

Research Article

Inexact Exponential Penalty Function with the Augmented Lagrangian for Multiobjective Optimization Algorithms

Appolinaire Tougma 🕞 and Kounhinir Some 🕞

Department of Mathematics, Norbert ZONGO University, RN14, Koudougou 376, Burkina Faso

Correspondence should be addressed to Kounhinir Some; sokous11@gmail.com

Received 10 May 2023; Revised 13 December 2023; Accepted 28 December 2023; Published 10 January 2024

Academic Editor: Keshlan S. Govinder

Copyright © 2024 Appolinaire Tougma and Kounhinir Some. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper uses an augmented Lagrangian method based on an inexact exponential penalty function to solve constrained multiobjective optimization problems. Two algorithms have been proposed in this study. The first algorithm uses a projected gradient, while the second uses the steepest descent method. By these algorithms, we have been able to generate a set of nondominated points that approximate the Pareto optimal solutions of the initial problem. Some proofs of theoretical convergence are also proposed for two different criteria for the set of generated stationary Pareto points. In addition, we compared our method with the NSGA-II and augmented the Lagrangian cone method on some test problems from the literature. A numerical analysis of the obtained solutions indicates that our method is competitive with regard to the test problems used for the comparison.

1. Introduction

In the multiobjective optimization area, the goal is to minimize or maximize or both at the same time one or several objective functions. However, in most cases, there is no single point that optimizes all objective functions at the same time. Therefore, many concepts have been developed, including the Pareto optimal conditions, in order to reach solutions in the multiple objectives case. Multiobjective optimization modeling has been used to solve many life-related problems, such as physical, economic, and transport [1–5].

Many methods have been proposed for solving optimization problems. These methods can be classified into two groups: exact methods and metaheuristic methods. In general, exact methods allow obtaining of the exact Pareto optimal solutions of a given problem, but they are unsuitable for solving problems with numerous variables and/or a large number of objective or constraint functions. Metaheuristic methods try to give good approximations of the true Pareto optimal solutions. Those methods are subjected to performance tests to evaluate some characteristics, such as computational time, convergence of algorithms to Pareto optimal solutions, and distribution of provided solutions on the Pareto front [6-10]. A lot of the methods in this family of metaheuristic methods are inspired by natural phenomena, and there are not any theorems or propositions that show how their algorithms are optimal or how they are convergent.

The majority of these two classes of methods are iterative, meaning that the research of the solution begins at an initial point. At each iteration, the current solution is improved in order to achieve the optimal solution. Sometimes, it is hard to figure out how many iterations are needed. However, there is a mathematical foundation and convergence properties based on theorems and propositions. Among the most used and well-known methods, we have the steepest descent, projected gradient, and Newton methods. Here are the recent works on these topics: the multicriteria optimization by steepest methods was proposed by Fliege and Svaiter [11]; the projected gradient methods have been used in wide work [6, 8, 9, 12, 13]; some of these variants are the works of Gonçalves and Prudente [14] on the extension of the Hager-Zhang conjugated gradient method and nonlinear conjugate gradient method proposed by Lucambio Pérez and Prudente [15]; those of the Newton method were proposed in [16, 17]; some variants like that quasi-Newton methods was presented in [18–20].

In practice, for solving multiobjective optimization problems with constraints, penalty functions are utilized to transform the initial problem into an unconstrained problem before commencing the optimization process. As a result, in recent years, several approaches using penalty functions have been proposed to address optimization problems subjected to inequality constraints, such as the augmented Lagrange function described in the following work [21-33]. This approach has been extensively employed for solving single-objective optimization problems. More recently, Cocchi et al. [34, 35] have developed an extension of this approach for the multiobjective case. Additionally, Upadhayay et al. [36] have proposed a method based on the cone method, which involves transforming the initial multiobjective problem into a single-objective problem and subsequently applying the augmented Lagrangian method.

In this paper, we extend the augmented Lagrange method to solve multiobjective optimization problems using an inexact exponential penalty function. The last version is Echebest et al.'s [7] work on the augmented Lagrange, using the penalty exponential function to solve a single-objective optimization problem. We proposed two algorithms with the same characteristics as the metaheuristics. They are stochastic, produce a population of solutions at a run, and provide a good approximation of the Pareto optimal solutions. One approach focuses on the steepest descent, and the other focuses on the projected gradient. Additionally, the theoretical convergence study has been done for the two algorithms. Our results have been compared to those of NSGA-II method on some test problems from the literature. On the test problems we have dealt with, our algorithms are best compared to NSGA-II. Furthermore, we also conducted a comparison with the augmented Lagrange cone method, as it also utilizes the augmented Lagrangian.

The following is the structure of the remaining paper. In Section 2, we have presented some preliminary concepts for multiobjective optimization. In Section 3, we have detailed our proposed method throughout the algorithms and theoretical and numerical performance studies. In Section 4, we give our conclusions and perspectives on this work.

2. Preliminaries

In the rest of our work, we will use the following notations: R_{++} is the set of positive reals, R^n is the set of column vectors of dimension n, and the image space of a matrix $A \in R^{m \times n}$ will be denoted by Im(A). The unit vector of dimension q will be denoted e. For any vector $u = (u_1, u_2, \dots, u_n)^T$, $v = (v_1, v_2, \dots, v_n)^T$, we define the convention for the following equalities and inequalities:

(i) $u = v \Leftrightarrow u_i = v_i$ for all $i = 1, 2, \dots, n$

(ii)
$$u < v \Leftrightarrow u_i < v_i$$
 for all $i = 1, 2, \dots, n$

- (iii) $u \leq v \Leftrightarrow u_i \leq v_i$ for all $i = 1, 2, \dots, n$
- (iv) $u \le v \Leftrightarrow u \le v$ with $u \ne v$

(v)
$$u \leq v \Leftrightarrow u \leq v$$
 with $u \neq v$

Without loss of generality, we will consider the multiobjective programming problem defined as follows:

$$\begin{array}{l} \min \ F(x) = \left(f_1(x), f_2(x), \cdots, f_q(x)\right), \\ \text{s.t}: \begin{cases} g_i(x) \leq 0 \quad \forall i \in I = \{1, 2, 3, \cdots, m\}, \\ x \in \Omega, \end{cases}$$

where $F : \mathbb{R}^n \longrightarrow \mathbb{R}^q$; $g_i : \mathbb{R}^n \longrightarrow \mathbb{R}$ are continuous and differentiable functions. Ω is a nonempty convex subset of \mathbb{R}^n . Let us denote \mathcal{X} the admissible space of the problem (1) defined by $\mathcal{X} = \{x \in \Omega : g(x) \le 0\}$.

We can state the following classical definitions of optimality in the Pareto sense, since it is not certain to find a solution that simultaneously minimizes all objective functions.

Definition 1. A point $x^* \in \mathcal{X}$ is the Pareto optimal for problem (1) if there exists no other $x \in \mathcal{X}$ such that

$$F(x) \le F(x^*),$$

$$F(x) \ne F(x^*).$$
(2)

The set of the Pareto optimality is thus given by $P_s = \{x \in \mathcal{X} : \nexists x \in \mathcal{X} : F(x) \le F(x^*) \text{ and } F(x) = F(x^*)\}$. Definition 1 gives an important property of the Pareto optimality, so we present the following definition, which proposes simpler conditions to obtain in the application.

Definition 2. A point $x^* \in \mathcal{X}$ is the weakly Pareto optimal for problem (1) if there exists no other $x \in \mathcal{X}$ such that

$$F(x) < F(x^*). \tag{3}$$

The set of Pareto optimality is thus given by $P = \{x^* \in \mathcal{X} : \exists x \in \mathcal{X} : F(x) < F(x^*)\}$. An existing relation between P and P_s is that P is large and contains P_s , i.e., $P_s \subseteq P$. We say that $x^* \in \mathbb{R}^n$ is a local Pareto optimal (resp., local weak Pareto optimal) if there exists a neighborhood $V(x^*) \in \mathbb{R}^n$ such that x^* is the Pareto optimal (resp., weak Pareto optimal) for F restricted to $V(x^*)$. Here, we are using a partial order induced by \mathbb{R}^n_{++} , $F(x) < F(x^*) \Leftrightarrow F(x) - F(x^*) \in -(\mathbb{R}^q_{++})$; this implies that a necessary but generally not sufficient condition for weak Pareto optimality is given by the following relation:

$$-(R^q_{++}) \cap Im\left(J_{F(x^*)}\right) = \emptyset, \tag{4}$$

where $J_{F(.)}$ denotes the Jacobian matrix of *F*. A point $x^* \in \mathcal{X}$ is stationary for *F* if it satisfies the relation (4). Now, a necessary condition for stationary optimality is given by the following definition.

Definition 3. A point $x^* \in \mathcal{X}$ is said to be the Paretostationary for problem (1) if, for all $d \in \mathbb{R}^n$,

$$\max_{j=1;q} \nabla f_j(x^*)^T d \ge 0.$$
(5)

Note also that if x^* is not the Pareto stationary, there exists a feasible direction d such that $\max_{j=1;q} \nabla f_j(x^*)^T d < 0$. By posing $\phi_x(d) = \max_{j=1;q} \nabla f_j(x^*)^T d$, we can see that $\phi_x(.)$ is continuous but does not admit a unique solution. Thus, we can define, as in [11], a problem which is well defined, i.e., it has a unique solution given by the following relation:

$$\min_{d \in \mathbb{R}^n} \max_{j=1;q} \nabla f_j(x^*)^T d + \frac{1}{2} \|d\|^2.$$
(6)

Now, by positing $\theta : \mathbb{R}^n \longrightarrow \mathbb{R}$, the function indicating the optimal value of problem (6), and *d*, the one indicating the optimal solution of problem (6), we have $\theta(x) \le 0$ for all $x \in \mathbb{R}^n$, and a point x^* is said to be the Pareto stationary if and only if $\theta(x^*) = 0$.

Now, we can give the following lemma, which proposes a well-known equivalent characterization of a Paretostationary point from the point of view of the projection.

Lemma 4. A point $x^* \in \mathcal{X}$ is said to be Pareto-stationary for problem (1) if, for all $d \in \mathbb{R}^n$,

$$x^* = \Pi_{\mathcal{X}} \left[x^* - \arg \min_{d} \max_{j=I;q} \nabla f_j (x^*)^T d \right], \qquad (7)$$

where $\Pi_{\Omega}[x]$ is the projection operator of the point x in the convex set \mathcal{X} .

Now, taking into account Definition 3 and Lemma 4, which characterize a Pareto-stationary point, we can define two equivalent definitions which characterize an ε -Pareto-stationary point.

Definition 5. ε -Approximate-Pareto-stationary-point APSP1. Let $\varepsilon > 0$. We say that $x \in \mathcal{X}$ is an ε -Pareto-stationary point for the problem (1) if

$$\min_{d \in \mathbb{R}^n} \max_{j=1;q} \nabla f_j(x^*)^T d \ge -\varepsilon.$$
(8)

Definition 6. ε -Approximate-Pareto-stationary-point APSP2. Let $\varepsilon > 0$. We say that $x \in \mathcal{X}$ is an ε -Pareto-stationary point for problem (1) if

$$\left\| x^* - \Pi_{\mathcal{X}} \left[x^* - \arg \min_{d} \max_{j=1;q} \nabla f_j(x^*)^T d \right] \right\| \le \varepsilon, \quad (9)$$

The methods we propose were established from the strategies of MOPG and MOSD methods. We have presented them, respectively, through Algorithm 1 and Algorithm 2. In these two algorithms, Armijo's rule was used to find the step of descent. Its algorithm is given by Algorithm 3, and we must note that the principle of Armijo's rule is to determine a real α_k , $\forall k \in N$ such as the values of the objective function always decrease in partial order by component in a finite number of iterations.

Now, we can present some main results that prove that the MOPG and MOSD algorithms produce solutions in a finite number of iterations. We will start by presenting the following lemma [11], which shows that Algorithm 3 is well-defined.

Lemma 7. Let $\sigma \in (0; 1)$, $x \in \mathcal{X}$, and $d \in \mathcal{X} - x$ such as $J_{F(x)}^T$. d < 0. Then, there exists $\overline{\gamma} < 1$ such as $F(x + \gamma d) < F(x) + \sigma \gamma$ $J_{F(x)}^T d$ for all $\gamma \in (0, \overline{\gamma}]$.

The following lemma [11, 34] shows that the MOPG and MOSD algorithms are well-defined, i.e., that these algorithms stop in a finite number of iterations.

Lemma 8. Let x^k be the sequence generated by Algorithm 1 and Algorithm 2. If F has bounded level sets in the sense that $\{x \in \mathbb{R}^n : F(x) \leq F(x^0)\}$ is compact, then each limit point of x^k is a Pareto-stationary point.

3. Main Results

3.1. Algorithms. In this section, we present the augmented Lagrangian function method with an inexact exponential penalty function to transform the constrained multiobjective programming problem into an unconstrained problem. The augmented Lagrangian function established from an inexact exponential penalty function is given by the following formula:

$$\mathscr{L}_{\rho}(x,\mu) = F(x) + \left(\sum_{i=1}^{m} \frac{\mu_i}{2\rho} \left[e^{\max\{0,\rho g_i(x)\}} - 1 \right]^2 \right) e, \quad (10)$$

for all $x \in \Omega$, $\mu \in R_+^m$ the Lagrange multiplier, $\rho \in R_{++}$ a penalty parameter and *e* a unit vector of R^q . It is important to note that the augmented Lagrangian function established from an exponential penalty function is differentiable. The technique is the same as the one used for quadratic cases, but we adapt it to exponential cases.

The gradient of the jth component of the augmented Lagrangian based on an inexact exponential penalty is given by

$$\nabla f_j(x) + \sum_{i=1}^m \mu_i \Big(e^{\max\{0; \rho_k g_i(x)\}} - 1 \Big) \nabla g_i(x).$$
(11)

Definition 5 and Definition 6 of the augmented Lagrangian subproblem established from an inexact penalty function are written as follows:

```
Data: F : \mathbb{R}^{n} \longrightarrow \mathbb{R}^{q}; x^{1} \in \mathbb{R}^{n}.

1 k=1

2 while x^{k} is not Pareto-stationary do

3 Compute

d^{k} \longleftarrow \arg \min_{d \in \mathscr{X} - x^{k} j = 1; q} \max \nabla f_{j}(x^{k})^{T} d + 1/2 ||d||^{2};

\alpha_{k} = \operatorname{Armijo} - \operatorname{Rule} - \operatorname{Line} - \operatorname{Search}(\mathbb{F}(.), x^{k}, d^{k});

4 x^{k+1} = x^{k} + \alpha_{k} d^{k}

5 k = k + 1;

6 end
```

Algorithm	1:	Multiobj	ective	projected	gradient	(MOPG)
-----------	----	----------	--------	-----------	----------	--------

Data: $F : \mathbb{R}^n \longrightarrow \mathbb{R}^q$; $x^1 \in \mathbb{R}^n$. 1 k=12 while x^k is not Pareto-stationary do 3 Compute $d^k \longleftarrow \arg \min_{d \in \mathbb{R}^n} \max_{j=1} \nabla f_j(x^k)^T d + 1/2 ||d||^2$; $\alpha_k = \operatorname{Armijo} - \operatorname{Rule} - \operatorname{Line} - \operatorname{Search}(\mathbb{F}(.), x^k, d^k)$; 4 $x^{k+1} = x^k + \alpha_k d^k$ 5 k = k + 1; 6 end

ALGORITHM 2: Multiobjective steepest descent (MOSD).

Data: $F : \mathbb{R}^n \longrightarrow \mathbb{R}^q$; $x \in \mathbb{R}^n$; $d \in \mathbb{R}^n$ $\alpha_0 > 0$; $\delta \in (0, 1)$; $\beta \in (0, 1)$. 1 $\alpha = \alpha_0$;; 2 while $F(x + \alpha d) \ge F(x) + \beta \alpha J_{F(x)} d$ do 3 $\alpha = \delta \alpha$; 4 end

ALGORITHM 3: Armijo-rule-line-search.

$$\min_{d \in \mathbb{R}^{n}} \max_{j=1; q} \left(\nabla f_{j}(x^{*}) + \sum_{i=1}^{m} \mu_{i} \left(e^{\max\left\{0; \rho_{k}g_{i}(x^{*})\right\}} - 1 \right) \nabla g_{i}(x^{*}) \right)^{T} d \ge 0,$$

$$x^{*} = \Pi_{\Omega} \left[x^{*} - \arg\min_{d} \max_{j=1; q} \left(\nabla f_{j}(x^{*}) + \sum_{i=1}^{m} \mu_{i} \left(e^{\max\left\{0; \rho_{k}g_{i}(x^{*})\right\}} - 1 \right) \nabla g_{i}(x^{*}) \right)^{T} d \right].$$
(12)

We can now define two different ways of defining ε -Paretostationary, expressed by the following two definitions.

Definition 9. Let $\varepsilon > 0$. A point $x \in \Omega$ is an ε -approximate Pareto stationary point APSP1 for the Lagrange based on an inexact exponential penalty if for all feasible direction $d \in \mathbb{R}^n$, it holds

$$\min_{d\in\mathbb{R}^n}\max_{j=1;q}\left(\nabla f_j(x^*) + \sum_{i=1}^m \mu_i \left(e^{\max\left\{0;\rho_k g_i(x^*)\right\}} - 1\right) \nabla g_i(x^*)\right)^T d \ge -\varepsilon.$$
(13)

Definition 10. Let $\varepsilon > 0$. A point $x \in \Omega$ is an ε -approximate Pareto stationary point APSP2 for the Lagrange based on an inexact exponential penalty if for all feasible direction $d \in \mathbb{R}^n$, it holds

$$\left\| x^* - \Pi_{\Omega} \left[x^* - \arg \min_{d} \max_{j=1,q} \left(\nabla f_j(x^*) + \sum_{i=1}^m \mu_i \left(e^{\max\left\{ 0: \rho_k g_i(x^*) \right\}} - 1 \right) \nabla g_i(x^*) \right)^T d \right] \right\| \le \varepsilon.$$
(14)

Thus, based on the ideas of [7, 34], we propose an augmented Lagrangian algorithm based on an inexact exponential penalization for solving multiobjective optimization programs.

A detailed description of the two algorithms is as follows. As an input, we define a set of nondominated points of the initial problem under bound constraint (without the other constraints). This is the set that will be considered as the set of reference points to find the Pareto optimal solutions. At each iteration, the Lagrange function established from an exponential penalty function is used with a penalty parameter ρ_k and Lagrange multipliers μ^k . The multiplier estimate for each point $x_t \in \mathcal{X}^{k+1}$ is given by the relation $\mu_i^{k+1} = \overline{\mu}_i^k e^{\rho_k g_i(x_t)}$ which is a multiplicative form, unlike the quadratic penalty form, where the dual update is additive. In the equation $\sigma_i^{k+1} = \max_{x \in \mathcal{X}^{k+1}} (\mu_i^{k+1} - \overline{\mu}_i^k / \rho_k)$ in line 17 of both algorithms, we have the parameter that measures progress in terms of infeasibility and complementarity. In line 4 of Algorithm 4 and Algorithm 5, each $x_t \in \mathcal{X}^k$ is used for exploration. If for Algorithm 5, $\theta_k(x) < 0$, or for Algorithm 4,

$$\left\|x^* - \Pi_{\Omega}\left[x^* - \arg\min_{d} \max_{j=1;q} \left(\nabla f_j(x^*) + \sum_{i=1}^m \mu_i \left(e^{\max\left\{0; \varphi_k g_i(x^*)\right\}} - 1\right) \nabla g_i(x^*)\right)^T d\right]\right\| > 0,$$
(15)

where $\theta_k(.)$ is the optimal value of the $d_k = \arg \min_{d \in \mathbb{R}^n} \max_{j=1,q} \nabla \mathscr{D}_{\rho}^j(x_t, \mu^k)^T d + (1/2) ||d||$ problem; the point x_t is used to generate a new descent direction d and a descent step α , which is obtained by $\alpha = \operatorname{Armijo} - \operatorname{Rule} - \operatorname{Line} - \operatorname{Search}(d, x_t, \mathscr{D}_{\rho_k}(.,\mu^k))$. Then, a new point z is determined by the MOPG or MOSD algorithm which is an ε_k -Paretostationary point where ε_k varies and converges to zero at each iteration. This new point is used to filter the points in X_{tmp} . If z dominates points in the set X_{tmp} , then we delete these points and add z to the set. For updating the Lagrange multipliers, note that for a $\mu_i^k > 0$, such that $\rho_k \longrightarrow 0$, the penalty term $\mu_i^k / \rho_k(e^{\max \{0, \rho_k g_i(x^k)\}} - 1) \longrightarrow \infty$ for nonfeasible points (i.e., $g_i(x^k) > 0$) and tends to 0 for feasible points (i.e. $g_i(x^k) \le 0$) [10].

We will now proceed to an analysis of the convergence of the two algorithms taking into account assumptions such as the convexity of the objective functions of the constraints and also the admissible space is nonempty.

3.2. Convergence Analysis. In this section, we present convergence results for Algorithm 4 and Algorithm 5. As usual in the scalar case, we also assume that the objective functions are convex, as indicated by the assumptions below.

Assumption 11. The set $\Omega \subseteq \mathbb{R}^n$ is closed and convex. The set $\mathcal{X} = \{x \in \mathbb{R}^n : x \in \Omega : g_i(x) \le 0, \forall i, i = 1, \overline{m}\}$ is not empty.

Assumption 12. The objective function F has bounded level sets in the multiobjective sense, i.e., the set $\{x \in \mathbb{R}^n, F(x) \leq F(x_0)\}$ is compact.

Assumption 13. The sequence $\{\varepsilon_k\}_{k \in \mathbb{N}}$ is such that $\lim_{k \to \infty} \varepsilon_k \to 0$.

Note that under Assumption 11, Assumption 12, and Proposition 14 from the work of Cocchi and Lapucci [34], we can deduce that the MOPG and MOSD algorithms of Algorithms 4 and 5, respectively, are well-defined, i.e., they stop in a finite number of iterations. For the step size, the good definition of Algorithm 3 results from Lemma 7 which is a main result proving that the step size is determined in a finite number of iterations.

The following proposition characterizes the solutions of the solution set generated by Algorithms 4 and 5.

Proposition 14. Let $\{X^{k+1}\}$ be the sequence of set generated by Algorithm 4 and Algorithm 5. Then, for each k and for each $x^{k+1} \in X^{k+1}$, x^{k+1} is an ε_k -Pareto-stationary point and is not dominated by any other point in X^{k+1} with respect to equation (10).

Now, for the convergence results for each point $x^k \in X^k$, we start by giving the following technical result.

Lemma 15. Let $\{X^k\}$ be a sequence of set generated by Algorithm 4 or Algorithm 5. Let $\{x^k\} \in \{X^k\}$ be a sequence for all k such that

$$\lim_{k \to \infty} x^k = x^*, \tag{16}$$

Assume that x^* is admissible. Then, for all $i = 1, 2, \dots, m$ such that $g_i(x^*) < 0$, we have $\overline{\mu}_i^k e^{\rho_k g_i(x^{k+1})} = 0$ for all k sufficiently large.

Proof. Let $g_i(x^*) < 0$, for all $i = 1, 2, \dots, m$. From the instructions of the algorithms, we consider the following two cases:

(a) $\rho_k \longrightarrow \infty$

The sequence $\{\bar{\mu}_i^k\}$ is bounded by definition, since $g_i(x^*)$ < 0; for $k \in K$ sufficiently large, we get $e^{\rho_k g_i(x^{k+1})} = 0$.

(b) ρ_k bounded

According to line (25) of the instructions of Algorithm 4 or Algorithm 5, there exists a k_0 such that for all $k > k_0$, $\|\sigma^{k+1}\| \le \tau \|\sigma^k\|$. We obtain for k sufficiently large $\lim_{k \in K} \sigma^{k+1} = 0$. Thus, $(e^{\rho_k g_i(x^{k+1})} - 1/\tau_k) \ne 0$ which implies that $\bar{\mu}_i^k = 0$. By the definition of μ_i^{k+1} , we get the result.

Then, we will prove that the points generated by the ALEPMO1 and ALEPMO2 algorithms are feasible based on the following propositions.

Proposition 16. Feasibility for ϵ -approximate ALEPMO1. Let $\{X^k\}$ be the sequence of set generated by Algorithm 4 with APSP1. Let $\{x^k\}$ be any sequence of points such that $x^k \in X^k$ for all k. Then, each cluster point x^* of $\{x^k\}$ is a feasible point of problem (1), i.e., $g_i(x^*) \leq 0$.

Proof. The proof of this proposition can be found in the very nature of the definition of Algorithm 4 and proposition 5 in the work of Drummond and Iusem in [12]. \Box

Proposition 17. Feasibility for ϵ -approximate ALEPMO2). Let $\{X^k\}$ be the sequence of set generated by Algorithm 5 with

Data: $\bar{\mu}_{\max} \in R^p_{++} - \{\infty\}; \tau \in [0, 1[; \gamma > 0; \rho_0 \in R_{++}; \bar{\mu}_0 \in [0, \bar{\mu}_{\max}];$ $\sigma_0 > 0$; $\{\varepsilon_k\} \in R_+$ such that $\lim_{k \to \infty} \varepsilon_k = 0$; X^0 a list of feasible nondominated points for the original problem. 1 for $k = 1, 2, \dots$ do Let $\mathscr{L}_{\rho_{\iota}}$ the current Augmented Lagrangian using the Exponential Penalty 2 Function defined as: $\mathscr{L}_{\rho}(x,\mu) = F(x) + (\sum_{i=1}^{m} (\mu_i/2\rho) [e^{\max\{0,\rho g_i(x)\}} - 1]^2)e;$ Set $X_{tmp} = X^k$; for $x_t \in X^k$ do 3 $\mathbf{if} \| x^* - \Pi_{\Omega}[x^* - \arg \min_d \max_{j=1} \overline{;} q(\nabla f_j(x^*) + \sum_{i=1}^m \mu_i (e^{\max\{0; \rho_k g_i(x^*)\}} - 1) \nabla g_i(x^*))^T d] \| > 0$ 4 then Set $d_k = \arg \min_{d \in \{X-x_k\}} \max_{i=1,q} \nabla \mathscr{L}_{\rho}^j(x_t, \mu^k)^T d + 1/2 ||d||;$ 5 Set α = Armijo – Rule – Line – Search $(d_k, x_t, \mathscr{L}_{\rho_k}(., \mu^k))$; 6 Set $z = MOPG(x_t + \alpha d_k, \mathscr{L}_{\rho_k(.,\mu^k)}, \varepsilon_k);$ 7 if $\exists y \in X_{tmp} : \mathscr{L}_{\rho_k(y,\mu^k)} \leqq \mathscr{L}_{\rho_k(z,\mu^k)}$ then 8 Set $X_{tmp} = X_{tmp} \setminus \{x \in X_{tmp} : \mathscr{L}_{\rho_k(z,\mu^k)} \notin \mathscr{L}_{\rho_k(x,\mu^k)}\} \cup \{z\};$ 9 10 end 11 end 12 end Set $X^{k+1} = X_{tmp}$; 13 for $i = 1, 2, \dots, m$ do 14 Set $\mu_i^{k+1} = \overline{\mu}_i^k e^{\rho_k g_i(x)}$; 15 Set $\tilde{\mu}_i^{k+1} = \max(0, \min(\mu_i^{k+1}, \bar{\mu}_{\max}));$ 16 Set $\sigma_i^{k+1} = \max_{x \in X^{k+1}} (\mu_i^{k+1} - \bar{\mu}_i^k / \rho_k);$ Set $\bar{\mu}_i^{k+1} = \max_{x \in X^{k+1}} \tilde{\mu}_i^{k+1};$ 17 18 19 end if $\|\sigma^{k+1}\| \leq \tau \|\sigma^k\|$ then 20 21 Set $\rho_{k+1} = \rho_k$; 22 else 23 Set $\rho_{k+1} = \gamma \rho_k$; 24 end 25 end

ALGORITHM 4: Augmented Lagrangian based on an inexact exponential penalty for multiobjective optimization (ALEXPMO1).

APSP2. Let $\{x^k\}$ be any sequence of points such that $x^k \in X^k$ for all k, with $\varepsilon_k \longrightarrow 0$. Let $\lim_{k \longrightarrow \infty} x^k = x^*$, $\mu_i^k > 0$ for all k and for all $i \in \{1, \dots, m\}$. Then, x^* is a feasible point of problem (1), i.e., $g_i(x^*) \le 0$.

Proof. Let $K \subseteq \{0, 1, \dots\}$ be an infinite subset such that

$$\lim_{k \to \infty} x^k = x^*. \tag{17}$$

Consider the two cases:

- (i) the sequence $\{\rho_k\}$ is bounded
- (ii) the sequence $\{\rho_k\}$ is unbounded, i.e., $\lim_{k \to \infty} \rho_k = \infty$

Case 1. Since $\{\rho_k\}$ is bounded, from the instruction of the algorithm, there must exist \bar{k} such that, for all $k > \bar{k}$, we have $\|\sigma^{k+1}\| \le \tau \|\sigma^k\|$. This means that

$$\lim_{k \to \infty} \left\| \sigma^{k+1} \right\| = 0, \tag{18}$$

i.e.,

$$\lim_{k \to \infty} \sigma_i^{k+1} = \lim_{k \to \infty} \max_{x \in X^{k+1}} \left\{ \frac{\mu_i^{k+1} - \mu_i^k}{\rho_k} \right\} = 0.$$
(19)

Since by assumption $\mu^k > 0$ for all *i* and *k*, it has to be

$$\lim_{k \to \infty} \max_{x \in X^{k+1}} \left\{ \frac{e^{\rho_k \mathcal{G}_i(x)} - 1}{\rho_k} \right\} = 0.$$
 (20)

Thus, we obtain $\lim_{k \to \infty} \max_{x \in X^{k+1}} \{e^{\rho_k g_i(x)}\} \le 1$ which implies that $\lim_{k \to \infty} \max_{x \in X^{k+1}} \{g_i(x)\} \le 0$. But $\max_{x \in X^{k+1}} \{g_i(x)\} \ge g_i(x^{k+1})$; hence,

$$g_i(x^*) = \lim_{k \to \infty} g_i(x^{k+1}) \le \lim_{k \to \infty} \max_{x \in X^{k+1}} \{g_i(x)\} \le 0.$$
(21)

Data: $\bar{\mu}_{\max} \in \mathbb{R}^p_{++} - \{\infty\}; \tau \in [0, 1[; \gamma > 0; \rho_0 \in \mathbb{R}_{++}; \bar{\mu}_0 \in [0, \bar{\mu}_{\max}]; \sigma_0 > 0; \{\varepsilon_k\} \in \mathbb{R}_+ \text{ such that } \lim_{k \to \infty} \varepsilon_k = 0; X^0 \text{ a list of feasible}$ nondominated points for the original problem. for $k = 1, 2, \dots$ do 1 Let $\mathscr{L}_{\rho_{\iota}}$ the current Augmented Lagrangian using the Exponential Penalty 2 Function defined as: $\mathcal{L}_{\rho}(x,\mu) = F(x) + \left(\sum_{i=1}^{m} (\mu_i/2\rho) \left[e^{\max\left\{0,\rho g_i(x)\right\}} - 1\right]^2\right)e, x \in \mathcal{X};$ Set $X_{tmp} = X^k$; for $x_t \in X^k$ do 3 if $\theta_k(x_t) < 0$ 4 then Set $d_k = \arg \min_{d \in \mathbb{R}^n} \max_{i=1,q} \nabla \mathscr{L}_{\rho}^j(x_t, \mu^k)^T d + 1/2 ||d||;$ 5 Set α = Armijo – Rule – Line – Search $(d_k, x_t, \mathscr{L}_{\rho_k}(., \mu^k));$ 6 Set $z = MOSD(x_t + \alpha d_k, \mathscr{L}_{\rho_k(.,\mu^k)}, \varepsilon_k);$ 7
$$\begin{split} \text{if } \exists y \in X_{tmp} &: \mathscr{L}_{\rho_k(y,\mu^k)} \leqq \mathscr{L}_{\rho_k(z,\mu^k)} \text{ then} \\ &\text{Set } X_{tmp} = X_{tmp} \setminus \big\{ x \in X_{tmp} : \mathscr{L}_{\rho_k(z,\mu^k)} \gneqq \mathscr{L}_{\rho_k(x,\mu^k)} \big\} \cup \{z\}; \end{split}$$
8 9 10 end 11 end 12 end Set $X^{k+1} = X_{tmp}$; 13 14for *i* = 1, 2, …, *m* do Set $\mu_i^{k+1} = \overline{\mu}_i^k e^{\rho_k g_i(x)}$; 15 Set $\tilde{\mu}_i^{k+1} = \max(0, \min(\mu_i^{k+1}, \bar{\mu}_{\max}));$ 16 Set $\sigma_i^{k+1} = \max_{x \in X^{k+1}} (\mu_i^{k+1} - \bar{\mu}_i^k / \rho_k);$ Set $\bar{\mu}_i^{k+1} = \max_{x \in X^{k+1}} \tilde{\mu}_i^{k+1};$ 17 18 19 end if $\|\sigma^{k+1}\| \leq \tau \|\sigma^k\|$ then 20 Set $\rho_{k+1} = \rho_k$; 21 22 else 23 Set $\rho_{k+1} = \gamma \rho_k$; 24 end 25 end

ALGORITHM 5: Augmented Lagrangian based on exponential penalty for multiobjective optimization (ALEXPMO2).

Case 2. From the instruction of Algorithm 5, we have at each iteration

$$\max_{j=1,q} \left\{ \left(\nabla f_j(x^{k+1}) + \sum_{i=1}^m \mu_i^k \left(e^{\max\left\{ 0; \rho_k g_i(x^{k+1}) \right\}} - 1 \right) \nabla g_i(x^{k+1}) \right)^T d \right\}$$

$$\geq -\varepsilon_k \quad \forall d \in \mathbb{R}^n.$$
(22)

Letting $d^{k+1} = x - x^{k+1}$ and $\bar{g}_i = e^{\max\{0, \rho_k g_i(x^{k+1})\}} - 1$, we obtain for all k

$$\max_{j=\bar{1},\bar{q}} \nabla f_j \left(x^{k+1} \right)^T \left(x - x^{k+1} \right) \ge -\varepsilon_k - \left(\sum_{i=1}^m \mu_i^k \bar{g}_i \left(x^{k+1} \right) \nabla g_i \left(x^{k+1} \right) \right)^T \cdot \left(x - x^{k+1} \right) \quad \forall x \in \Omega.$$
(23)

Since \mathcal{X} is nonempty, we can choose $x \in \{x \in \mathcal{X} : g(x) \le 0\}$. Using the convexity of the g_i , we can bound the last term as follows:

$$-\left(\sum_{i=1}^{m} \mu_{i}^{k} \bar{g}_{i}\left(x^{k+1}\right) \nabla g_{i}\left(x^{k+1}\right)\right)^{T}\left(x-x^{k+1}\right)$$

$$\geq \sum_{i=1}^{m} \mu_{i}^{k} \bar{g}_{i}\left(x^{k+1}\right) \left(g_{i}\left(x^{k+1}-g_{i}(x)\right)\right)$$

$$\geq \sum_{i=1}^{m} \mu_{i}^{k} \bar{g}_{i}\left(x^{k+1}\right) \left(g_{i}\left(x^{k+1}\right)\right).$$
(24)

This inequality is satisfied if the g_i s are convex (i.e., $g_i(x) \ge g_i(x^{k+1}) + \nabla g_i(x^{k+1})^T(x - x^{k+1}))$, and -g(x) is non-negative since $x \in \mathcal{X}$.

Using equation (23) and by dividing by e^{ρ_k} , we obtain

$$\max_{j=\bar{1},\bar{q}} \frac{\nabla f_{j}(x^{k+1})^{T}(x-x^{k+1})}{e^{\rho_{k}}} \ge -\frac{\varepsilon_{k}}{e^{\rho_{k}}} + \sum_{i=1}^{m} \mu_{i}^{k} e^{-\rho_{k}} \bar{g}_{i}(x^{k+1}) g(x^{k+1}).$$
(25)

Now, suppose by contradiction that $\{i \in \{1, \dots, q\}: g_i(x^*) > 0\} \neq \emptyset$.

Given that, $\rho_k \longrightarrow \infty$, $\{\mu^k\}$ is bounded, and the g_i are continuous; for sufficiently large k, we have

$$e^{-\rho_k}\bar{g}_i(x^{k+1}) = e^{\max\left\{-\rho_k,\rho_k\left(g_i(x^{k+1})-1\right)\right\}} - e^{-\rho_k} = 0, \forall i: g_i(x^*) < 0.$$
(26)

In addition, we have

$$\lim_{k \to \infty} \sum_{i:g_i(x^*)=0} \left(e^{\max\left\{ -\rho_k, \rho_k\left(g_i(x^{k+1})-1\right)\right\}} - e^{-\rho_k} \right) g_i(x^{k+1}) = 0.$$
(27)

Let us set $A = \lim_{k \to \infty} \max_{j=1,q} (\nabla f_j(x^{k+1})^T (x - x^{k+1}) / e^{\rho_k})$. Thus, using the equation (25), we have

$$A \geq \lim_{k \to \infty} -\frac{\varepsilon_{k}}{e^{\rho_{k}}} + \lim_{k \to \infty} \sum_{i:g_{i}(x^{*})>0} \mu_{i}^{k} \left(e^{\max\left\{-\rho_{k}\cdot\rho_{k}\left(g_{i}(x^{k+1})-1\right)\right\}} - e^{-\rho_{k}}\right) g\left(x^{k+1}\right)$$
$$\geq \lim_{k \to \infty} -\frac{\varepsilon_{k}}{e^{\rho_{k}}} + \lim_{k \to \infty} \sum_{i:g_{i}(x^{*})>0} \mu_{i}^{k} \left(e^{\rho_{k}\left(g_{i}(x^{k+1})-1\right)} - e^{-\rho_{k}}\right) g\left(x^{k+1}\right).$$

$$(28)$$

For *k* sufficiently large $k \in K$, we obtain

$$0 \ge \lim_{k \longrightarrow \infty} \sum_{i:g_i(x^*) > 0} \mu_i^k \left(e^{\rho_k \left(g_i(x^{k+1}) - 1 \right)} - e^{-\rho_k} \right) g\left(x^{k+1} \right), \quad (29)$$

which is absurd. Thus, the set $\{i \in \{1, \dots, q\}: g_i(x^*) > 0\} = \emptyset$, i.e., x^* is feasible.

Finally, we prove that a limit point of the sequence generated by the algorithm is a Pareto-optimal point.

Proposition 18. Optimality for ϵ -approximate ALEPMO1. Let $\{X^k\}$ be the sequence of set generated by Algorithm 4 with APSP1. Let $\{x^k\}$ be any sequence of points such that $x^k \in X^k$ for all k. Suppose that the sequence $\{\epsilon^k \rho_k\}$ is bounded. Then, every cluster point x^* of $\{x^k\}$ is a Pareto-stationary point of problem (1).

Proof. Let $K \in \{0, 1, \dots\}$ be an infinite subset such as

$$\lim_{k \to \infty} x^{k+1} = x^*.$$
(30)

According to the Proposition 16, we have $x^* \in \Omega$. Let us suppose by contradiction that x^* is not the Pareto-stationary for problem (1), since by definition, Ω is convex; there exists $z \in \Omega$ such that $g(z) \leq 0$ and

TABLE 1: List of multiobjective optimization test problems.

Problems	п	9	Parameter bounds	Source
BNH1	2	2	$[0, 5]^2$	[42]
SCH	1	2	[-4, 4]	[5]
M-LAP1	2	2	$[-7, 1]^2$	[34]
M-LAP2	а	2	$[0, 1]^n$	[34]
JOS1	b	2	$[0, 1]^n$	[42]
ZLT1	С	2	$[-1000, 1000]^n$	[42]
MLF1	1	2	[0, 20]	[42]
M-MLF2	1	2	[0, 20]	[42]
LRS1	2	2	[-50, 50]	[42]
MHHM1	1	3	[0, 1]	[42]
MHHM2	2	3	[0, 1]	[42]
IKK1	1	3	[-50, 50]	[42]
VFM1	2	3	[-2, 2]	[42]
DGO1	1	2	[-10, 13]	[42]
SSFYY1	1	2	[-100, 100]	[42]
VU1	2	2	[-3, 3]	[42]
BK1	2	2	$[-5, 10]^2$	[42]
SSFYY1	1	2	[-100, 100]	[42]
VU2	2	2	$[-3, 3]^2$	[42]
COSH	1	2	[0, 5]	[43]
KD1	2	2	$[0.1,1]\times[0,1]$	[44]
SRN	2	2	$[-20, 20]^2$	[42]
SSFYY2	2	2	$[-100, 100]^2$	[42]

$$\max_{j=1;q} \nabla f_j(x^*)^T (z - x^*) < 0.$$
(31)

Instructions of Algorithm 1, by posing

$$\bar{g}_{i}^{k} = e^{\max\left\{0,\rho_{k}g_{i}(x^{k+1})\right\}} - 1,$$

$$\hat{x}^{k+1} = \Pi_{\Omega} \left[x^{k+1} - \max_{j=1;q} \nabla f_{j}(x^{k+1}) - \sum_{i=1}^{m} \mu_{i}\bar{g}_{i}^{k} \nabla g_{i}(x^{k+1}) \right],$$
(32)

we get at each iteration that

$$\left\| x^{k+1} - \widehat{x}^{k+1} \right\| \le \varepsilon_k. \tag{33}$$

Using the properties of the projection, we have for all $x \in \Omega$ that

$$\left(x^{k+1} - \max_{j=1;q} \nabla f_j\left(x^{k+1}\right) - \sum_{i=1}^m \mu_i^k \overline{g}_i^k \nabla g_i\left(x^{k+1}\right) - \widehat{x}^{k+1}\right)^T \left(z - \widehat{x}^{k+1}\right) \le 0.$$
(34)



FIGURE 1: Performance profiles of CPU time of the ALEXPMO1, ALEXPMO2, and NSGA-II.

By adding and subtracting x^{k+1} and rearranging, we get

Let us set

$$\max_{j=1;q} \nabla f_{j} \left(x^{k+1} \right)^{T} \left(z - x^{k+1} \right) \geq -\max_{j=1;q} \nabla f_{j} \left(x^{k+1} \right)^{T} \left(x^{k+1} - \hat{x}^{k+1} \right) \\ + \left\| x^{k+1} - \hat{x}^{k+1} \right\|^{2} + \left(x^{k+1} - \hat{x}^{k+1} \right)^{T} \left(z - x^{k+1} \right) \\ - \left(\sum_{i=1}^{m} \mu_{i}^{k} \bar{g}_{i}^{k} \nabla g_{i} \left(x^{k+1} \right) \right)^{T} \left(z - x^{k+1} \right) \\ - \left(\sum_{i=1}^{m} \mu_{i}^{k} \bar{g}_{i}^{k} \nabla g_{i} \left(x^{k+1} \right) \right)^{T} \left(x^{k+1} - \hat{x}^{k+1} \right).$$

$$(35)$$

$$C = \left(\sum_{i=1}^{m} \mu_i^k \bar{g}_i^k \nabla g_i \left(x^{k+1}\right)\right)^T \left(z - x^{k+1}\right) - \left(\sum_{i=1}^{m} \mu_i^k \bar{g}_i^k \nabla g_i \left(x^{k+1}\right)\right)^T \left(x^{k+1} - \hat{x}^{k+1}\right).$$
(36)

And using the convexity of g the last two terms can be bounded as follows:



FIGURE 2: Performance profiles of the purity metric of the ALEXPMO1, ALEXPMO2, and NSGA-II.

$$C \ge \sum_{i=1}^{m} \mu_{i}^{k} \bar{g}_{i}^{k} \left(g_{i} \left(x^{k+1} \right) - g_{i}(z) \right) - \left(\sum_{i=1}^{m} \mu_{i}^{k} \bar{g}_{i}^{k} \nabla g_{i} \left(x^{k+1} \right) \right)^{T} \\ \cdot \left(x^{k+1} - \bar{x}^{k+1} \right) \ge \sum_{i=1}^{m} \mu_{i}^{k} \bar{g}_{i}^{k} g_{i} \left(x^{k+1} \right) \\ - \sum_{i=1}^{m} \mu_{i}^{k} \bar{g}_{i}^{k} \left\| \nabla g_{i} \left(x^{k+1} \right) \right\| \left\| x^{k+1} - \bar{x}^{k+1} \right\|.$$
(37)

Now, considering the term $\sum_{i=1}^m \mu_i^k \bar{g}_i^k g_i(x^{k+1}),$ we have

$$\sum_{i=1}^{m} \mu_{i}^{k} \bar{g}_{i}^{k} g_{i}\left(x^{k+1}\right) = \sum_{i:g_{i}(x^{*})=0} \mu_{i}^{k} g_{i}\left(x^{k+1}\right) \left(e^{\max\left\{0,\rho_{k} g_{i}\left(x^{k+1}\right)\right\}} - 1\right) \\ + \sum_{i:g_{i}(x^{*})<0} \mu_{i}^{k} g_{i}\left(x^{k+1}\right) \left(e^{\max\left\{0,\rho_{k} g_{i}\left(x^{k+1}\right)\right\}} - 1\right)$$
(38)
= 0.

Recalling that $||x^{k+1} - \hat{x}^{k+1}|| \le \varepsilon_k$, we have

$$-\sum_{i=1}^{m} \mu_{i}^{k} \overline{g}_{i}^{k} \left\| \nabla g_{i} \left(x^{k+1} \right) \right\| \left\| x^{k+1} - \widehat{x}^{k+1} \right\| \ge -\varepsilon_{k} \sum_{i=1}^{m} \mu_{i}^{k} \overline{g}_{i}^{k} \left\| \nabla g_{i} \left(x^{k+1} \right) \right\|.$$

$$(39)$$

Let us set $B = \max_{j=1, q} \nabla f_j (x^{k+1})^T (z - x^{k+1})$. Now, replacing the different transformations in equation (35), we obtain

$$B \ge -\max_{j=1;q} \nabla f_{j} \left(x^{k+1} \right)^{T} \left(x^{k+1} - \hat{x}^{k+1} \right) + \left\| x^{k+1} - \hat{x}^{k+1} \right\|^{2} \\ + \left(x^{k+1} - \hat{x}^{k+1} \right)^{T} \left(z - x^{k+1} \right) - \varepsilon_{k} \sum_{i=1}^{m} \mu_{i}^{k} \bar{g}_{i}^{k} \left\| \nabla g_{i} \left(x^{k+1} \right) \right\|.$$

$$(40)$$



FIGURE 3: Performance profiles of the Γ -spread metric of the ALEXPMO1, ALEXPMO2, and NSGA-II.

Passing to the limit for $k \in K$, k sufficiently large, since ∇f_j $j = 1, \dots, q, \nabla g_i i = 1, \dots, m$ are continuous, $||x^{k+1} - \hat{x}^{k+1}|| \longrightarrow 0$, the sequence $\{\mu_i^k\}$ is bounded, and $e^{\max\{0, \rho_k g_i(x^{k+1})\}} - 1 \longrightarrow 0$, we obtain

$$\max_{j=1;q} \nabla f_j(x^*)^T (z - x^*) \ge 0, \tag{41}$$

which contradicts our initial hypothesis.

Proposition 19. Optimality for ϵ -approximate ALEPMO2. Let $\{X^k\}$ be the sequence of set generated by Algorithm 4 with APSP2. Let $\{x^k\}$ be any sequence of points such that $x^k \in X^k$ for all k. Then, every cluster point x^* of $\{x^k\}$ is a Paretostationary point of problem (1). *Proof.* Let $K \in \{0, 1, \dots\}$ be an infinite subset such as

$$\lim_{k \to \infty} x^{k+1} = x^*. \tag{42}$$

According to Proposition 17, we have $x^* \in \Omega$. Suppose by contradiction that x^* is not Pareto-stationary for problem (1), since by definition, Ω is convex, there exists $z \in \Omega$ such that $g(z) \le 0$ and

$$\max_{j=1;q} \nabla f_j(x^*)^T (z - x^*) < 0.$$
(43)

By posing

$$\bar{g}_{i}^{k} = e^{\max\left\{0, \rho_{k} g_{i}\left(x^{k+1}\right)\right\}} - 1, \tag{44}$$



FIGURE 4: Performance profiles of the Δ-spread metric of the ALEXPMO1, ALEXPMO2, and NSGA-II.

and instructions of Algorithm 5, we have at each iteration

$$\max_{j=1;q} \nabla f_j \left(x^{k+1} \right)^T \left(z - \widehat{x}^{k+1} \right) \ge -\varepsilon_k - \sum_{i=1}^m \mu_i^k \overline{g}_i^k \nabla g_i \left(x^{k+1} \right)^T \left(z - \widehat{x}^{k+1} \right).$$

$$\tag{45}$$

Now consider the term

$$-\sum_{i=1}^{m}\mu_{i}^{k}\overline{g}_{i}^{k}\nabla g_{i}\left(x^{k+1}\right)^{T}\left(z-\widehat{x}^{k+1}\right),$$
(46)

since by definition the constraints are convex, using the properties of convexity, we get

$$\bar{g}_i^k \nabla g_i \left(x^{k+1} \right)^T \left(z - \hat{x}^{k+1} \right) \le g_i(z) - g_i \left(x^{k+1} \right), \qquad (47)$$

which implies that

$$-\sum_{i=1}^{m} \mu_{i}^{k} \bar{g}_{i}^{k} \nabla g_{i} \left(x^{k+1} \right)^{T} \left(z - \hat{x}^{k+1} \right) \geq \sum_{i=1}^{m} \mu_{i}^{k} \bar{g}_{i}^{k} \left(g_{i} \left(x^{k+1} \right) - g_{i}(z) \right).$$
(48)

Using the fact that $g_i(z) \le 0$ by definition, we have

$$-\sum_{i=1}^{m}\mu_{i}^{k}\overline{g}_{i}^{k}\nabla g_{i}\left(x^{k+1}\right)^{T}\left(z-\widehat{x}^{k+1}\right) \geq \sum_{i=1}^{m}\mu_{i}^{k}\overline{g}_{i}^{k}g_{i}\left(x^{k+1}\right).$$
 (49)

Methods	Time	Purity	Δ -spread	Г-spread
ALEXPMO1	039.3581	1.0000	10.4368	0.0334
ALEXPMO2	039.3456	1.0000	10.4368	0.0334
NSGA-II	141.3812	0.0000	01.3681	1.0986
		TABLE 3: BNH1 problem.		
Mathada	Time	Durity	Asproad	<i>C</i> spread
Withillous		i unty		
ALEXPMO1	52.21	1.00	0.78	5.47
ALEXPMO2	54.13	1.00	0.77	6.70
NSGA-II	40.62	0.91	0.83	7.76

TABLE 2: MLF1 problem.

```
TABLE 4: M-LAP<sub>2</sub>-30.
```

Methods	Time	Purity	Δ -spread	Γ -spread
ALEXPMO1	213.68	0.9806	0.7937	33.5762
ALEXPMO2	206.40	0.9806	0.7938	32.6896
NSGA-II	072.07	0.0300	0.9198	0.047

TABLE 5: M-LAP₂-100 problem.

Methods	Time	Purity	Δ -spread	Γ -spread
ALEXPMO1	2068.5145	0.9720	0.7798	0.0187
ALEXPMO2	0791.4442	0.9900	0.7791	0.0183
NSGA-II	0071.8289	0.0000	0.8403	0.0931

Equation (25) becomes

$$\max_{j=1;q} \nabla f_j \left(x^{k+1} \right)^T \left(z - \widehat{x}^{k+1} \right) \ge -\varepsilon_k + \sum_{i=1}^m \mu_i^k \overline{g}_i^k g_i \left(x^{k+1} \right).$$
(50)

Thus, considering the term

$$\mu_{i}^{k} \bar{g}_{i}^{k} g_{i} \left(x^{k+1} \right) = \mu_{i}^{k} \left(e^{\max \left\{ 0, \rho_{k} g_{i} \left(x^{k+1} \right) \right\}} - 1 \right) g_{i} \left(x^{k+1} \right), \quad (51)$$

if $x^* \in \mathcal{X}$, for k sufficiently large, we have

$$\mu_i^k \left(e^{\max\left\{ 0, \rho_k g_i(x^{k+1}) \right\}} - 1 \right) g_i(x^{k+1}) = 0, \forall i : g_i(x) \le 0.$$
 (52)

Recalling that for $k \in K$, $k \longrightarrow \infty$, and $\varepsilon_k \longrightarrow 0$, we obtain

$$\max_{j=1;q} \nabla f_j(x^*)^T (z - x^*) \ge 0,$$
 (53)

which contradicts our initial assumption.

3.3. Numerical Experiences. In this section, we apply Algorithms 4 and 5 to problems with bound constraints $l \le x \le u$, linear and nonlinear constraints. We first compared the

two methods that we named, respectively, ALEXPMO1 and ALEXPMO2, and then, as our methods are nonscalar, we compared them with a well-known nonscalar method for solving multiobjective optimization programs, namely, the NSGA-II method. As a reminder, the NSGA-II method is a genetic algorithm based on a nondominant strategy. The code of the NSGA-II method is available at https://www .mathworks.com/matlabcentral/fileexchange/49806-matlabcode-for-constrained-nsga-ii-dr-s-baskar-s-tamilselvi-and-pr-varshini.

In order to compare the different methods, we use the performance profiles developed by Dolan and Moré [37] and later used in many works [34, 35, 38-41] regarding the purity metric and the spread metrics (Δ -spread and Γ -spread metrics). The purity metric measures the quality of the Pareto front generated by an algorithm. It gives the percentage of nondominated solutions generated by the method [41]. The purity metric is given by the following formula:

$$Purity(S) = \frac{|F_{p,s} \bigcap F_p|}{|F_{p,s}|},$$
(54)

with $F_{p,s}$, the solutions generated by a solver $s \in S$ for a problem $p \in P$, where S is the set of solvers and P is the set of test

Table	6:	M-JOS-100	prob	lem.
-------	----	-----------	------	------

Methods	Time	Purity	Δ -spread	Г-spread
ALEXPMO1	034.7689	1.0000	0.7479	0.0443
ALEXPMO2	045.9536	1.0000	1.0878	0.3084
NSGA-II	110.6087	0.8651	1.0000	0.1397



FIGURE 5: Pareto front of MLF1 and BNH1 problems.

problems. F_p represents the set of solutions generated by all solvers for the problem p ($F_p = \bigcup F_{p,s}$) without the dominated points.

The spread metrics used are Γ -Spread and Δ -Spread. The Γ -Spread metric measures the maximum spacing of solutions generated by a solver [41]. It is given by the following formula:

$$\Gamma - \text{Spread}(S) = \max_{j \in \{1; \dots; q\}} \left(\max_{i \in \{0; \dots; N\}} \left\{ \delta_{i,j} \right\} \right), \quad (55)$$

where *N* represents the number of solutions generated by a solver, *m* is the number of objective functions, and $\delta_{i,j} = f_j(x_{k+1}) - f_j(x_k)$ whose values of $f_j(x_k)$ are arranged in ascending order. The Δ -spread metric measures the distribution of solutions generated by a solver [41]. It is calculated by the following formula:

$$\Delta - \text{Spread}(S) = \max_{j \in \{1; \dots; q\}} \frac{\delta_{0,j} + \delta_{N,j} + \sum_{i=1}^{N-1} \left| \delta_{i,j} - \overline{\delta}_{i,j} \right|}{\delta_{0,j} + \delta_{N,j} + (N-1)\overline{\delta}_{i,j}}, \quad (56)$$

where $\delta_{i,j}$ is the average of the $\delta_{i,j}$ with $j = 1, \dots, N - 1$. $\delta_{0,j}$ and $\delta_{N,j}$ represent the extreme points indexed by 0 and N + 1. Thus, we used the technique proposed in [41] to com-

pute the extreme points for problems that do not have an analytic front. We first removed the dominant points from the meeting on all these fronts. Then, for each component of the objective function, we selected the pair corresponding to the highest distance in pairs measured using $f_i(.)$.

We then use the performance profiles proposed in [37, 40] for an appreciation of the performance of the four metrics presented above. We refer the reader to the articles cited above for more information on performance profiles. Recall that the performance profiles are presented by a diagram of a cumulative distribution function $\rho(\tau)$ which is defined as follows:

$$\rho_{s}(\tau) = \frac{1}{|P|} \left| \left\{ p \in P : r_{p,s} \le \tau \right\} \right|, \tag{57}$$

with $r_{p,s} = t_{p,s}/\min \{t_{p,\overline{s}} : \overline{s} \in S\}$. Since performance profiles are used for metrics whose lowest value indicates better performance and metric purity, we will pose $t_{p,s} = 1/t_{p,s}$ as proposed in [40]. For more information on the metrics, we refer the reader to the references cited above.

We have implemented Algorithm 4 and Algorithm 5 in MATLAB. The search directions d^k and the optimal value θ_k are computed by solving the subproblem (6). Algorithm 4 and Algorithm 5 are implemented with the following



FIGURE 6: Pareto front of M-LAP2-30 and M-LAP2-100.

parameters: $\mu_0 = 1 \in \mathbb{R}^m$, $\rho_0 = 10^6$, $\tau_0 = 0.9$, $\gamma = 10$, $\sigma_0 = 10 \in \mathbb{R}^m$, and $\mu_{\text{max}} = 10^4$. Since we use Armijo's rule to determine the descent step, we use $\delta = 0.9$ and $\beta = 10^{-4}$. For the NSGA-II method, we used the default parameters except for the number of generations which was set to 20,000 since the number of generations and the number of executions for each problem allows it to reduce the sensitivity of its genetic operators.

As we presented previously, in the section devoted to the convergence analysis, Algorithms 4 and 5 converge for convex problems. Therefore, we have tested their performance with convex problems existing in the literature. Table 1 presents the set of test problems that we used, which are a total of 70. The first column gives the name of the problems, the second shows the number of variables, the third presents the number of objectives, the fourth is devoted to the bound constraints, and the last column is dedicated to highlight the sources of the problem. Since the algorithms run on problems that have bound constraints, we have defined search domains in the form $l \le x \le u$ for problems that do not have bound constraints. These problems have their names preceded by the letter M in Table 1 to indicate that they have been modified. The bound constraints are transformed to linear constraints in the following way: given that the domain is defined as $\{x \in \mathbb{R}^n : l \le x \le u\}$, we obtain $l-x \le 0$ and $x-u \le 0$ whose number is 2n with n as the number of variables. For Algorithm 4 and Algorithm 5, the set of initial nondominated points is determined in the Ω space and the projection in Algorithm 4 into the Ω domain.We used an HP EliteBook laptop equipped with an Intel Core i7-3687U processor with a base frequency range of 2.10 GHz to 2.60 GHz and 4 GB of RAM to test our algorithms.



FIGURE 7: Pareto front of M-JOS-100.

where

 $a \in \{2, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100\},\$ $b \in \{2, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100\},\$ $c \in \{2, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100\}.$ (58)

We begin our analysis by comparing the performance of



FIGURE 8: Pareto front of SCH, JOS₅₀, and JOS₁₀₀.

the algorithms ALEXPMO1, ALEXPMO2, and NSGA-II in terms of computation time, the purity metric, and the spread metrics.

Figure 1 shows that ALEXPMO1 and ALEXPMO2 are competitive, and that ALEXPMO2 has the highest probability of being the best method. Comparing with NSGA-II, we see that ALEXPMO2 is in the lead with an interest factor of 8 and a probability of 0.50. However, if we increase this interest factor to more than 8, NSGA-II becomes the best method with a probability of 0.90 for an interest factor of 10.

Figure 2 examines the performance of the algorithms in terms of purity. Figure 2(a) shows that there is no significant difference between ALEXPMO2 and ALEXPMO1. However, Figures 2(b) and 2(c) reveal that ALEXPMO2 and ALEXPMO1 surpass NSGA-II with a probability of about 0.90 of being the best methods.

Figure 3 focuses on the performance of the algorithms in terms of Γ -spread. According to Figure 3(a), ALEXPMO1 is better than ALEXPMO2 for an interest factor of less than 21. However, ALEXPMO2 is better than ALEXPMO1 for an

interest factor greater than 21. As for Figures 3(b) and 3(c), ALEXPMO2 is better than NSGA-II and ALEXPMO1 for an interest factor less than 51 with a probability of about 0.30 and remains competitive for an interest factor greater than 51.

Figure 4 gives the performance of the algorithms in terms of Δ -spread. Figures 4(a)–4(c) show that ALEXPMO1 and ALEXPMO2 are competitive compared to NSGA-II in the uniform distribution of solutions on the Pareto front.

We have especially added a comparative study of MLF1, BNH1, M-LAP2, and M-JOS problems. At first, we present the Pareto optimal front of these problems by using our two algorithms and also the NSGA-II algorithm.

Here, we give the values of the performance parameters that we present above. It is about the computational time, purity, and spread (Δ -spread and Γ -spread) of all three algorithms mentioned above presented in Tables 2–6.

We have examined the Pareto fronts for different reference problems, including one variable MLF1 (a multimodal problem), BNH1 (two variables constrained), M-LAP2 (30



FIGURE 9: Pareto front of JOS₅₀₀, DGO1, and MLF1.

	Purity		Δ -spre	ad	Γ-spread	
	ALEXPMO1	ALCM	ALEXPMO1	ALCM	ALEXPMO1	ALCM
SCH	1	1	0.8029	1.1277	0.0561	1.5791
JOS ₅₀	0.9967	0.8040	0.7499	0.9818	0.0285	0.6830
JOS ₁₀₀	1	0.8000	0.7496	0.9878	0.0283	0.6821
JOS ₅₀₀	1	0.5165	0.7496	1.2198	0.0288	0.3380
DGO1	1	1	1.0328	0.7398	0.0120	0.2524
MLF1	1	0	1.4367	2.9095	0.0334	0.6335

and 100 variables), and M-JOS1 (100 variables). Figures 5(a) and 5(b), 6(a) and 6(b), and 7 show the results obtained with 100 solutions. The results indicate that the ALEXPMO1 and ALEXPMO2 methods are superior to NSGA-II according to the purity, Δ -spread, and Γ -spread metrics. For the MLF1 problem, we find that the solutions proposed by

ALEXPMO1 and ALEXPMO2 are closer to the global front than those generated by NSGA-II.

3.4. Comparisons of ALEXPMO1 with another Lagrangian Method. In this subsection, we compare the ALEXPMO1 method with another well-established method called

"augmented Lagrangian cone method" (ALCM), developed by Upadhayay et al. [36]. The purpose of this comparison is to evaluate the relative performances of both approaches, namely, ALEXPMO1 and ALCM.

We selected six representative test problems that were successfully solved by ALCM to conduct this comparison. The parameters of our method, ALEXPMO1, were fixed as defined in the previous sections. For the ALCM method, we set the parameter m_1 to 30 for all problems, chose η_1 as a random number generated within the interval (0,1), set ε to a value of 10^{-4} , fixed α at 2.5, set E_{sup} to 10, and set the bounds of the Lagrange multipliers, λ_{min} and λ_{max} , to 0 and 1, respectively.

To solve the subproblem, we set the parameters ρ to 0.95, ξ to 0.80, initialized x_0 with random values within the interval (0, 1), and set t_0 to 0.

By conducting a comprehensive comparative analysis of the performances of both methods on the six test problems, including the SCH problem, JOS problems with 50, 100, and 500 variables, the DGO1 problem, and the MLF1 problem, we found that ALEXPMO1 outperformed ALCM in terms of solution distribution on the Pareto front. Although ALCM may be competitive for specific problems, it is important to highlight that, with the chosen parameters, ALEXPMO1 manages to identify solutions on the global Pareto front that are superior to those obtained by ALCM for the multimodal problem MLF1.

To support our results, we conducted a visual comparison of the Pareto fronts generated by both methods, illustrated in Figures 8(a)-8(c) and Figures 9(a)-9(c). These in-depth and objective analyses provide a better understanding of the advantages and limitations of each method, thus contributing to the advancement of research in the field of multiobjective optimization.

Table 7 presents the comparative study of ALEXPMO1 and ALCM, while Figures 8 and 9 illustrate the Pareto fronts generated by both methods for each problem. This detailed and objective analysis allows for a better understanding of the advantages and limitations of each method and contributes to the advancement of research in the field of multiobjective optimization.

4. Conclusion

In this study, we presented a new approach for solving multiobjective optimization problems, which combines an inexact exponential penalty function with the augmented Lagrangian technique. To solve the subproblem obtained by using our new approach, we used the steepest descent or projected gradient, which allowed us to produce, respectively, Algorithms 4 and 5 for general convex multiobjective optimization problems. The convergence properties of both methods have been examined using assumptions such as convexity and boundlessness. Our numerical experiments indicate that the two proposed algorithms are competitive compared to existing methods in the literature.

Data Availability

The data used to support the conclusion of the study are included in the paper.

Conflicts of Interest

The authors declare no competing interest.

Authors' Contributions

Appolinaire Tougma and Kounhinir Some contributed equally to this work.

Acknowledgments

The authors wish to thank the anonymous referees for their remarks that contributed to improve the presentation.

References

- A. Babaeinesami, P. Ghasemi, M. Abolghasemian, and A. P. Chobar, "A Stackelberg game for closed-loop supply chains under uncertainty with genetic algorithm and gray wolf optimization," *Supply Chain Analytics*, vol. 4, article 100040, 2023.
- [2] S. Lian, S. Meng, and Y. Wang, "An objective penalty functionbased method for inequality constrained minimization problem," *Mathematical Problems in Engineering*, vol. 2018, Article ID 7484256, 7 pages, 2018.
- [3] A. Rashidi Komijan, P. Ghasemi, K. Khalili-Damghani, and F. HashemiYazdi, "A new school bus routing problem considering gender separation, special students and mix loading: a genetic algorithm approach," *Journal of Optimization in Industrial Engineering.*, vol. 14, no. 2, pp. 23–39, 2021.
- [4] M. Reza Pourhassan, M. Khadem Roshandeh, P. Ghasemi, and M. Seyed Bathaee, "A multi echelon location-routing-inventory model for a supply chain network: NSGA II and multi-objective whale optimization algorithm," *International Journal of Supply* and Operations Management, vol. 11, pp. 23–31, 2023.
- [5] K. Somé, A. Compaoré, W. Sawadogo, B. Ulungu, and B. Somé, "Improving of numerical performance of the MOMA method," *International Journal of Mathematics and Computer Research*, vol. 5, no. 5, pp. 1817–1827, 2017.
- [6] J. Y. B. Cruz, L. R. L. Pérez, and J. G. Melo, "Convergence of the projected gradient method for quasiconvex multiobjective optimization," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 74, no. 16, pp. 5268–5273, 2011.
- [7] N. Echebest, M. D. Sánchez, and M. L. Schuverdt, "Convergence results of an augmented Lagrangian method using the exponential penalty function," *Journal of Optimization Theory and Applications*, vol. 168, no. 1, pp. 92–108, 2016.
- [8] N. S. Fazzio and M. L. Schuverdt, "Convergence analysis of a nonmonotone projected gradient method for multiobjective optimization problems," *Optimization Letters*, vol. 13, no. 6, pp. 1365–1379, 2019.
- [9] E. Fukuda and L. Drummond, "On the convergence of the projected gradient method for vector optimization," *Optimization*, vol. 60, no. 8-9, pp. 1009–1021, 2011.
- [10] P. Tseng and D. P. Bertsekas, "On the convergence of the exponential multiplier method for convex programming," *Mathematical Programming*, vol. 60, no. 1-3, pp. 1–19, 1993.
- [11] J. Fliege and B. F. Svaiter, "Steepest descent methods for multicriteria optimization," *Mathematical Methods of Operations Research (ZOR)*, vol. 51, no. 3, pp. 479–494, 2000.
- [12] L. Drummond and A. N. Iusem, "A projected gradient method for vector optimization problems," *Computational Optimization and Applications*, vol. 28, no. 1, pp. 5–29, 2004.

- [13] E. Fukuda and L. Graña Drummond, "Inexact projected gradient method for vector optimization," *Computational Optimization and Applications*, vol. 54, no. 3, pp. 473–493, 2013.
- [14] M. L. N. Gonçalves and L. F. Prudente, "On the extension of the Hager–Zhang conjugate gradient method for vector optimization," *Computational Optimization and Applications*, vol. 76, no. 3, pp. 889–916, 2020.
- [15] L. Lucambio Pérez and L. F. Prudente, "Nonlinear conjugate gradient methods for vector optimization," *SIAM Journal on Optimization*, vol. 28, no. 3, pp. 2690–2720, 2018.
- [16] T. D. Chuong, "Newton-like methods for efficient solutions in vector optimization," *Computational Optimization and Applications*, vol. 54, no. 3, pp. 495–516, 2013.
- [17] J. Fliege, L. M. G. Drummond, and B. F. Svaiter, "Newton's method for multiobjective optimization," *SIAM Journal on Optimization*, vol. 20, no. 2, pp. 602–626, 2009.
- [18] N. Mahdavi-Amiri and F. Salehi Sadaghiani, "A superlinearly convergent nonmonotone quasi-Newton method for unconstrained multiobjective optimization," *Optimization Methods* and Software, vol. 35, no. 6, pp. 1223–1247, 2020.
- [19] V. Morovati, H. Basirzadeh, and L. Pourkarimi, "Quasi-Newton methods for multiobjective optimization problems," 4OR, vol. 16, no. 3, pp. 261–294, 2018.
- [20] L. R. Prudente and D. R. Souza, "A quasi-Newton method with Wolfe line searches for multiobjective optimization," *Journal* of Optimization Theory and Applications, vol. 194, no. 3, pp. 1107–1140, 2022.
- [21] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt, "On augmented Lagrangian methods with general lower-level constraints," *SIAM Journal On Optimization*, vol. 18, no. 4, pp. 1286–1309, 2008.
- [22] E. G. Birgin, J. M. Martínez, and L. F. Prudente, "Optimality properties of an augmented Lagrangian method on infeasible problems," *Computational Optimization and Applications*, vol. 60, no. 3, pp. 609–631, 2015.
- [23] E. Birgin and J. Martínez, Practical Augmented Lagrangian Methods for Constrained Optimization, Society for Industrial and Applied Mathematics, 2014.
- [24] E. G. Birgin and J. M. Martínez, "On the application of an augmented Lagrangian algorithm to some portfolio problems," *EURO Journal On Computational Optimization*, vol. 4, no. 1, pp. 79–92, 2016.
- [25] E. G. Birgin, C. A. Floudas, and J. M. Martínez, "Global minimization using an augmented Lagrangian method with variable lower-level constraints," *Mathematical Programming*, vol. 125, no. 1, pp. 139–162, 2010.
- [26] Y. Cao, A. Seth, and C. D. Laird, "An augmented Lagrangian interior-point approach for large-scale NLP problems on graphics processing units," *Computers & Chemical Engineering*, vol. 85, pp. 76–83, 2016.
- [27] E. P. de Carvalho, A. dos Santos, and T. F. Ma, "Reduced gradient method combined with augmented Lagrangian and barrier for the optimal power flow problem," *Applied Mathematics and Computation*, vol. 200, no. 2, pp. 529–536, 2008.
- [28] N. Fazzio, Theory and Methods for Multiobjective Optimization Problems, Ph.D. Thesis, Universidad Nacional de La Plata, 2018.
- [29] S. He and S. Zhou, "A nonlinear augmented Lagrangian for constrained minimax problems," *Applied Mathematics and Computation*, vol. 218, no. 8, pp. 4567–4579, 2011.

- [30] R. Hug, E. Maitre, and N. Papadakis, "On the convergence of augmented Lagrangian method for optimal transport between nonnegative densities," *Journal of Mathematical Analysis and Applications*, vol. 485, no. 2, article 123811, 2020.
- [31] C. Kanzow and D. Steck, "An example comparing the standard and safeguarded augmented Lagrangian methods," *Operations Research Letters*, vol. 45, no. 6, pp. 598–603, 2017.
- [32] Y. Pan, "Lagrangian relaxation for the multiple constrained robust shortest path problem," *Mathematical Problems in Engineering.*, vol. 2019, article 3987278, 13 pages, 2019.
- [33] A. M. A. C. Rocha, T. F. M. C. Martins, and E. M. G. P. Fernandes, "An augmented Lagrangian fish swarm based method for global optimization," *Journal of Computational and Applied Mathematics*, vol. 235, no. 16, pp. 4611–4620, 2011.
- [34] G. Cocchi and M. Lapucci, "An augmented Lagrangian algorithm for multi-objective optimization," *Computational Optimization and Applications*, vol. 77, no. 1, pp. 29–56, 2020.
- [35] G. Cocchi, M. Lapucci, and P. Mansueto, "Pareto front approximation through a multi-objective augmented Lagrangian method," *EURO Journal on Computational Optimization*, vol. 9, article 100008, 2021.
- [36] A. Upadhayay, D. Ghosh, Q. H. Ansari, and Jauny, "Augmented Lagrangian cone method for multiobjective optimization problems with an application to an optimal control problem," *Optimization and Engineering*, vol. 24, no. 3, pp. 1633–1665, 2023.
- [37] E. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.
- [38] M. Ansary and G. Panda, "A sequential quadratically constrained quadratic programming technique for a multiobjective optimization problem," *Engineering Optimization*, vol. 51, no. 1, pp. 22–41, 2019.
- [39] M. Ansary and G. Panda, "A sequential quadratic programming method for constrained multi-objective optimization problems," *Journal of Applied Mathematics and Computing*, vol. 64, no. 1-2, pp. 379–397, 2020.
- [40] C. Brás and A. L. Custódio, "On the use of polynomial models in multiobjective directional direct search," *Computational Optimization and Applications*, vol. 77, no. 3, pp. 897–918, 2020.
- [41] A. Custódio, J. F. A. Madeira, A. I. F. Vaz, and L. N. Vicente, "Direct multisearch for multiobjective optimization," *SIAM Journal on Optimization*, vol. 21, no. 3, pp. 1109–1140, 2011.
- [42] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.
- [43] W. H. Zhang and T. Gao, "A min-max method with adaptive weightings for uniformly spaced Pareto optimum points," *Computers & Structures*, vol. 84, no. 28, pp. 1760–1769, 2006.
- [44] K. Deb, "Multi-objective genetic algorithms: problem difficulties and construction of test problems," *Evolutionary Computation*, vol. 7, no. 3, pp. 205–230, 1999.