Evolutionary Alternating Direction Method of Multipliers for Constrained Multi-Objective Optimization with Unknown Constraints *

Shuang Li[1], Ke Li[2], Wei Li[1] and Ming Yang[1]

[1]Control and Simulation Center, Harbin Institute of Technology, 150001, Harbin, China
[2]Department of Computer Science, University of Exeter, EX4 4QF, Exeter, UK
frank@hit.edu.cn

Abstract: Constrained multi-objective optimization problems (CMOPs) pervade real-world applications in science, engineering, and design. Constraint violation has been a building block in designing evolutionary multi-objective optimization algorithms for solving constrained multi-objective optimization problems. However, in certain scenarios, constraint functions might be unknown or inadequately defined, making constraint violation unattainable and potentially misleading for conventional constrained evolutionary multi-objective optimization algorithms. To address this issue, we present the first of its kind evolutionary optimization framework, inspired by the principles of the alternating direction method of multipliers that decouples objective and constraint functions. This framework tackles CMOPs with unknown constraints by reformulating the original problem into an additive form of two subproblems, each of which is allotted a dedicated evolutionary population. Notably, these two populations operate towards complementary evolutionary directions during their optimization processes. In order to minimize discrepancy, their evolutionary directions alternate, aiding the discovery of feasible solutions. Comparative experiments conducted against five state-of-the-art constrained evolutionary multi-objective optimization algorithms, on 120 benchmark test problem instances with varying properties, as well as two real-world engineering optimization problems, demonstrate the effectiveness and superiority of our proposed framework. Its salient features include faster convergence and enhanced resilience to various Pareto front shapes.

Keywords: Multi-objective optimization, constraint handling, alternating direction method of multipliers, evolutionary algorithm.

1 Introduction

Constrained multi-objective optimization problems (CMOPs) are ubiquitous in real-world applications. They are found in a broad spectrum of fields, including, but not limited to, tackling complex problems in computational science [1], streamlining design and control in engineering systems [2], and aiding in economic decision-making and strategy [3]. Despite their prevalence, CMOPs often present formidable challenges, primarily due to the interplay between unknown or poorly defined constraints and the requirement to balance multiple, potentially conflicting, objectives. For example, in applying reinforcement learning to autonomous driving, avoiding hazardous events like collisions is a critical constraint [4]. The safety-critical nature of such applications makes quantifying constraint violations impractical, often leading to binary outcomes of compliance or violation. Consequently, we introduce the concept of CMOPs with unknown constraints (CMOP/UC), a topic that, to the best of our knowledge, has not been adequately addressed in the evolutionary computation community.

Evolutionary algorithms (EAs), due to their inherent population-based nature, have seen widely accepted as an effective means for multi-objective optimization. Without loss of generality, three foundational algorithms—elitist non-dominated sorting genetic algorithms (NSGA-II) [5], multi-objective

*This manuscript is currently submitted for possible publication. Reviewers can feel free to use this in peer review.

evolutionary algorithms based on decomposition (MOEA/D) [6], and indicator-based EAs (IBEA) [7]—have significantly contributed to the body of literature in the evolutionary multi-objective optimization (EMO) domain. Building on these, substantial progress has been made in developing constraint handling techniques (CHTs) for CMOPs. Initial strategies, as seen in [8] and [9], predominantly favored feasible solutions, but this approach often loses its selection pressure when no feasible solution exist—a common occurrence in scenarios with a narrow feasible region. Subsequently, many studies have focused on using constraint violation (CV) information to guide evolutionary search, incorporating CV into the Pareto dominance relation [5, 10–14], or developing modified environmental selection mechanisms that balance convergence, feasibility—assessed by CV—while maintaining population diversity [15–22]. A separate line of research has looked into designing specialized repair operators that adjust infeasible solutions using CV values [23–27]. However, these techniques predominantly rely on CV information, which poses limitations for CMOP/UC scenarios. In these cases, CV information might be unavailable or poorly defined, as highlighted in our recent empirical study [28]. This limitation is particularly pronounced in highly constrained feasible regions, where prevalent CHTs struggle to generate feasible solutions.

In the field of Bayesian optimization (BO) [29], a well established method for black-box optimization, significant efforts have been made in addressing optimization problems with unknown constraints. A notable approach within this realm is the expected improvement (EI) with constraints, which views constraint satisfaction as a probabilistic classification challenge [30–32]. Beyond EI, constrained BO has seen the incorporation of various acquisition functions, such as integrated conditional EI [33], expected volume reduction [34], and predictive entropy search [35, 36]. These have shown promising results. Additionally, some algorithms employ the augmented Lagrangian function, reducing reliance on feasible solutions during the initial BO process. This approach allows the statistical model to integrate global information while the augmented Lagrangian provides localized control [37–39]. Nevertheless, given these methods were originally designed to address single-objective optimization problems, they are hardly directly applicable to deal with CMOP/UC.

Inspired by the principles of the alternating direction method of multipliers (ADMM) [40], which effectively decouples objective functions and constraints, we introduce a novel framework for tackling CMOP/UC. This framework, to our knowledge the first of its kind and hereafter referred to as evolutionary ADMM (`EADMM`), is outlined as follows.

- In `EADMM`, the CMOP/UC is reformulated into two subproblems in additive form, each managed by its own evolutionary population. These populations operate in complementary evolutionary directions, a strategy empirically shown to effectively navigate highly constrained infeasible regions.
- A key feature of `EADMM` is the strategy to minimize discrepancies between the co-evolving populations by alternating their evolutionary directions. This approach fosters the discovery of feasible solutions while minimizing objective functions.
- The `EADMM` framework is universally compatible, allowing integration of any existing EMO algorithm in a plug-and-play manner. For a proof-of-concept purpose, we will apply it within three iconic algorithms, including NSGA-II, IBEA, and MOEA/D.
- Our rigorous evaluation of `EADMM` included 120 benchmark test problems and two real-world engineering challenges. The results showcase `EADMM`'s effectiveness compared to five high-performance peer algorithms.

In the rest of this paper, we will provide some preliminary knowledge pertinent to this paper along with a pragmatic review of existing works on constrained multi-objective optimization in Section 2. In Section 3, we will delineate the algorithmic implementations of our proposed `EADMM` framework. The experimental settings are given in Section 4, and the empirical results are presented and discussed in Section 5. At the end, Section 6 concludes this paper and sheds some lights on future directions.

2 Preliminaries

This section starts with some basic definitions pertinent to this paper followed by a gentle tutorial of ADMM. At the end, we give a pragmatic overview of some selected up-to-date developments of EAs for CMOPs.

2.1 Basic Definitions

The CMOP/UC considered in this paper is defined as:

$$\begin{array}{ll} \underset{\mathbf{x}\in\Omega}{\text{minimize}} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))^\top \\ \text{subject to} & \mathbf{G}(\mathbf{x}) = (\underbrace{0, \ldots, 0}_{\ell})^\top \end{array}, \tag{1}$$

where $\Omega = [x_i^L, x_i^U]_{i=1}^n \subseteq \mathbb{R}^n$ is the search space, and $\mathbf{x} = (x_1, \ldots, x_n)^\top$ is a candidate solution. $\mathbf{F} : \Omega \to \mathbb{R}^m$ consists of m conflicting objective functions, and $\mathbb{R}^m$ is the objective space. $\mathbf{G}(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_\ell(\mathbf{x}))^\top : \Omega \to \mathbb{R}^\ell$ consists of $\ell \geq 1$ constraint functions. Note that this paper assumes $g_i(\mathbf{x})$ only returns a crispy response 0 when $\mathbf{x} \in \Omega$ is feasible, otherwise it returns 1 where $i \in \{1, \ldots, \ell\}$.

Definition 1. *Given two feasible solutions $\mathbf{x}^1$ and $\mathbf{x}^2$, $\mathbf{x}^1$ is said to Pareto dominate $\mathbf{x}^2$ (denoted as $\mathbf{x}^1 \preceq \mathbf{x}^2$) if and only if $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$, $\forall i \in \{1, \ldots, m\}$ and $\exists j \in \{1, \ldots, m\}$ such that $f_j(\mathbf{x}^1) < f_j(\mathbf{x}^2)$.*

Definition 2. *A solution $\mathbf{x}^* \in \Omega$ is Pareto-optimal with respect to (1) if $\nexists \mathbf{x} \in \Omega$ such that $\mathbf{x} \preceq \mathbf{x}^*$.*

Definition 3. *The set of all Pareto-optimal solutions is called the Pareto-optimal set (PS). Accordingly, $PF = \{\mathbf{F}(\mathbf{x}) | \mathbf{x} \in PS\}$ is called the Pareto-optimal front (PF).*

Definition 4. *The ideal objective vector $\mathbf{z}^* = (z_1^*, \ldots, z_m^*)^\top$ consists of the minimum of each objective function, i.e., $z_i^* = \min_{\mathbf{x}\in\Omega} f_i(\mathbf{x})$, $i \in \{1, \ldots, m\}$.*

Definition 5. *The nadir objective vector $\mathbf{z}^{\text{nad}} = (z_1^{\text{nad}}, \ldots, z_m^{\text{nad}})^\top$ consists of the worst objective function of the PF, i.e., $z_i^{\text{nad}} = \max_{\mathbf{x}\in PS} f_i(\mathbf{x})$, $i \in \{1, \ldots, m\}$.*

2.2 Working Mechanism of ADMM

This subsection starts with a gentle tutorial of the basic working mechanism of ADMM, followed by its extension for constrained optimization with unknown constraints. Given an equality-constrained convex optimization problem[1]:

$$\begin{array}{ll} \underset{\mathbf{x}\in\Omega}{\text{minimize}} & f(\mathbf{x}) \\ \text{subject to} & A\mathbf{x} = \mathbf{b} \end{array}, \tag{2}$$

where $A \in \mathbb{R}^{\ell\times n}$ and $\mathbf{b} \in \mathbb{R}^\ell$. ADMM considers splitting $f(\mathbf{x})$ into the following additive form:

$$\begin{array}{ll} \underset{\mathbf{x}\in\Omega,\ \mathbf{y}\in\Omega}{\text{minimize}} & p(\mathbf{x}) + q(\mathbf{y}) \\ \text{subject to} & A\mathbf{x} + B\mathbf{y} = \mathbf{c} \end{array}, \tag{3}$$

where $\mathbf{y} = (y_1, \ldots, y_n)^\top \in \Omega$ is an auxiliary variable, $B \in \mathbb{R}^{\ell\times n}$, and $\mathbf{c} \in \mathbb{R}^\ell$. The variable of (2) has been split into two parts, i.e., $\mathbf{x}$ and $\mathbf{y}$, with $f(\mathbf{x})$ being separable across this splitting. In particular, $p(\mathbf{x})$ and $q(\mathbf{x})$ are assumed to be convex. To handle the constraints in (3), ADMM constructs the augmented Lagrangian function (ALF) as an unconstrained surrogate of (3):

$$\begin{aligned} L_\rho(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}) \triangleq\ & p(\mathbf{x}) + q(\mathbf{y}) + \boldsymbol{\lambda}^\top (A\mathbf{x} + B\mathbf{y} - \mathbf{c}) \\ & + \frac{\rho}{2} \|A\mathbf{x} + B\mathbf{y} - \mathbf{c}\|_2^2 \end{aligned}, \tag{4}$$

[1] Note that ADMM is also applicable for non-convex optimization problems [40].

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_d)^\top \in \mathbb{R}^d$ is the Lagrange multiplier vector with regard to the constraints of equation (3). $\|A\mathbf{x} + B\mathbf{y} - \mathbf{c}\|_2^2$ constitutes the penalty term where $\rho > 0$ is the penalty parameter and $\|\cdot\|_2$ is the Euclidean norm. Instead of solving (3), ADMM works on the ALF by updating $\mathbf{x}$, $\mathbf{y}$ and $\boldsymbol{\lambda}$ in an iterative manner. During the $(k+1)$-th iteration, ADMM executes the following three steps.

$$
\begin{aligned}
\text{Step 1: } \mathbf{x}^{k+1} &= \underset{\mathbf{x}}{\operatorname{argmin}}\ L_\rho\left(\mathbf{x}, \mathbf{y}^k, \boldsymbol{\lambda}^k\right) \\
&= \underset{\mathbf{x}}{\operatorname{argmin}}\ p(\mathbf{x}) + (\boldsymbol{\lambda}^k)^\top \left(A\mathbf{x} + B\mathbf{y}^k - \mathbf{c}\right) \\
&\quad + \frac{\rho}{2}\left\|A\mathbf{x} + B\mathbf{y}^k - \mathbf{c}\right\|_2^2. \\
\text{Step 2: } \mathbf{y}^{k+1} &= \underset{\mathbf{y}}{\operatorname{argmin}}\ L_\rho\left(\mathbf{x}_{k+1}, \mathbf{y}, \boldsymbol{\lambda}^k\right) \\
&= \underset{\mathbf{y}}{\operatorname{argmin}}\ q(\mathbf{y}) + (\boldsymbol{\lambda}^k)^\top \left(A\mathbf{x}^{k+1} + B\mathbf{y} - \mathbf{c}\right) \\
&\quad + \frac{\rho}{2}\left\|A\mathbf{x}^{k+1} + B\mathbf{y} - \mathbf{c}\right\|_2^2. \\
\text{Step 3: } \boldsymbol{\lambda}^{k+1} &= \boldsymbol{\lambda}^k + \rho\left(A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{c}\right).
\end{aligned}
$$

Let us further consider a constrained optimization problem follows the problem formulation in (1):

$$
\begin{array}{ll}
\underset{\mathbf{x}\in\Omega}{\text{minimize}} & f(\mathbf{x}) \\
\text{subject to} & \mathbf{G}(\mathbf{x}) = (\underbrace{0, \ldots, 0}_{\ell})^\top
\end{array}
\tag{5}
$$

According to the formulation in (3), we introduce an auxiliary variable $\mathbf{y}^i \in \Omega$ for $g_i(\mathbf{x})$ where $i \in \{1, \ldots, \ell\}$. Then, (5) can be written into the following separable form:

$$
\begin{array}{ll}
\underset{\mathbf{x}\in\Omega,\ \mathbf{y}^i\in\Omega}{\text{minimize}} & f(\mathbf{x}) + \sum_{i=1}^{\ell}\left[\xi\mathbb{1}\left(g_i\left(\mathbf{y}^i\right) == 0\right)\right] \\
\text{subject to} & \mathbf{x} = \mathbf{y}^i,\ i \in \{1, \ldots, \ell\}
\end{array},
\tag{6}
$$

where $\xi > 0$ is a scaling factor. $\mathbb{1}(\cdot)$ is an indicator function that returns 1 when its argument is true, otherwise it returns 0. Following the formulation in (4), the ALF for (6) is constructed as:

$$
\begin{aligned}
L_\rho(\mathbf{x}, \mathbf{y}^i, \boldsymbol{\lambda}^i) \triangleq f(\mathbf{x}) + \sum_{i=1}^{\ell} &\left[\xi\mathbb{1}\left(g_i\left(\mathbf{y}^i\right) == 0\right)\right. \\
&\left. + \boldsymbol{\lambda}^{i\top}(\mathbf{x} - \mathbf{y}^i) + \frac{\rho}{2}\left\|\mathbf{x} - \mathbf{y}^i\right\|_2^2\right]
\end{aligned}.
\tag{7}
$$

Thereafter, ADMM uses the three-step procedure to solve this ALF in an iterative manner. As discussed in Section 1, ADMM was originally proposed for single-objective optimization. Its extension to CMOP/UC and its synergy with EA will be elaborated in Section 3.

2.3 Literature Review of EAs for CMOPs

Since CMOP/UC have not been studied in the EMO community, we focus on the existing developments of using EAs for CMOPs whose CV information is accessible. In particular, the existing works are organized into the following five categories. Interested readers are referred to some excellent survey papers (e.g., [41–44]) for more details.

The first category mainly leverages the feasibility information during the evolutionary search process. In particular, it always grants feasible solutions a higher priority to survive to the next iteration. For example, Fonseca and Flemming [8] proposed a unified framework for CMOPs that prioritizes the constraint satisfaction over the optimization of objective functions. Coello Coello and Christiansen proposed a naïve constraint handling method that simply dumps the infeasible solutions [9].

The second category mainly augments the CV information to the environmental selection of EMO. In [5], Deb et al. proposed a constrained dominance relation to replace the vanilla Pareto dominance relation. This enables the dominance-based EMO algorithms such as NSGA-II and NSGA-III [10] to be readily applicable to tackle CMOPs. A similar idea is applied as an alternative criterion for updating subproblems in MOEA/D variants [10,45,46]. Besides, the constrained dominance relation is further augmented with other heuristics to provide additional selection pressure to infeasible solutions whose CV values have a marginal difference. For example, the number of violated constraints [11], ϵ-constraint from the classic multi-objective optimization [12, 47, 48], stochastic ranking from the single-objective optimization [13, 49, 50], and the angle between different solutions [14].

In addition to the above feasibility-driven CHTs, the third category is designed to balance the trade-off between convergence and feasibility during the evolutionary search process. One of the most popular methods along this line of research is to use the CV information to augment a penalty term to the objective function. This helps transform the constrained optimization problem into an unconstrained one. In [15], Jiménez et al. proposed a min-max formulation that drives feasible and infeasible solutions to evolve towards optimality and feasibility, respectively. In [16], a Ray-Tai-Seow algorithm was proposed to leverage the interplay of the objective function values and the CV information to compare and rank non-dominated solutions. Based on the similar rigor, some modified ranking mechanisms (e.g., [17, 51, 52]) were developed by taking advantage of the information from both the objective and constraint spaces. Instead of prioritizing feasible solutions, some researchers (e.g., [18, 22, 53]) proposed to exploit useful information from infeasible ones in case they can provide additional diversity to the current evolutionary population.

As a step further, the fourth category aims at simultaneously balancing convergence, diversity, and feasibility during the evolutionary search process. As a pioneer along this line, Li et al. proposed to use two co-evolving and complementary populations, denoted as two-archive EA (C-TAEA), for solving CMOPs [19]. In particular, one archive, denoted as the convergence-oriented archive (CA), pushes the population towards the PF; while the other one, denoted as the diversity-oriented archive, complements the behavior of the CA and provides as much diversified information as possible. After the development of C-TAEA, the multi-population strategy has become a popular algorithmic framework for handling CMOPs (e.g., [20, 54, 55]). Besides, another idea is to transform the CMOP into an unconstrained counterpart by augmenting the constraints as additional objectives. For example, [21] directly uses the convergence, diversity, and feasibility of obtained solutions as the auxiliary objectives to guide the evolutionary population. [22] takes the CV degree as an additional objective and assigns different weights to feasible and infeasible solutions to guide the evolutionary population to move towards the promising regions.

Instead of working on the environmental selection, the fifth category focuses on developing bespoke reproduction operators that repair infeasible solutions. For example, [23] proposed different mutation operators to generate offspring from infeasible and feasible solutions, respectively. Likewise, Xu et al. proposed a differential evolution variant by using an infeasible-guiding mutation for offspring reproduction [24]. In [25], a Pareto descent repair operator was proposed to explore possible feasible solutions along the gradient information around infeasible one in the infeasible region. In [26], a feasible-guided strategy was developed to guide infeasible solutions towards the feasible region along the direction starting from an infeasible solution and ending at its closest feasible solution. In [27], a simulated annealing was applied to accelerate the progress of movements from infeasible solutions towards feasible ones.

The last category consists of some hybrid strategies that otherwise hardly belong to any of the above categories. In [56–58], some mathematical programming methods were applied as a local search to improve the performance of EAs for solving CMOPs with equality constraints. Another idea is to split the evolutionary search process into multiple stages. For example, [59] proposed a push and pull search framework that drives the population to jump over the infeasible region and move towards the PF without considering the constraints at the push stage. Thereafter, an improved ϵ-constraint method is applied to guide the population move towards the feasible PF at the pull stage. In [60], an adaptive two-stage EA was proposed to first drive the population towards the feasible region. Then,

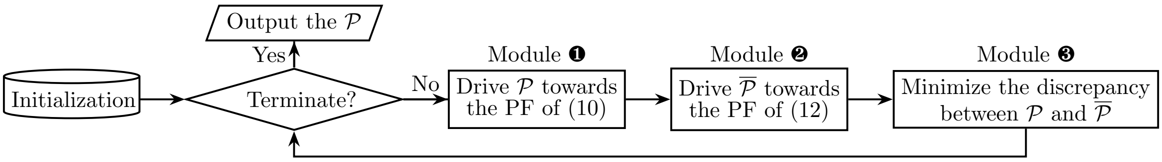

Figure 1: Flowchart of our proposed `EADMM` flowchart.

the population is guided to be spread along the feasible boundary in the second stage. In [27, 61] and [62], surrogate-assisted EAs were proposed to handle problems with computationally expensive objective functions.

Remark 1. *Although the algorithms in the first category do not leverage the CV information during the evolutionary search process, the evolutionary population can easily lose selection pressure when it is filled with infeasible solutions. This is not uncommon when tackling problems with a narrow feasible region. The loss of selection pressure may lead to premature convergence and negatively affect the algorithm's ability to explore the solution space effectively.*

Remark 2. *After the development of the constrained dominance relation [5], the CV information has become one of the building blocks of most, if not all, prevalent CHTs of the EMO algorithms falling into the second and third categories. As reported in our recent study [28], it is surprising to see that the performance of this type of methods is not significantly downgraded when the CV information is not accessible. This can be attributed to the overly simplified infeasible regions that do not pose significant challenges to the existing CHTs. The robustness of these methods might also stem from the inherent adaptability of EAs and their ability to cope with various challenges.*

Remark 3. *As for the repair operators and hybrid strategies belonging to the last two categories, they are also tied with the CV information. In particular, they need to take advantage of either the latent information of CV, such as gradient, to guide the offspring reproduction towards the feasible boundary, or the feasibility information to pull infeasible solutions back to the feasible region. When dealing with unknown constraints, the reliance on CV information may limit the applicability of these strategies and hinder their performance in identifying feasible and optimal solutions.*

3 Proposed Algorithm

The flowchart of our proposed `EADMM` framework is outlined in Fig. 1, similar to `C-TAEA`, `EADMM` maintains two co-evolving populations and consists of three algorithmic components. One (denoted as $\mathcal{P}$) is used to search for the PF of (1) (i.e., Module ❶); while the other one (denoted as $\overline{\mathcal{P}}$) is devoted to exploring the search space without considering any constraint (i.e., Module ❷). Furthermore, a local search is developed to minimize the discrepancy between $\mathcal{P}$ and $\overline{\mathcal{P}}$ (i.e., Module ❸). In the following paragraphs, we first reformulate the CMOP/UC into an additive form as in ADMM. Then, we delineate each algorithmic component of `EADMM` step by step.

3.1 Problem Reformulation

First of all, we rewrite (1) to the following form:

$$\begin{array}{ll} \text{minimize} & \mathbf{p}(\mathbf{x}) + \mathbf{q}(\mathbf{x}), \\ \text{subject to} & \mathbf{G}(\mathbf{x}) = (\underbrace{0, \ldots, 0}_{\ell})^\top, \end{array} \tag{8}$$

where both $\mathbf{p}(\mathbf{x})$ and $\mathbf{q}(\mathbf{x})$ are essentially $\mathbf{F}(\mathbf{x})$. In this case, (8) is equivalent to (1). According to the problem formulation of ADMM in (3), we introduce an auxiliary variable $\mathbf{y} \in \Omega$ to (8) thus it is further rewritten as:

$$\begin{array}{ll}\underset{\mathbf{x}\in\Omega,\ \mathbf{y}\in\Omega}{\text{minimize}} & \mathbf{p}(\mathbf{x})+\mathbf{q}(\mathbf{y})\\ \text{subject to} & \mathbf{G}(\mathbf{x})=(\underbrace{0,\ldots,0}_{\ell})^{\top},\ \mathbf{x}=\mathbf{y}.\end{array} \tag{9}$$

Remark 4. *In (9), the optimization of $\mathbf{p}(\mathbf{x})$ is equivalent to the CMOP/UC defined in (1) while the optimization of $\mathbf{q}(\mathbf{y})$ is the variant of (1) without considering the constraints $\mathbf{G}(\mathbf{x})$.*

Remark 5. *The constraint $\mathbf{x}=\mathbf{y}$ in (9) is designed to leverage the information from the pseudo non-dominated solutions without considering the constraints $\mathbf{G}(\mathbf{x})$ to guide the search for the Pareto-optimal solutions of (1). In this case, we can expect the pseudo non-dominated solutions may be infeasible.*

3.2 Implementation of Module ❶

According to Remark 4, the purpose of Module ❶ shown in Fig. 1 is to guide $\mathcal{P}$ to approximate the PF of the following optimization problem:

$$\begin{array}{ll}\underset{\mathbf{x}\in\Omega}{\text{minimize}} & \mathbf{p}(\mathbf{x})\\ \text{subject to} & \mathbf{G}(\mathbf{x})=(\underbrace{0,\ldots,0}_{\ell})^{\top},\end{array} \tag{10}$$

where $\mathbf{p}(\mathbf{x})$ is $\mathbf{F}(\mathbf{x})$ which makes (10) be equivalent to (1). To handle unknown constraints, we propose to modify the conventional CHTs, mainly based on the CV information as reviewed in Section 2.3, to adapt to the Pareto-, indicator-, and decomposition-based EMO frameworks. As examples, here we choose three iconic algorithms `NSGA-II` [5], `IBEA` [7], and `MOEA/D` [6] as the representative algorithm for each framework, respectively.

3.2.1 Modification on NSGA-II

Here we only need to modify the dominance relation. Specifically, a solution $\mathbf{x}^1$ is said to constraint-dominate $\mathbf{x}^2$, if: 1) $\mathbf{x}^1$ is feasible while $\mathbf{x}^2$ is not; 2) both of them are infeasible while the number of constraints violated by $\mathbf{x}^1$ is less than that of $\mathbf{x}^2$; 3) both of them are infeasible and satisfy the same number of constraints while $\mathbf{x}^1\preceq\mathbf{x}^2$ without considering constraints; or 4) both of them are feasible while $\mathbf{x}^1\preceq\mathbf{x}^2$.

3.2.2 Modification on IBEA

For `IBEA`, we only need to consider feasible solutions in the environmental selection. Note that if none of the offspring solutions are feasible, all parents survive to the next iteration.

3.2.3 Modification on MOEA/D

To adapt `MOEA/D` to handle CMOP/UC, we only need to modify the solution update mechanism. Specifically, given an offspring solution $\mathbf{x}^c$, it can replace the selected parent $\tilde{\mathbf{x}}$ if one of the following four conditions is met: 1) $\mathbf{x}^c$ is feasible while $\tilde{\mathbf{x}}$ is not; 2) both of them are infeasible while the number of constraints violated by $\mathbf{x}^c$ is less than that of $\tilde{\mathbf{x}}$; 3) both of them are infeasible and meet the same number of constraints while $g^{\text{tch}}(\mathbf{x}^c\mid\mathbf{w},\tilde{\mathbf{z}}^*)\leq g^{\text{tch}}(\tilde{\mathbf{x}}\mid\mathbf{w},\tilde{\mathbf{z}}^*)$; or 4) both of them are feasible and $g^{\text{tch}}(\mathbf{x}^c\mid\mathbf{w},\tilde{\mathbf{z}}^*)\leq g^{\text{tch}}(\tilde{\mathbf{x}}\mid\mathbf{w},\tilde{\mathbf{z}}^*)$. In particular, $g^{\text{tch}}(\cdot\mid\mathbf{w},\tilde{\mathbf{z}}^*)$ is the widely used Tchebycheff function [63]:

$$g^{\text{tch}}(\cdot\mid\mathbf{w},\tilde{\mathbf{z}}^*)=\max_{1\leq i\leq m}\frac{|f_i(\cdot)-\tilde{z}^*|}{w_i}, \tag{11}$$

where $\mathbf{w}=(w_1,\ldots,w_m)^\top$ and $\sum_{i=1}^m w_i=1$ is the weight vector of the subproblem associated with $\tilde{\mathbf{x}}$, and $\tilde{\mathbf{z}}^*$ is the estimated ideal point according to the current evolutionary population, where $\tilde{z}_i^*=\min_{\mathbf{x}\in\mathcal{P}\cup\{\mathbf{x}^c\}} f_i(\mathbf{x})$, $i\in\{1,\ldots,m\}$.

Remark 6. *Note that the above modifications are so minor that the corresponding innate algorithmic frameworks are kept intact. In addition, they are applicable to any other variants of `NSGA-II`, `IBEA`, and `MOEA/D` in a plug-in manner.*

3.3 Implementation of Module ❷

The optimization problem considered in Module ❷ is:

$$\underset{\mathbf{y}\in\Omega}{\text{minimize}} \quad \mathbf{q}(\mathbf{y}), \tag{12}$$

where $\mathbf{q}(\mathbf{y})$ is essentially $\mathbf{F}(\mathbf{x})$ as defined in Section 3.1. To solve this problem, any off-the-shelf EMO algorithm can be applied without any modification. Here we choose the vanilla `NSGA-II`, `IBEA`, and `MOEA/D` to keep the algorithmic framework of Module ❷ consistent with Module ❶.

Remark 7. *Module ❷ is designed to enable $\overline{\mathcal{P}}$ to explore the search space as much as possible without being restricted by the feasibility. This helps provide a necessary diversity that promotes the interplay between $\mathcal{P}$ and $\overline{\mathcal{P}}$ as in Module ❸.*

3.4 Implementation of Module ❸

The purpose of Module ❸ is to minimize the discrepancy between $\mathcal{P}$ and $\overline{\mathcal{P}}$ thus to satisfy the constraint $\mathbf{x} = \mathbf{y}$ defined in (9). To this end, we need to take the offspring populations generated in both Module ❶ and Module ❷, denoted as $\mathcal{Q}$ and $\overline{\mathcal{Q}}$ respectively, into consideration. It consists of the following five consecutive steps.

Step 1: Use $\overline{\mathcal{Q}}$ to update $\mathcal{P}$ according to the environmental selection mechanism of the backbone algorithm used in Module ❶.

Step 2: Use $\mathcal{Q}$ to update $\overline{\mathcal{P}}$ according to the environmental selection mechanism of the backbone algorithm used in Module ❷.

Step 3: Create a temporary archive $\hat{\mathcal{S}} = \left\{\hat{\mathbf{x}}^i\right\}_{i=1}^{|\hat{\mathcal{S}}|}$ where $\hat{\mathbf{x}}^i \in \mathcal{Q}$. In addition, $\exists \mathbf{x} \in \mathcal{P} \wedge \exists \overline{\mathbf{x}} \in \overline{\mathcal{P}}$ such that $\hat{\mathbf{x}}^i \preceq \mathbf{x} \wedge \hat{\mathbf{x}}^i \preceq \overline{\mathbf{x}}$ according to Definition 1. The constraints $\mathbf{G}(\mathbf{x}) = \underbrace{(0,\ldots,0)}_{\ell}^{\top}$ are not considered.

Step 4: For each solution $\hat{\mathbf{x}}^i \in \hat{\mathcal{S}}$, solve the following optimization problem:

$$\check{\mathbf{x}}^i = \underset{\mathbf{x}\in\Omega}{\operatorname{argmin}} \sum_{j=1}^{\ell} \mathbb{1}\left(g_j(\mathbf{x}) == 0\right) + \rho \left\|\mathbf{x} - \hat{\mathbf{x}}^i\right\|_2^2 \tag{13}$$

where $i \in \{1, \ldots, |\hat{\mathcal{S}}|\}$ and $\rho = \frac{\gamma}{|\mathcal{P}|} \times \ell$, γ is the number of feasible solutions in $\mathcal{P}$.

Step 5: Use $\check{\mathbf{x}}^i$ where $i \in \{1, \ldots, |\hat{\mathcal{S}}|\}$ to update $\mathcal{P}$ and $\overline{\mathcal{P}}$ according to the environmental selection mechanism of the corresponding backbone algorithm in Modules ❶ and ❷, respectively.

Remark 8. *To elucidate the underpinnings of Module ❸, let us first consider an example shown in Fig. 2(a) where the PF of (10) overlaps with that of (12). Here, $\overline{\mathcal{P}}$ will be impelled towards the PF of (12) under the guidance of Module ❷ (denoted as $\rightsquigarrow$). Meanwhile, $\mathcal{P}$ prioritizes the search for feasible solutions, steered by Module ❶. When the feasible region is very narrow, $\mathcal{P}$ may flounder in the infeasible region (denoted as $\longleftrightarrow$). This can be attributed to the limited selection pressure that can mislead the search towards less promising areas. To mitigate this, Steps 1, 2, and 5 of Module ❸ will drive $\mathcal{P}$ to move towards $\overline{\mathcal{P}}$ (denoted as $\rightsquigarrow$). Since the PF of (10) overlaps with the PF of (12), it signifies that solutions adhering to the condition $\mathbf{x} = \mathbf{y}$ are within the solution space of (12), enabling $\mathcal{P}$ to locate these solutions under the navigation of $\overline{\mathcal{P}}$. Moreover, when an offspring $\hat{\mathbf{x}}$ (denoted as ▲) derived from $\mathcal{P}$ dominates both $\mathcal{P}$ and $\overline{\mathcal{P}}$, Step 4 comes into play, executing a local search centered around $\hat{\mathbf{x}}$.*

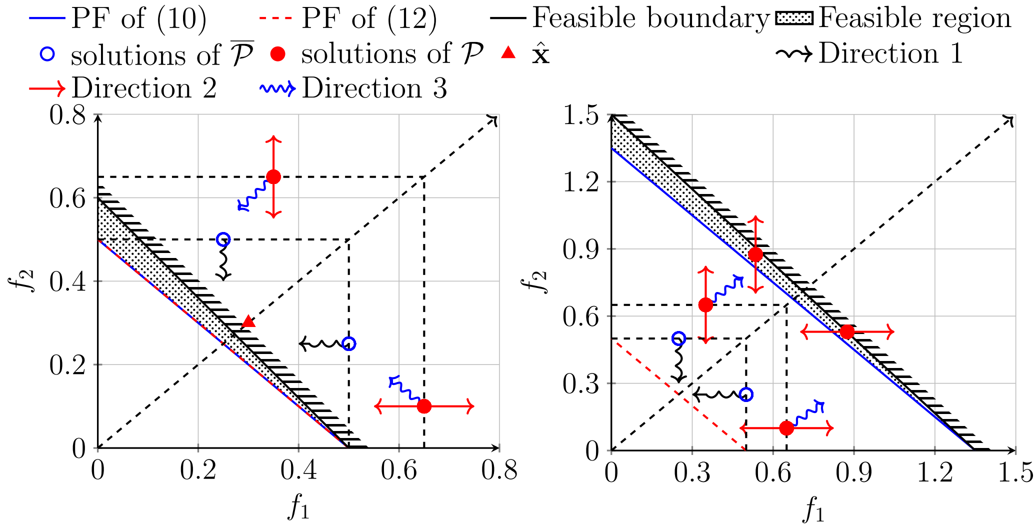

Figure 2: Illustrations of the working mechanism of EADMM/MOEAD, where ⇢ is the weight vector, --- is the contour line of the Tchebycheff function.

Remark 9. *On the other hand, considering the example shown in the Fig. 2(b), wherein the PF of* (10) *does not overlap with* (12)*, the solutions of* $\mathcal{P}$ *located in the feasible region will continue to explore the feasible region guided by Module* ❶ *(denoted as* ⟷*). Whereas the solutions in* $\mathcal{P}$ *that follow* $\overline{\mathcal{P}}$ *and move into the infeasible region will be pulled back to the feasible region by the operations performed in Module* ❶ *(denoted as* ⇝*). In such cases, the function of Steps 1, 2, and 5 in Module* ❸ *is twofold: to track the evolutionary trajectory of* $\overline{\mathcal{P}}$ *and to consider the outcomes of the local search conducted in Step 4. When superior solutions meeting the constraint* $\mathbf{x} = \mathbf{y}$ *are identified, they prompt* $\mathcal{P}$ *to shift towards the area of the search space where these advanced solutions are located.*

Remark 10. *In theory, any off-the-shelf optimization algorithm can be applied to solve* (13)*. In this paper, we choose the genetic algorithm toolbox of the MATLAB 2021a for a proof-of-concept purpose.*

Remark 11. *Step 4 serves as a local search around* $\hat{\mathbf{x}}^i$ *where* $i \in \{1, \ldots, |\hat{\mathcal{S}}|\}$*. This helps search for feasible solutions with regard to (10) and satisfy the constraint* $\mathbf{x} = \mathbf{y}$ *defined in (9).*

Remark 12. *Due to the consideration of the number of feasible solutions in* $\mathcal{P}$*, it makes the local search conducted in the Step 4 be adaptive. Specifically, a smaller* γ *tends to search for feasible solutions in a larger region close to* $\hat{\mathbf{x}}^i$*; whereas a larger* γ *leads to the search of feasible solutions in a smaller region close to* $\hat{\mathbf{x}}^i$ *where* $i \in \{1, \ldots, |\hat{\mathcal{S}}|\}$*.*

3.5 Time Complexity Analysis

The time complexity of our proposed algorithm, which consists of three algorithmic components, is analyzed separately for each module. For both Modules ❶ and ❷, their computational complexity is consistent with their corresponding backbone algorithms. Specifically, the complexity is $\mathcal{O}(mN^2)$ for `NSGA-II`, $\mathcal{O}(N^2)$ for `IBEA`, and $\mathcal{O}(mTN)$ for `MOEA/D`, where N is the size of $\mathcal{P}$ and $\overline{\mathcal{P}}$, and T is the number of weight vectors in the neighborhood of each weight vector of `MOEA/D`. As for Module ❸, Steps 1, 2, and 5 employ the same environmental selection mechanism as Modules ❶ and ❷, resulting in a time complexity that aligns with their corresponding backbone algorithms. In Step 4, since the optimization of (13) is performed $|\hat{\mathcal{S}}|$ times, the computational complexity amounts to $\mathcal{O}\left(|\hat{\mathcal{S}}|\tilde{N}^2\right)$, where $\tilde{N}$ is the population size used in the genetic algorithm. In summary, the overall complexity of our proposed `EADMM` framework is $\max\left\{N^2, \mathcal{O}\left(|\hat{\mathcal{S}}|\tilde{N}^2\right)\right\}$.

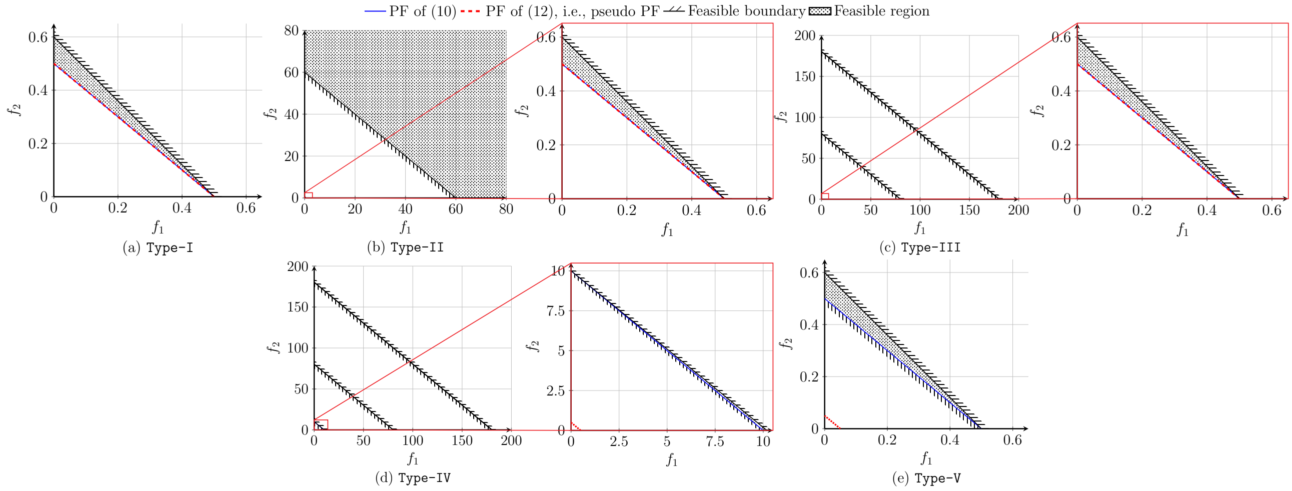

Figure 3: Illustrations for the PFs and the feasible regions of the synthetic test problems built upon C1-DTLZ1. Subfigures (a) to (e) represents the cases ranging from `Type-I` to `Type-V`.

4 Experimental Setup

This section introduces the settings of our empirical study including the benchmark test problems, the peer algorithms, the performance metrics, and the statistical tests.

4.1 Benchmark Test Problems

Our empirical study considers both synthetic test problems and real-world engineering challenges. We will briefly introduce their characteristics in the following paragraphs.

4.1.1 Synthetic Test Problems

As highlighted in Remark 2, existing test problems are insufficient for adequately benchmarking the existing CHTs in EMO for CMOP/UC scenarios. To address this gap, we have developed a new set of synthetic test problems, drawing inspiration from the C-DTLZ benchmark suite [10]. These new problems exhibit the following five types of characteristics:

- `Type-I`: As the illustrative example shown in Fig. 3(a), the feasible region of this type of problem is a narrow area adjacent to the PF, making most of the search space infeasible. Constrained EMO algorithms relying on the CV information may struggle to navigate towards the PF through this predominantly infeasible space.
- `Type-II`: An extension of `Type-I`, this category includes additional feasible regions distant from the PF, as shown in Fig. 3(b). The large gap between these disparate feasible areas often diminishes the selection pressure necessary for evolutionary populations to traverse the regions using binary CV signals.
- `Type-III`: Building upon `Type-I`, this type intersperses the infeasible space with two narrowly spaced feasible regions, as depicted in Fig. 3(c). Without access to CV information, evolutionary populations may become impeded by these narrow feasible areas.
- `Type-IV`: Inspired by C3-DTLZ problems, the PF here is overshadowed by a 'pseudo' PF when constraints are disregarded, as in Fig. 3(d). This presents additional complexities that can misguide an evolutionary population towards the infeasible region above the PF.
- `Type-V`: Sharing the same PF and feasible regions as `Type-I`, this category introduces the complexity of a 'pseudo' PF similar to `Type-IV`, as shown in Fig. 3(e), elevating the challenge.

Remark 13. *Without loss of generality, Fig. 3 only uses C1-DTLZ1 as the baseline to illustrate the above five types of characteristics. Further, considering the stochastic nature of real-world problems, the constraints are with noises in the proposed synthetic test problems. The mathematical definitions and the corresponding visual illustrations of all test problem instances can be found in Section A of Appendix A of the supplemental document.*

4.1.2 Real-world Problems

In addition to the synthetic test problems, we also examine two real-world optimization problems derived from different domains.

- We selected the lunar lander task from Gymnasium[2] as our first test case. This task involves applying reinforcement learning (RL) to control the rocket's landing trajectories. To optimize performance, we consider fine-tuning 12 hyperparameters associated with the RL algorithm. Our objectives are to maximize total rewards while minimizing landing duration. The task includes two unknown constraints: maintaining the lander's awakeness and avoiding crashes.

- The second real-world problem we address concerns water distribution systems (WDS), an indispensable and highly costly component of public infrastructure. The planning and management of WDS typically involve balancing multiple conflicting objectives, such as operational cost, system resilience, and profitability. Our experiments focus on the Pescara network (PES) [64], a well-known benchmark in WDS optimization, which comprises 99 pipes. The optimization goal is to adjust the diameters of these pipes to improve overall system performance. Our objectives are to minimize operational costs and maximize system resilience. Additionally, we consider two unknown constraints: preventing backflow and avoiding pipe blockage.

The detailed mathematical formulation and settings of these two engineering optimization problems can be found in Section B of Appendix A of the supplemental document.

4.2 Peer Algorithms and Parameter Settings

For a proof-of-concept purpose, we use `NSGA-II`, `IBEA`, and `MOEA/D` as the backbone algorithms under our proposed `EADMM` framework. The corresponding `EADMM` instances are denoted as `EADMM/NSGA-II`, `EADMM/IBEA`, and `EADMM/MOEA/D`, respectively. Five high-performance EMO algorithms for CMOPs, including `C-TAEA` [19], `PPS` [59], `MOEA/D-DAE` [65], `ToP` [66], `CMOCSO` [67], are chosen as the peer algorithms. As discussed in Section 2.3, these peer algorithms require accessing to the CV information which is yet available in CMOP/UC. To address this issue, we follow the practice in [28] to replace the CV with a crispy value, i.e., the CV value of a feasible solution is 0; otherwise, it is 1 if the solution is infeasible.

The parameter settings are listed as follows.

- Number of function evaluations (FEs): The maximum number of FEs for different problems are listed in Table I and Table II of Appendix B of the supplemental document. All synthetic test problems are scalable to any number of objectives while we consider $m \in \{2, 3, 5, 10\}$ in our experiments.

- Reproduction operators: The parameters associated with the simulated binary crossover and polynomial mutation are set as $p_c = 1.0$, $\eta_c = 20$, $p_m = \frac{1}{n}$, $\eta_m = 20$. As for those use differential evolution [68] for offspring reproduction, we set $CR = F = 0.5$.

- Algorithm specific parameters: The parameter settings of peer algorithms considered in our experiments are set identical to their original papers. Note that our proposed `EADMM` framework does not introduce any additional parameters, except the innate ones of the backbone algorithms. All parameter settings are detailed in Table III of Appendix B of the supplemental document.

[2] https://gymnasium.farama.org/

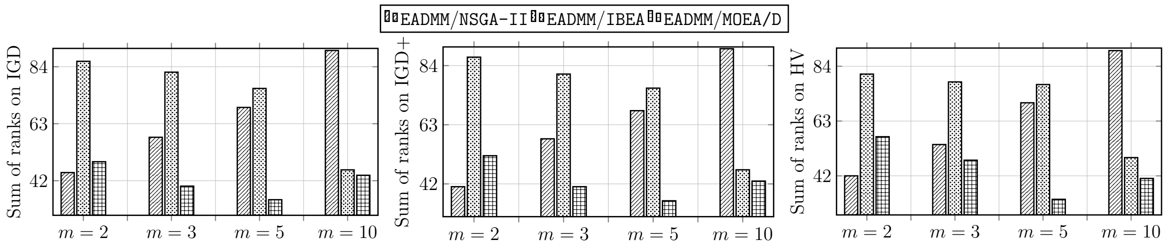

Figure 4: Total Scott-Knott test ranks achieved by each of the three algorithm instances of our proposed framework (the smaller the rank is, the better performance achieved).

- Number of repeated runs: Each algorithm is independently run on each test problem for 31 times with different random seeds.

4.3 Performance Metrics and Statistical Tests

We employ inverted generational distance (IGD) [69], IGD$^+$ [70], and hypervolume (HV) [71] in performance assessment. The reference point used in the HV evaluation is constantly set as $(\underbrace{1.1, \ldots, 1.1}_{m})^\top$. Note that all these three performance metrics can assess both convergence and diversity. The smaller the IGD and IGD$^+$, or the larger the HV, the better result is achieved by the corresponding algorithm.

In view of the stochastic nature of EAs, we use the following three statistical tests to conduct a statistical interpretation of the significance of the comparison results.

- Wilcoxon signed-rank test [72]: This is a non-parametric statistical test that makes no assumption about the underlying distribution of the data. It has been recommended in many empirical studies in the EA community [73]. The significance level is set to $p = 0.05$ in our experiments.

- A_{12} effect size [74]: To ensure the resulted differences are not generated from a trivial effect, we apply A_{12} as the effect size measure to evaluate the probability that one algorithm is better than another. Specifically, given a pair of peer algorithms, $A_{12} = 0.5$ means they are *equal*. $A_{12} < 0.5$ denotes that one is worse for more than 50% of the times. $0.36 \leq A_{12} < 0.44$ indicates a *small* effect size while $0.29 \leq A_{12} < 0.36$ and $A_{12} < 0.29$ mean a *medium* and a *large* effect size, respectively.

- Scott-Knott test: Instead of merely comparing the raw metric values, we apply the Scott-Knott test to rank the performance of different peer algorithms over 31 runs on each test problem. In a nutshell, the Scott-Knott test uses a statistical test and effect size to divide the performance of peer algorithms into several clusters. The performance of peer algorithms within the same cluster is statistically equivalent. The clustering process terminates until no split can be made. Finally, each cluster can be assigned a rank according to the mean metric values achieved by the peer algorithms within the cluster. The smaller the rank is, the better performance of the algorithm achieves.

5 Empirical Studies

In this section, we present and analyze the empirical results of our experiments from three aspects.

5.1 Performance Comparison of Three EADMM Instances

As introduced in Section 4.2, we developed three algorithm instances based on the EADMM framework. The primary goal of our initial experiment is to assess the comparative performance of these instances. We present the statistical comparison results of IGD, IGD$^+$, and HV, based on the Wilcoxon signed-rank test, in Tables IV to XVI in Appendix C of the supplemental document. From these results, it is interesting to observe that no single algorithm consistently excels across all benchmark test problems.

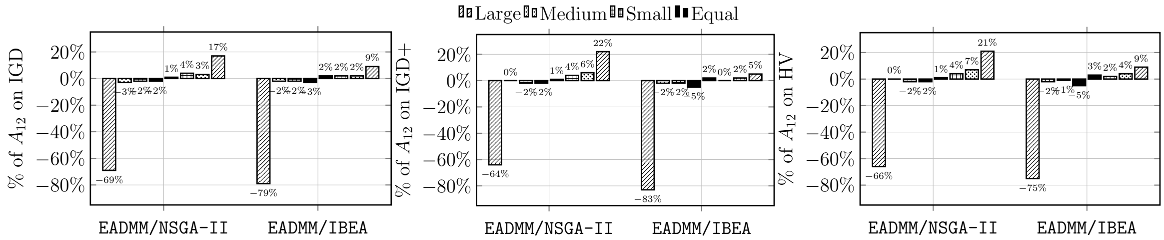

Figure 5: Percentage of the large, medium, small, and equal A_{12} effect size, respectively, when comparing `EADMM/MOEA/D` against `EADMM/NSGA-II` and `EADMM/IBEA`.

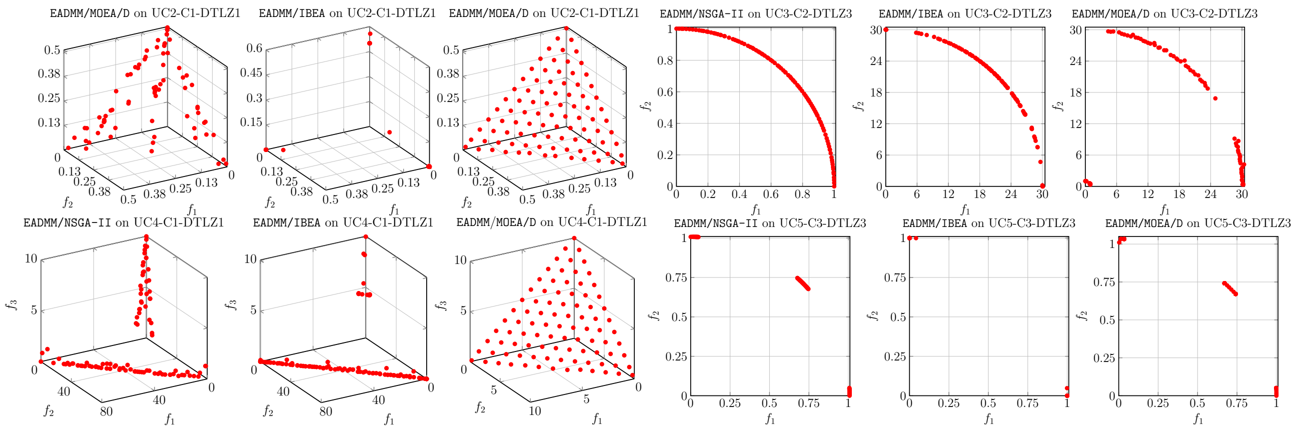

Figure 6: Scatter plots of the final solutions obtained by each of the three algorithm instances of our proposed framework on four synthetic test problems with the median IGD values.

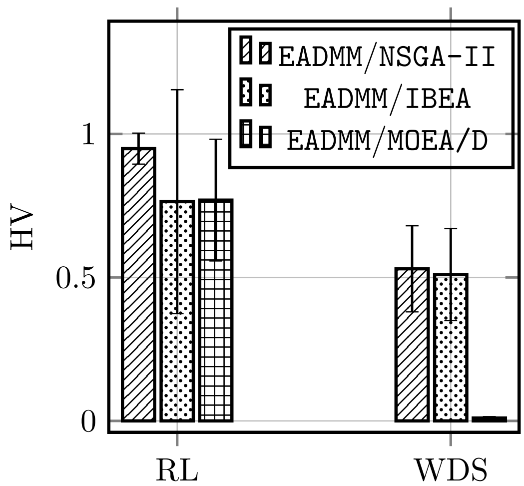

Figure 7: Bar charts with error bars of HV values obtained by each of the three algorithm instances of our proposed framework on two real-world problems.

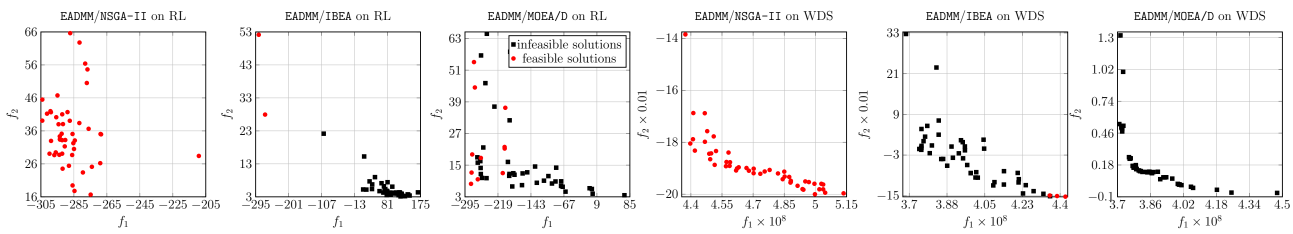

Figure 8: Scatter plots of the final solutions (with the median HV values) obtained by `EADMM/NSGA-II`, `EADMM/IBEA`, and `EADMM/MOEA/D` on two real-world problems.

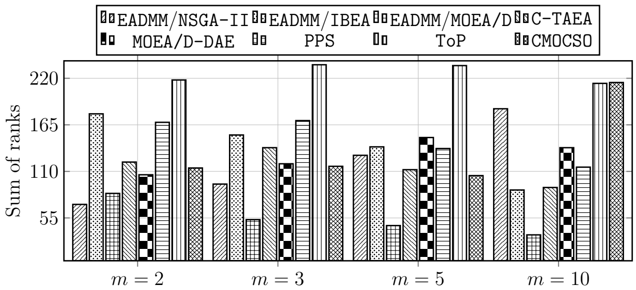

Figure 9: Total Scott-Knott test ranks achieved by each of the three proposed algorithms and other five state-of-the-art peer algorithms (the smaller the rank is, the better performance achieved).

To better understand the performance rankings among these three instances, we employ the Scott-Knott test. This test categorizes them into different groups based on IGD, IGD$^+$, and HV values for each synthetic test problem. We opted not to list all ranking results ($30 \times 4 \times 3 = 360$ in total) to avoid clutter. Instead, we summarize these results in bar charts in Fig. 5, providing an overview of their comparative performance. Notably, `EADMM/MOEA/D` emerges as the top-performing instance in 80% of comparisons. `EADMM/IBEA` and `EADMM/NSGA-II` also show strong performances in specific scenarios, although the latter's effectiveness diminishes with an increasing number of objectives.

To facilitate visual comparisons, we select some plots of the final solutions obtained by each of the three algorithm instances of our proposed framework on four synthetic test problems in Fig. 6[3]. It is clear to see that the solutions found by `EADMM/MOEA/D` exhibit the best convergence and diversity on the 3-objective UC2-C1-DTLZ1 and UC4-C1-DTLZ1 problems. However, on 2-objective UC3-C2-DTLZ3 and UC5-C3-DTLZ3 problems, the convergence and diversity of solutions found by `EADMM/MOEA/D` are worse than that found by `EADMM/NSGA-II`. This means that, on the one hand, the proposed `EADMM` framework effectively guides the populations through the infeasible regions and avoids them being misled by the pseudo-PF. On the other hand, the performance of algorithms instances of `EADMM` is significantly affected by the backbone algorithms.

Due to the absence of ground truth PF for the real-world engineering problems, we base our performance comparisons on the HV metric. The bar charts with error bars in Fig. 7 show that `EADMM/NSGA-II` is the most competitive. We also plot the final solutions, both feasible and infeasible, in Fig. 8. For the lunar lander task, `EADMM/NSGA-II` excels in guiding the population toward feasible regions, while more than half of the final solutions obtained by `EADMM/IBEA` and `EADMM /MOEA/D` are located in the infeasible region. As for the WDS planning and management task, all solutions obtained by `EADMM/NSGA-II` are feasible. Although `EADMM/IBEA` finds some feasible solutions, most of its population are still located in the infeasible region. In contrast, `EADMM/MOEA/D` fails to find any feasible solution. This can be attributed to the disparate scales of the two objective functions, where f_1 is on the order of 10^9 larger than f_2. Such an imbalance skews the evolutionary direction toward f_1, thereby sacrificing the population diversity and inadvertently steering the search into infeasible regions.

5.2 Performance Comparisons with Other Five State-of-the-art Peer Algorithms

In this subsection, we compare the performance of all our three `EADMM` instances against the five state-of-the-art peer algorithms introduced in Section 4.2. Similar to the practice conducted in Section 5.1, we first employ the Wilcoxon signed-rank test to compare the IGD, IGD$^+$, and HV results of each of our three algorithm instances with regard to the other five state-of-the-art peer algorithms. The statistical results are given in Tables IV to XVI of Appendix C of our supplemental document. It is interesting to note that the comparison results align with our observation in Section 5.1. In a nut shell, we find that `EADMM/MOEA/D` is the best algorithm on the synthetic test problems while it obtains

[3]Full results of visual comparisons can be found in Fig.6 to Fig.17 of Section B of Appendix C of the supplementary document.

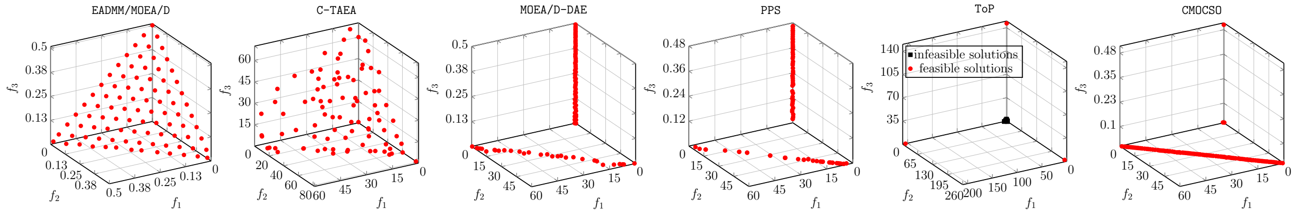

Figure 10: Final solutions found by different algorithms with the median IGD values on three objective UC2-C2-DTLZ1 test problems.

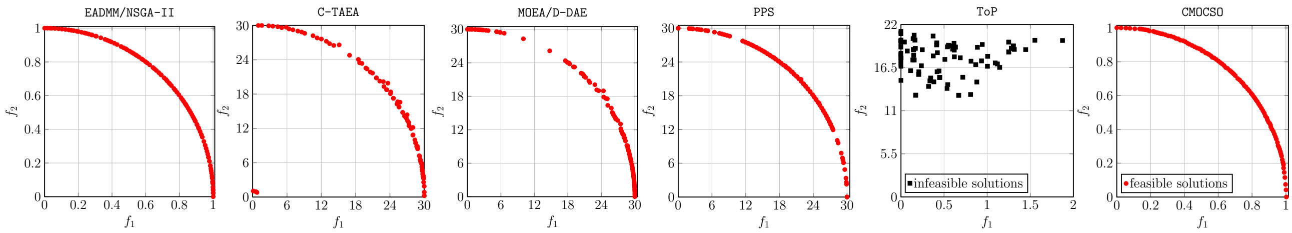

Figure 11: Final solutions found by different algorithms with the median IGD values on two objective UC3-C2-DTLZ3 test problems.

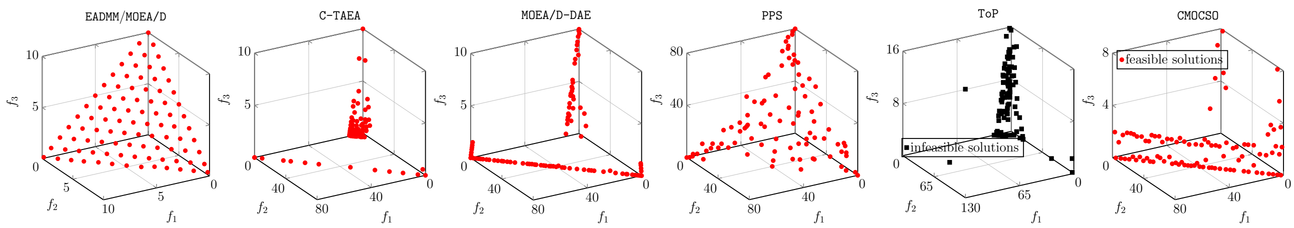

Figure 12: Final solutions found by different algorithms with the median IGD values on three objective UC4-C1-DTLZ1 test problems.

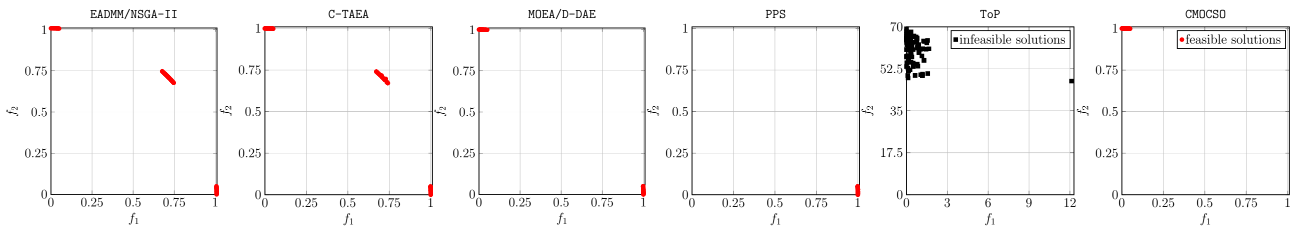

Figure 13: Final solutions found by different algorithms with the median IGD values on two objective UC5-C3-DTLZ3 test problems.

significantly better metric values in 80% comparisons. As for the two real-world engineering problems, `EADMM/NSGA-II` is the most competitive one.

As done in Section 5.1, we apply the Scott-Knott test to sort the performance of all algorithms on the synthetic test problem instance across different number of objectives. To facilitate a better interpretation of these massive comparison results, we summarize the test results obtained across all test problem instances for each algorithm and show them as the bar charts in Fig. 9. From this result, we find that two of our proposed `EADMM` instances secure the top two positions in almost all comparisons. This validates the superiority of the proposed `EADMM` framework. Specifically, like the observations in Section 5.1, `EADMM/NSGA-II` is the best algorithm when $m = 2$ while `EADMM/MOEA/D` wins in other cases. Note that although `EADMM/IBEA` is relatively less competitive than its two siblings, it still is ranked in the second and third place on $m = 10$ and $m = 5$ test problem instances respectively.

To have a better visual interpretation of the superiority achieved by our algorithm instances, let us look into the population distribution of the final solutions against the other five peer algorithms in Fig. 10 to Fig. 13. Due to space limitations, only selected examples are included here, with complete results available in the supplemental document[4]. These plots reveal that the solutions from `EADMM/MOEA/D` and `EADMM/NSGA-II` demonstrate notable superiority over the other five peer algorithms. We will now briefly analyze the potential reasons behind this performance.

- From the second columns of Fig. 10 to Fig. 13, it is evident that `C-TAEA` demonstrates superior convergence in UC5-C3-DTLZ3 compared to the other three problems. This superior performance can be attributed to the UC5-C3-DTLZ3's feasible region being adjacent to the PF. Here, the convergence-oriented archive of `C-TAEA` is effectively steered towards the feasible region, aided by its diversity-oriented archive. In contrast, the other three problems feature multiple feasible regions not adjacent to the PF. Without CV guidance, the convergence-oriented archive's solutions are still directed towards the feasible regions. However, the update mechanism in the CA tends to enhance solution diversity within each feasible region. As a result, the convergence in these problems is not as pronounced as in UC5-C3-DTLZ3.

- The data in the third columns of Fig. 10 to Fig. 13 show that `MOEA/D-DAE` struggles with converging populations to the feasible regions on the PF for the first three problems. It only manages to locate solutions in some local feasible regions of the UC5-C3-DTLZ3. This issue arises because `MOEA/D-DAE` depends on the rate of change in CV to assess the evolutionary state of populations. It utilizes an ε-based CHT to direct populations towards feasible areas. However, in the absence of CV information, the algorithm's detect-and-escape strategy finds it challenging to accurately discern the evolutionary state. Consequently, the ε-based CHT inadvertently leads populations to linger in feasible regions that are not on the PF.

- Examining the fourth columns of Fig. 10 to Fig. 13 we notice that the performance of `PPS` mirrors that of `MOEA/D-DAE`. This similarity is largely because `PPS` also utilizes an ε-based CHT that depends on CV information. Specifically, for the first three problems, `PPS` prematurely transitions to the pull stage, which happens before the populations reach the feasible regions on the PF. Lacking CV information, the CHT mistakenly pulls the infeasible solutions back to a non-PF feasible region. In the case of UC5-C3-DTLZ3, while `PPS` identifies solutions in some local feasible regions during the push stage, the transition to the pull stage proves ineffective. Without CV guidance, the algorithm struggles to explore other feasible regions.

- As illustrated in the fifth columns of Fig. 10 to Fig. 13, `ToP` fails to find feasible solutions for the last three problems. This limitation stems from the algorithm's reliance on CV information during the single-objective optimization stage. In the absence of CV data, the populations are erroneously directed towards infeasible regions. Consequently, when `ToP` transitions to the multi-objective optimization stage, the populations continue gravitating towards areas with better objective function fitness values, albeit infeasible. This misdirection makes it challenging for the populations to locate any feasible solutions.

- As we can see from the sixth columns of Fig. 10 to Fig. 13, although `CMOCSO` outperforms the other four peer algorithms, it still does not successfully find solutions in all the local feasible regions of UC5-C3-DTLZ3. Moreover, it struggles to converge all solutions to the feasible regions on the PF in both UC2-C1-DTLZ1 and UC4-C1-DTLZ1. This limitation is mainly due to the algorithm's reliance on the ε-based CHT. When deprived of CV information, the CHT may lead to the selection of misleading winner particles. Consequently, `CMOCSO`'s cooperative swarm optimizer faces difficulties in maintaining both the diversity and convergence of the particles.

As we did in Section 5.1, we compare the HV values of our three `EADMM` instances against five state-of-the-art peer algorithms on two real-world engineering problems. The statistical results are presented as bar charts with error bars in Fig. 14. Additionally, the final solutions corresponding to the median HV values are illustrated in Fig. 15. From the bar charts in Fig. 14, it is evident that `EADMM/NSGA-II`

[4]See footnote 3

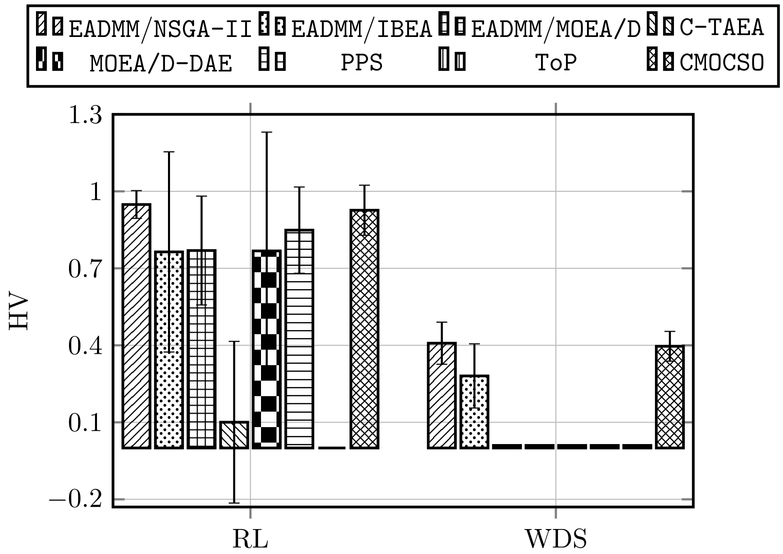

Figure 14: Bar charts with error bars of HV values obtained by each of the three algorithm instances of our proposed framework and other five state-of-the-art algorithms.

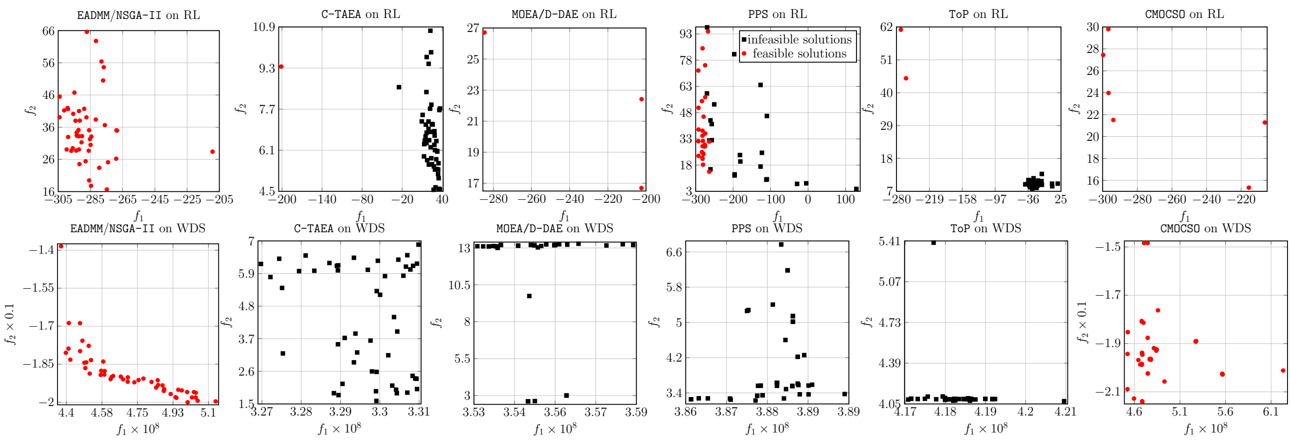

Figure 15: Final solutions found by different algorithms with the median HV values on real-world problems.

achieves the best HV values in both the lunar lander and the WDS planning and management tasks. This performance is further supported by the scatter plots in Fig. 15, where `EADMM/NSGA-II` clearly outperforms other algorithms. Specifically, for the lunar lander task, all solutions by `EADMM/NSGA-II` are feasible. In the WDS task, while both `EADMM/NSGA-II` and `CMOCSO` successfully guide the population towards the feasible region, the prior one demonstrates superior convergence and diversity. In contrast, algorithms such as `C-TAEA`, `MOEA/D-DAE`, `PPS`, and `ToP` fail to find any feasible solutions. We will briefly analyze the potential reasons as follows.

- As MOEA/D variants, `C-TAEA`, `MOEA/D-DAE`, and `PPS` struggle with significantly disparate objective functions, as discussed in Section 5.1. The scatter plots in Fig. 15 show that these algorithms tend to optimize f_1 at the expense of solution diversity. Consequently, their populations gravitate towards regions where f_1 is more optimal but infeasible.

- `ToP` is a two-stage evolutionary algorithm that initially emphasizes single-objective optimization, primarily focusing on the objective with higher fitness improvement. In the WDS planning and management task, this approach results in a biased selection pressure towards the first objective. Such bias leads to reduced population diversity and drives the population away from feasible regions. Therefore, when `ToP` transitions to the multi-objective optimization stage, it encounters difficulties in locating feasible solutions due to the diminished diversity.

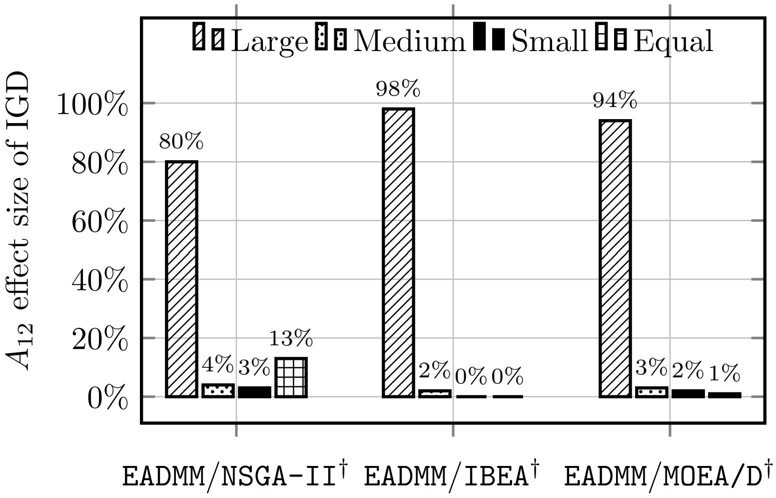

Figure 16: Percentage of the large, medium, small, and equal A12 effect size of IGD.

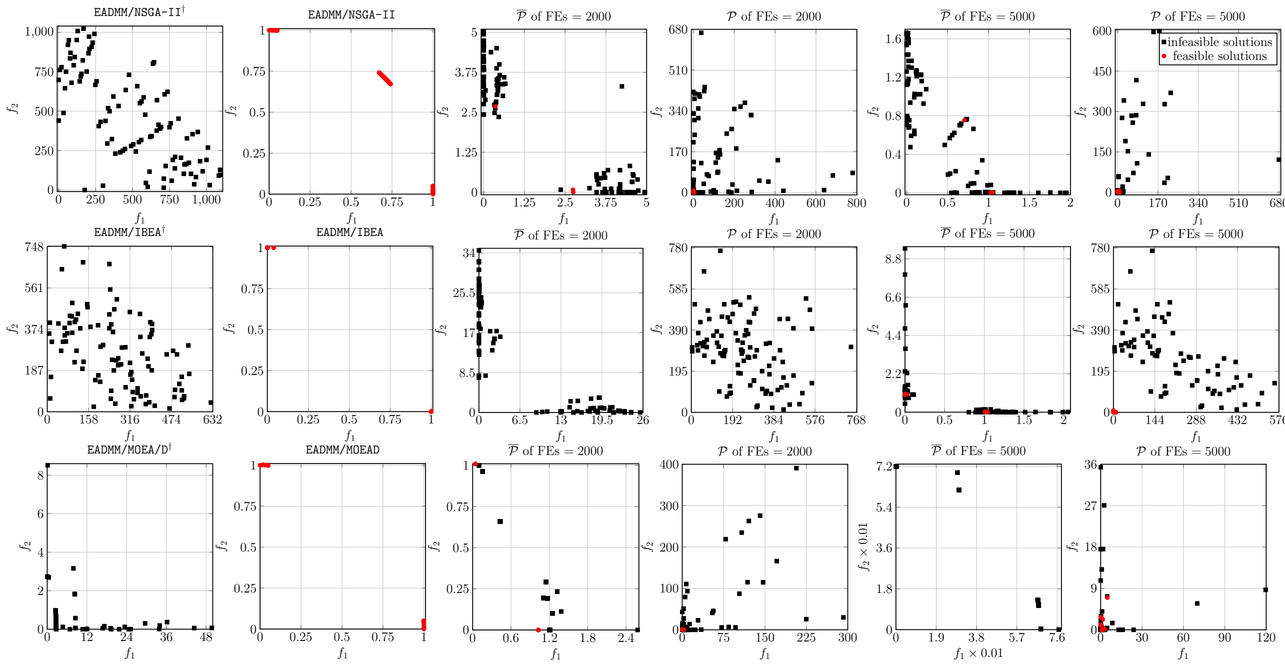

Figure 17: The first column plots are the final solutions found by `EADMM/NSGA-II`†, `EADMM/IBEA` † and `EADMM/MOEA/D` † with the median IGD values on UC4-C3-DTLZ2 respectively. The last five column plots are the solutions found by the corresponding algorithm instances under our proposed `EADMM` framework, the $\overline{\mathcal{P}}$ and $\mathcal{P}$ obtained by these instances using different FEs respectively.

5.3 Ablation study with regard to the EADMM

The empirical study presented in Section 5.2 demonstrates the superior performance of our proposed `EADMM` framework compared to five state-of-the-art EMO algorithms. As illustrated in Fig. 1, Modules ❷ and ❸ are the key distinguishing elements of our framework. We plan to assess the effectiveness of these modules through an ablation study. This study will involve comparing the performance of the complete `EADMM` framework against a version that exclusively uses Module ❶, denoted by the † symbol.

Based on the Wilcoxon signed-rank test, the statistical comparison of IGD, IGD$^+$, and HV values, presented in Tables XVII to XXVIII of Appendix C of the supplementary document, alongside the A_{12} effect size depicted in Fig. 16, indicates a consistent performance degradation when Modules ❷ and ❸ are ablated. As an illustrative example, Fig. 17 demonstrates this point. The ablated versions, `EADMM/NSGA-II`†, `EADMM/IBEA`†, and `EADMM/MOEA/D`†, all fail to find feasible solutions on UC4-C3-DTLZ2. In contrast, the full versions of `EADMM/NSGA-II`, `EADMM/IBEA`, and `EADMM/MOEA/D` successfully drive their populations to converge into the feasible regions. The reasons for this phenomenon are explained as follows.

- As we can see from the last four columns of plots in Fig. 17, it is noticeable that the solutions in $\overline{\mathcal{P}}$ exhibit better objective function fitness compared to $\mathcal{P}$. Additionally, $\mathcal{P}$ at $FEs = 5000$ shows improved feasibility and objective function fitness over the final solutions of the algorithms marked with $\dagger$, as seen in the first column of plots. This observation suggests that $\overline{\mathcal{P}}$ effectively guides $\mathcal{P}$ towards the PF, in line with the discussion in Remark 8 and as illustrated in the left plot of Fig. 2.
- In the fourth column of plots in Fig. 17, some solutions of $\overline{\mathcal{P}}$ are in the infeasible region, showing better objective function fitness than those in feasible areas. However, the final solutions of $\mathcal{P}$, depicted in the second column of plots, are all within feasible regions. This indicates that solutions in $\mathcal{P}$, initially following $\overline{\mathcal{P}}$ into the infeasible region, are eventually redirected back to the feasible areas. This behavior is described in Remark 9 and shown in the right plot of Fig. 2.

Furthermore, in Fig. 17, we observe that the solutions of different algorithms have shown varying degrees of convergence and diversity. The potential reasons for these phenomena are discussed as follows.

- Focusing on the first column of plots in Fig. 17, we see that the solutions by `EADMM/MOEA/D`† show the best convergence in the objective space. This is mainly because Module ❶ in `EADMM/MOEA/D` prioritizes infeasible solutions with better objective function fitness, provided they meet the same number of constraints. In contrast, Module ❶ in `EADMM/IBEA` treats all infeasible solutions equally, while `EADMM/NSGA-II` prioritizes infeasible solutions based on the number of met constraints. The convergence advantage of `EADMM/MOEA/D` is also evident in the $\overline{\mathcal{P}}$ and $\mathcal{P}$ in the third row of plots, enhancing its performance in synthetic test problems with many objectives.
- In the first row of plots in Fig. 17, the $\overline{\mathcal{P}}$ and $\mathcal{P}$ of `EADMM/NSGA-II` demonstrate greater diversity compared to `EADMM/IBEA` and `EADMM/MOEA/D`. Consequently, `EADMM/NSGA-II` is the only algorithm among those in the second column of plots to successfully locate feasible solutions in all feasible regions on the PF. This can be attributed to its crowding distance assignment mechanism, which prevents clustering of solutions in the objective space. Such diversity grants `EADMM/NSGA-II` exceptional performance in real-world problems and synthetic test problems in the 2-objective cases.

6 Conclusions and Future Directions

CMOP/UC are important in various real-world optimization scenarios, particularly in safety-critical situations. Despite their significance, CMOP/UC has been significantly underexplored in the EMO community. In this paper, we bridge this gap by synergizing traditional ADMM with evolutionary meta-heuristics to propose the `EADMM` framework. This approach is simple yet effective in tackling CMOP/UC challenges. Our comprehensive experiments, conducted on a range of synthetic test problems and real-world engineering optimization scenarios, validate the robustness and effectiveness of the `EADMM` framework. Further, our ablation study also demonstrates the vital roles of the ADMM problem formulation (i.e., Modules ❷ and ❸ in `EADMM`), reinforcing the framework's performance capabilities. As a pioneering work in the field, we had an eureka moment that the implications of CMOP/UC and the `EADMM` framework extend far beyond the scope of this paper, there are several open questions remained.

- *Theoretical analysis*: The theoretical properties of ADMM have been rigorously studied in the class optimization literature. It will be interesting to leverage the mathematical underpinnings of ADMM to gain a deeper theoretical understanding of `EADMM` in terms of its optimization behavior and convergence properties.

- *Scalability studies*: Further research into the scalability of the `EADMM` framework, particularly in handling problems with a large number of variables, would be beneficial. This could include the development of parallelized or distributed versions of the framework to leverage computational resources more effectively.
- *Real-world applications*: Expanding the application of the `EADMM` framework to more diverse and complex real-world problems, such as large-scale environmental modeling or advanced engineering design, would be an exciting avenue. This requires a multi-disciplinary collaboration to ensure our research to be aligned with the practical needs and challenges of different domains.

Acknowledgment

K. Li was supported in part by the UKRI Future Leaders Fellowship under Grant MR/S017062/1 and MR/X011135/1; in part by NSFC under Grant 62376056 and 62076056; in part by the Royal Society under Grant IES/R2/212077; in part by the EPSRC under Grant 2404317; in part by the Kan Tong Po Fellowship (KTP\R1\231017); and in part by the Amazon Research Award and Alan Turing Fellowship. S. Li, W. Li, and M. Yang were supported by the National Natural Science Foundation of China under Grant 62273119.

References

[1] D. L. Thurston and S. Srinivasan, "Constrained optimization for green engineering decision-making," *Environ. Sci. Technol.*, vol. 37, no. 23, pp. 5389–5397, 2003.

[2] J. Andersson, "Applications of a multi-objective genetic algorithm to engineering design problems," in *EMO'03: Proc. of the Second International Conference on Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, vol. 2632, 2003, pp. 737–751.

[3] A. Ponsich, A. L. Jaimes, and C. A. C. Coello, "A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 321–344, 2013.

[4] J. García and F. Fernández, "A comprehensive survey on safe reinforcement learning," *J. Mach. Learn. Res.*, vol. 16, pp. 1437–1480, 2015.

[5] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[6] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.

[7] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *PPSN'04: Proc. of 8th International Conference on Parallel Problem Solving from Nature*, 2004, pp. 832–842.

[8] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms. i. A unified formulation," *IEEE Trans. Syst. Man Cybern., Part A*, vol. 28, no. 1, pp. 26–37, 1998.

[9] C. A. C. Coello and A. D. Christiansen, "MOSES: A multiobjective optimization tool for engineering design," *Eng. Opt.*, vol. 31, no. 3, pp. 337–368, 1999.

[10] H. Jain and K. Deb, “An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: handling constraints and extending to an adaptive approach,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, 2014.

[11] A. Oyama, K. Shimoyama, and K. Fujii, “New constraint-handling method for multi-objective and multi-constraint evolutionary optimization,” *Trans. Jpn. Soc. Aeronaut*, vol. 50, pp. 56–62, 2007.

[12] T. Takahama and S. Sakai, “Efficient constrained optimization by the ϵ constrained rank-based differential evolution,” in *CEC’12: Proc. of the 2012 IEEE Congress on Evolutionary Computation*, 2012, pp. 1–8.

[13] H. Geng, M. Zhang, L. Huang, and X. Wang, “Infeasible elitists and stochastic ranking selection in constrained evolutionary multi-objective optimization,” in *SEAL’06: Proc. of the 2006 Simulated Evolution and Learning International Conference*, vol. 4247, 2006, pp. 336–344.

[14] Z. Fan, W. Li, X. Cai, K. Hu, H. Lin, and H. Li, “Angle-based constrained dominance principle in MOEA/D for constrained multi-objective optimization problems,” in *CEC’16: Proc. of the 2016 IEEE Congress on Evolutionary Computation*, 2016, pp. 460–467.

[15] F. Jiménez, A. F. Gómez-Skarmeta, G. Sánchez, and K. Deb, “An evolutionary algorithm for constrained multi-objective optimization,” in *CEC’02: Proc. of the 2002 IEEE Congress on Evolutionary Computation*, 2002, pp. 1133–1138.

[16] T. Ray, K. Tai, and K.-C. Seow, “Multiobjective design optimization by an evolutionary algorithm,” *Eng. Opt.*, vol. 33, no. 4, pp. 399–424, 2001.

[17] N. Young, “Blended ranking to cross infeasible regions in constrained multi-objective problems,” in *CIMCA’05: Proc. of the 2005 International Conference on Computational Intelligence Modeling, Control and Automation*, 2005, pp. 191–196.

[18] K. Li, K. Deb, Q. Zhang, and S. Kwong, “An evolutionary many-objective optimization algorithm based on dominance and decomposition,” *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, 2015.

[19] K. Li, R. Chen, G. Fu, and X. Yao, “Two-archive evolutionary algorithm for constrained multi-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 303–315, 2019.

[20] Y. Tian, T. Zhang, J. Xiao, X. Zhang, and Y. Jin, “A coevolutionary framework for constrained multiobjective optimization problems,” *IEEE Trans. Evol. Comput.*, vol. 25, no. 1, pp. 102–116, 2021.

[21] Q. Long, “A constraint handling technique for constrained multi-objective genetic algorithm,” *Swarm Evol. Comput.*, vol. 15, pp. 66–79, 2014.

[22] C. Peng, H. Liu, and F. Gu, “An evolutionary algorithm with directed weights for constrained multi-objective optimization,” *Appl. Soft Comput.*, vol. 60, pp. 613–622, 2017.

[23] X. Yu, X. Yu, Y. Lu, G. G. Yen, and M. Cai, “Differential evolution mutation operators for constrained multi-objective optimization,” *Appl. Soft Comput.*, vol. 67, pp. 452–466, 2018.

[24] B. Xu, W. Duan, H. Zhang, and Z. Li, “Differential evolution with infeasible-guiding mutation operators for constrained multi-objective optimization,” *Appl. Intell.*, vol. 50, no. 12, pp. 4459–4481, 2020.

[25] K. Harada, J. Sakuma, I. Ono, and S. Kobayashi, “Constraint-handling method for multi-objective function optimization: Pareto descent repair operator,” in *EMO’06: Proc. of the 4th International Conference on Evolutionary Multi-Criterion Optimization*, 2006, pp. 156–170.

[26] L. Jiao, J. Luo, R. Shang, and F. Liu, "A modified objective function method with feasible-guiding strategy to solve constrained multi-objective optimization problems," *Appl. Soft Comput.*, vol. 14, pp. 363–380, 2014.

[27] H. K. Singh, T. Ray, and W. Smith, "C-PSA: constrained Pareto simulated annealing for constrained multi-objective optimization," *Inf. Sci.*, vol. 180, no. 13, pp. 2499–2513, 2010.

[28] S. Li, K. Li, and W. Li, "Do we really need to use constraint violation in constrained evolutionary multi-objective optimization?" in *PPSN'22: Proc. of the 17th International Conference on Parallel Problem Solving from Nature*, ser. Lecture Notes in Computer Science, vol. 13399, 2022, pp. 124–137.

[29] R. Garnett, *Bayesian Optimization.* Cambridge University Press, 2023.

[30] M. Schonlau, W. J. Welch, and D. R. Jones, "Global versus local search in constrained optimization of computer models," *Lect. Notes Ser.*, pp. 11–25, 1998.

[31] M. A. Gelbart, J. Snoek, and R. P. Adams, "Bayesian optimization with unknown constraints," in *UAI'14: Proc. of the 30th Conference on Uncertainty in Artificial Intelligence*, 2014, pp. 250–259.

[32] J. R. Gardner, M. J. Kusner, Z. E. Xu, K. Q. Weinberger, and J. P. Cunningham, "Bayesian optimization with inequality constraints," in *ICML'14: Proc. of the 31nd International Conference on Machine Learning*, vol. 2014, 2014, pp. 937–945.

[33] J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, "Optimization under unknown constraints," *Bayesian Stat.*, vol. 9, no. 9, p. 229, 2011.

[34] V. Picheny, "A stepwise uncertainty reduction approach to constrained global optimization," in *AISTATS'14: Proc. of the 2014 International Conference on Artificial intelligence and statistics*, 2014, pp. 787–795.

[35] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani, "Predictive entropy search for efficient global optimization of black-box functions," in *NIPS'14: Proc. of the 2014 Annual Conference on Neural Information Processing Systems*, 2014, pp. 918–926.

[36] J. M. Hernández-Lobato, M. A. Gelbart, M. W. Hoffman, R. P. Adams, and Z. Ghahramani, "Predictive entropy search for bayesian optimization with unknown constraints," in *ICML'15: Proc. of the 32nd International Conference on Machine Learning*, vol. 37, 2015, pp. 1699–1707.

[37] R. B. Gramacy, G. A. Gray, S. Le Digabel, H. K. Lee, P. Ranjan, G. Wells, and S. M. Wild, "Modeling an augmented lagrangian for blackbox constrained optimization," *Technometrics*, vol. 58, no. 1, pp. 1–11, 2016.

[38] V. Picheny, R. B. Gramacy, S. M. Wild, and S. L. Digabel, "Bayesian optimization under mixed constraints with a slack-variable augmented lagrangian," in *NIPS'16: Proc. of the 2014 Annual Conference on Neural Information Processing Systems*, 2016, pp. 1435–1443.

[39] S. Ariafar, J. Coll-Font, D. H. Brooks, and J. G. Dy, "ADMMBO: bayesian optimization with unknown constraints using ADMM," *J. Mach. Learn. Res.*, vol. 20, pp. 123:1–123:26, 2019.

[40] S. P. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[41] Z. Fan, Y. Fang, W. Li, J. Lu, X. Cai, and C. Wei, "A comparative study of constrained multi-objective evolutionary algorithms on constrained multi-objective optimization problems," in *CEC'17: Proc. of the 2017 IEEE Congress on Evolutionary Computation*, 2017, pp. 209–216.

REFERENCES

[42] R. Tanabe and A. Oyama, "A note on constrained multi-objective optimization benchmark problems," in *CEC'17: Proc. of the 2017 IEEE Congress on Evolutionary Computation*, 2017, pp. 1127–1134.

[43] F. Snyman and M. Helbig, "Solving constrained multi-objective optimization problems with evolutionary algorithms," in *ICSI'17: Proc. of the 2017 International Conference on Swarm Intelligence*, 2017, pp. 57–66.

[44] J. Liang, X. Ban, K. Yu, B. Qu, K. Qiao, C. Yue, K. Chen, and K. C. Tan, "A survey on evolutionary constrained multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 27, no. 2, pp. 201–221, 2022.

[45] M. A. Jan and Q. Zhang, "MOEA/D for constrained multiobjective optimization: Some preliminary experimental results," in *UKCI'10: Proc. of the 2010 UK Workshop on Computational Intelligence*, 2010, pp. 1–6.

[46] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, 2016.

[47] S. Z. Martínez and C. A. C. Coello, "A multi-objective evolutionary algorithm based on decomposition for constrained multi-objective optimization," in *CEC'14: Proc. of the 2014 IEEE Congress on Evolutionary Computation*, 2014, pp. 429–436.

[48] M. Asafuddoula, T. Ray, and R. A. Sarker, "A decomposition-based evolutionary algorithm for many objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 445–460, 2015.

[49] W.-Q. Ying, W.-P. He, Y.-X. Huang, D.-T. Li, and Y. Wu, "An adaptive stochastic ranking mechanism in moea/d for constrained multi-objective optimization," in *ISAI'16: Proc. of the 2016 International Conference on Information System and Artificial Intelligence*, 2016, pp. 514–518.

[50] Z. Liu, Y. Wang, and B. Wang, "Indicator-based constrained multiobjective evolutionary algorithms," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 51, no. 9, pp. 5414–5426, 2019.

[51] A. Angantyr, J. Andersson, and J. Aidanpaa, "Constrained optimization based on a multiobjective evolutionary algorithm," in *CEC'03: Proc. of the 2003 IEEE Congress on Evolutionary Computation*, 2003, pp. 1560–1567.

[52] Y. G. Woldesenbet, G. G. Yen, and B. G. Tessema, "Constraint handling in multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 514–525, 2009.

[53] A. E. Sorkhabi, M. D. Amiri, and A. R. Khanteymoori, "Duality evolution: an efficient approach to constraint handling in multi-objective particle swarm optimization," *Soft Comput.*, vol. 21, no. 24, pp. 7251–7267, 2017.

[54] J. Wang, Y. Li, Q. Zhang, Z. Zhang, and S. Gao, "Cooperative multiobjective evolutionary algorithm with propulsive population for constrained multiobjective optimization," *IEEE Trans. Syst. Man. Cybern.*, vol. 52, no. 6, pp. 3476–3491, 2021.

[55] Z. Liu, B. Wang, and K. Tang, "Handling constrained multiobjective optimization problems via bidirectional coevolution," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10 163–10 176, 2021.

[56] F. Caraffini, F. Neri, and L. Picinali, "An analysis on separability for memetic computing automatic design," *Inf. Sci.*, vol. 265, pp. 1–22, 2014.

[57] S. Datta, A. Ghosh, K. Sanyal, and S. Das, "A radial boundary intersection aided interior point method for multi-objective optimization," *Inf. Sci.*, vol. 377, pp. 1–16, 2017.

[58] V. A. S. Hernández, O. Schütze, H. Wang, A. Deutz, and M. Emmerich, "The set-based hypervolume newton method for bi-objective optimization," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 2186–2196, 2018.

[59] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. D. Goodman, "Push and pull search for solving constrained multi-objective optimization problems," *Swarm Evol. Comput.*, vol. 44, pp. 665–679, 2019.

[60] Y. Tian, Y. Zhang, Y. Su, X. Zhang, K. C. Tan, and Y. Jin, "Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9559–9572, 2021.

[61] P. Singh, I. Couckuyt, F. Ferranti, and T. Dhaene, "A constrained multi-objective surrogate-based optimization algorithm," in *CEC'14: Proc. of 2014 IEEE Congress on Evolutionary Computation*, 2014, pp. 3080–3087.

[62] R. Datta and R. G. Regis, "A surrogate-assisted evolution strategy for constrained multi-objective optimization," *Expert Syst.*, vol. 57, pp. 270–284, 2016.

[63] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching-based selection in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 6, pp. 909–923, 2013.

[64] C. Bragalli, C. D'Ambrosio, J. Lee, A. Lodi, P. Toth *et al.*, "Water network design by minlp," IBM, Tech. Rep., 2008.

[65] Q. Zhu, Q. Zhang, and Q. Lin, "A constrained multiobjective evolutionary algorithm with detect-and-escape strategy," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 938–947, 2020.

[66] Z. Liu and Y. Wang, "Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 870–884, 2019.

[67] F. Ming, W. Gong, D. Li, L. Wang, and L. Gao, "A competitive and cooperative swarm optimizer for constrained multi-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, 2022, accepted for publication.

[68] R. Storn and K. V. Price, "Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[69] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, 2003.

[70] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *EMO'15: Proc. of the 8th International Conference on Evolutionary Multi-Criterion Optimization*, vol. 9019, 2015, pp. 110–125.

[71] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, 1999.

[72] W. Frank, *Individual comparisons by ranking methods.* Springer, 1992.

[73] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011.

[74] A. Vargha and H. D. Delaney, "A critique and improvement of the cl common language effect size statistics of mcgraw and wong," *J. Educ. Behav. Stat.*, vol. 25, no. 2, pp. 101–132, 2000.