world Application Under Interval-valued Fuzzy Environment, *Iran. J. Fuzzy Syst.* 13 (2016), 41–68. <https://doi.org/10.22111/ijfs.2016.2821>.
- [25] R.T. Rockafellar, Augmented Lagrange Multiplier Functions and Duality in Nonconvex Programming, *SIAM J. Control.* 12 (1974), 268–285. <https://doi.org/10.1137/0312021>.
- [26] H. Salah, Gestion des Actifs Financiers: de l'Approche Classique à la Modélisation Non Paramétrique en Estimation du DownSide Risk pour la Constitution d'un Portefeuille Efficient, Ph.D. Dissertation, Université Claude Bernard-Lyon I, 2015.
- [27] A. Tougma, A. Kaboré, K. Somé, Hyperbolic Augmented Lagrangian Algorithm for Multiobjective Optimization Problems, *Gulf J. Math.* 16 (2024), 151–170. <https://doi.org/10.56947/gjom.v16i2.1876>.
- [28] T. Yousefi, O. Aktas, Portfolio Optimization with Multi-Objective Optimization Algorithms, *Int. J. Adv. Nat. Sci. Eng. Res.* 7 (2024), 382–389.
- [29] C. Zhong, G. Li, Z. Meng, Beluga Whale Optimization: A Novel Nature-Inspired Metaheuristic Algorithm, *Knowledge-Based Syst.* 251 (2022), 109215. <https://doi.org/10.1016/j.knosys.2022.109215>.