Complex & Intelligent Systems (2024) 10:7711–7740
https://doi.org/10.1007/s40747-024-01542-9

ORIGINAL ARTICLE

Adaptive multi-stage evolutionary search for constrained multi-objective optimization

Huiting Li[1] · Yaochu Jin[1,2,3] · Ran Cheng[4]

Received: 13 September 2023 / Accepted: 3 April 2024 / Published online: 31 July 2024
© The Author(s) 2024

Abstract

In this paper, we propose a multi-stage evolutionary framework with adaptive selection (MSEFAS) for efficiently handling constrained multi-objective optimization problems (CMOPs). MSEFAS has two stages of optimization in its early phase of evolutionary search: one stage that encourages promising infeasible solutions to approach the feasible region and increases diversity, and the other stage that enables the population to span large infeasible regions and accelerates convergence. To adaptively determine the execution order of these two stages in the early process, MSEFAS treats the optimization stage with higher validity of selected solutions as the first stage and the other as the second one. In addition, at the late phase of evolutionary search, MSEFAS introduces a third stage to efficiently handle the various characteristics of CMOPs by considering the relationship between the constrained Pareto fronts (CPF) and unconstrained Pareto fronts. We compare the proposed framework with eleven state-of-the-art constrained multi-objective evolutionary algorithms on 56 benchmark CMOPs. Our results demonstrate the effectiveness of the proposed framework in handling a wide range of CMOPs, showcasing its potential for solving complex optimization problems.

Keywords Multi-stage evolutionary framework with adaptive selection · Constrained multiobjective optimization problems · Evolutionary search · Infeasible solutions

Introduction

In real-world applications, a variety of optimization problems encompass multiple conflicting objectives and numerous equality and/or inequality constraints. Examples of such problems can be found in the design of tall buildings [1], optimization of robot grippers [2], development of composite structures [3], vehicle routing with time windows [4], and structural damper systems [5]. These challenges are collectively referred to as Constrained Multi-Objective Optimization Problems (CMOPs).

In general, a CMOP can be mathematically formulated as:

$$\begin{cases} \min imize & F(x) = (f_1(x), ..., f_M(x))^T \\ subject\ to & g_j(x) \le 0, \quad j = 1, ..., r \\ & h_i(x) = 0, \quad i = 1, ..., l \\ & x = (x_1, ..., x_V) \end{cases} \tag{1}$$

where $F(x) \in Y$ is composed of M objective functions, and Y is the objective space; $g_j(x)$ and $h_i(x)$ represent the jth inequality constraint and the ith equality constraint respectively; r and l represent the number of inequality and equality constraints respectively; $x \in \Omega$ is the solution consisting of V decision variables, and Ω is the decision variable space. The overall constraint violation of the solution x is defined as follows [6]:

✉ Yaochu Jin
jinyaochu@westlake.edu.cn

Huiting Li
3219415399@qq.com

Ran Cheng
ranchengcn@gmail.com

1 State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, China

2 School of Engineering, Westlake University, Hangzhou 310030, China

3 Department of Computer Science, University of Surrey, Guildford GU2 7XH, UK

4 Department of Computer Science and Engineering, Guangdong Provincial Key Laboratory of Brain-Inspired Intelligent Computation, Southern University of Science and Technology, Shenzhen, China

$$CV(x) = \sum_{j=1}^{r} \max\{0, g_j(x)\} + \sum_{i=1}^{l} |h_i(x)|. \quad (2)$$

If a solution meets all of the constraints, it is considered a feasible solution; otherwise, it is considered an infeasible solution. A feasible solution x_1 is said to dominate another feasible solution x_2 if $f_h(x_1) \leq f_h(x_2)$ for all $h \in 1, ..., M$ and $f_g(x_1) < f_g(x_2)$ for at least one $g \in 1, ..., M$. If no other feasible solution dominates it, a solution is known as a Pareto optimal solution. The set of all Pareto optimal solutions in the decision space is called the Pareto optimal set, and the image of these solutions in the objective space is known as the Pareto front (PF). When solving CMOPs, the goal is to approximate the PF with a set of well-converged and well-distributed feasible solutions [7].

CMOPs present significant challenges due to the constraints that can make a large portion of the search space infeasible, partitioning the feasible space into narrow, disconnected regions and potentially making the PF of unconstrained multi-objective optimization problems (MOPs) partially or completely infeasible [8]. These changes to the search space, caused by various constraints [9], make CMOPs generally more challenging to solve compared to unconstrained MOPs.

To address these challenges, an effective approach is to simultaneously balance the satisfaction of constraints and the optimization of the objective functions. Researchers have developed several techniques for achieving this balance, including the multi-ranking technique [10, 11], multi-stage technique [9, 12–17], and co-evolutionary technique [18–20]. Among these, multi-stage techniques have received increasing attention for balancing constraint satisfaction and objective optimization through the use of different criteria and strategies at different stages. However, current methods for solving CMOPs may not be efficient enough due to slow convergence or a lack of ability to explore feasible regions. Additionally, these methods may not be able to solve all CMOPs well due to the varying characteristics of different problems. Consequently, addressing CMOPs with varying characteristics effectively requires consideration of several key aspects. First, determining the optimal initial optimization population for different CMOPs during the early optimization stage is crucial in order to expedite the optimization process. Second, the adaptive selection of suitable optimization methods for various stages is essential, considering the evolving characteristics of the population of different CMOPs in the early stage. Lastly, adaptively allocating the appropriate number of execution stages, contingent upon the effectiveness of the employed optimization methods, is an important factor in devising a proficient approach to solving CMOPs.

To address the above challenges, an effective approach is to simultaneously balance the satisfaction of constraints and the optimization of the objective functions. Researchers have developed several techniques for achieving this balance, including the multi-ranking techniques [10, 11], multi-stage techniques [9, 12–17], and co-evolutionary techniques [18–20]. Among them, the challenge of co-evolutionary techniques is to balance convergence and diversity in solving CMOPs. Most co-evolutionary CMOEAs follow the principle that convergence takes precedence over diversity, such as C-NSGAII [21], but in fact both are equally important. Some algorithms, such as [22], fall into an easily searchable part of PF because of ignoring solutions with relatively poor convergence but good diversity performance, resulting in low effectiveness in solving CMOPs of disconnected feasible regions and discrete PFs. To achieve a balanced search in feasible and infeasible regions, CMOEAs such as C-TAEA [18], CCMODE [19] and CCMO [20] have been developed. Unfortunately, these algorithms ignore the information of infeasible solutions, resulting in poor optimization results on some CMOPs. To address the above limitations, the c-DPEA algorithm [23] is further developed. By introducing the information about infeasible solutions, c-DPEA is able to achieve good results in solving CMOPs with disconnected feasible regions and discrete PFs. However, the diversity of solutions is insufficient due to the existence of adaptive penalty term in the late optimization process. In addition, most existing multi-population based CMOEAs transfer information between two populations, because efficient information transfer can effectively improve the performance of both populations. Information transfer, however, is neglected in most CMOEAs.

The multi-stage techniques have also received increasing attention for balancing constraint satisfaction and objective optimization through the use of different criteria and strategies at different stages. However, current methods for solving CMOPs may not be efficient enough due to slow convergence or a lack of the ability to explore feasible regions. Additionally, these methods may not be able to solve all CMOPs well because of the varying characteristics of different problems. Consequently, addressing CMOPs with varying characteristics effectively requires consideration of several key aspects. For multi-stage optimization, the strategy for switching from one stage to another is also very important, because it directly affects the performance of CMOEAs. Researchers have proposed a number of switching strategies. For example, in PPS [24], the switch is performed when the maximum rate of change between the ideal point and the lowest point is less than a predefined value. In ToP [13], the switch is performed when multiple feasible solutions are obtained and some of them are of high quality. In CMOEA-MS [9], the switch is performed when the proportion of feasible solutions in the population is less than a predefined value. So the rational-

ity of the switching strategy has an important impact on the optimization performance.

Considering the above issues in most existing multi-population and multi-stage CMOEAs, we aim to design an effective framework that can balance the convergence, diversity and feasibility of CMOPs, and give corresponding algorithms to satisfy the preferences in the entire optimization process for CMOPs with different characteristics. It also requires high-quality information transfer between populations in different optimization processes to solve as many CMOPs as possible.

Based on the above discussion, this work proposes a multi-stage evolutionary framework with adaptive selection. An algorithm based on this framework, called MSEFAS, is developed, which provides corresponding algorithms for solving CMOPs with different characteristics in different optimization stages. MSEFAS can realize high-quality information exchange between populations in different optimization processes to solve CMOPs more efficiently. Different from the switching strategy of existing CMOEAs, this paper designs a new strategy to allocate the execution stage for each algorithm. The main contributions of this work are as follows:

- MSEFAS proposes a two-phase multi-stage evolutionary algorithm to address the challenges of CMOPs. The early phase consists of two stages: the first stage focuses on optimizing competitive infeasible solutions with favorable objective values and minimal constraint violations, facilitating a balance between diversity and feasibility. The second stage aims to enhance the convergence or diversity performance on CMOPs with limited feasible regions. In the later phase of the evolutionary process, a third stage is introduced to efficiently handle the diverse characteristics of CMOPs by examining the relationship between the constrained Pareto optimal front (CPF) and the unconstrained Pareto optimal front (UPF).
- MSEFAS adaptively assigns the execution order to each algorithm in the early phase, providing initial optimized populations based on their effectiveness in selecting individuals for CMOPs and MOPs during multiple optimizations. Notably, the c-DPEA algorithm [23] is employed in the optimization stage that introduces infeasible solutions.
- MSEFAS adaptively allocates execution stages to each algorithm, considering the cumulative effectiveness of individuals selected by both the infeasible solutions optimization algorithm and the offspring individual-based optimization algorithm for CMOPs and MOPs across multiple optimizations.

The remainder of this paper is structured as follows. In "Related work and motivation", we review related work in the field of CMOPs and motivation. In "Proposed MSEFAS framework", we present our proposed multi-stage evolutionary framework with adaptive selection (MSEFAS) for efficiently solving CMOPs with different characteristics. In "Experimental studies", we present the experimental results of MSEFAS on various CMOPs. Finally, in "Conclusion", we draw our conclusions and discuss future work.

Related work and motivation

This section briefly reviews existing relevant work and provides the motivation for this paper. Table 1 summarizes and evaluates relevant constrained multi-objective evolutionary algorithms (CMOEAs).

According to the techniques for effectively balancing the satisfaction of constraints and the optimization of objective functions, CMOEAs can be classified into three categories: multi-ranking techniques, multi-stage techniques, and co-evolutionary techniques.

The first category is the multi-ranking techniques. To overcome the disadvantage of constraint handling techniques (CHTs) that prefers constraints over objectives and leads to premature convergence, Ma et al. [10] proposed a new CHT based on constrained dominance principle (CDP). In CHT, the fitness function of each solution is defined as the weighted sum of both the CDP-based solution ranking and the Pareto-dominant-based solution ranking (ie, without considering any constraints). Simultaneously, the weights adopted in the fitness function are associated with the proportion of feasible solutions in the current population. Although the proposed ToR considers both convergence and feasibility, it neglects diversity. Moreover, since different problems have varying preferences for constraints and objectives, the constraint bias given by ToR is consistently greater than the objective bias, making it difficult to strike a balance between constraints and objectives.

The second category is multi-stage techniques. These methods involve dividing the optimization process into multiple stages to adaptively balance objective optimization and constraint satisfaction. Tian et al. [9] proposed a two-stage evolutionary algorithm that adjusts the fitness evaluation strategy during the evolution process to achieve this balance. The first stage disregards constraints to search the UPF, enabling the population to explore the feasible region across the infeasible region; the second stage searches for optimal solutions distributed at the boundary of the feasible region by considering constraints. However, the method only adjusts the search of these two stages based on the proportion of feasible solutions, which leads to a lower overall convergence rate. Liu et al. [13] introduced a two-phase framework called ToP, in which the first stage finds promising feasible regions by transforming the CMOP into a constrained

Table 1 Summary and evaluation of the relevant CMOEAs

Category	Algorithms	Advantages	Shortcomings
Multi-ranking techniques	ToR [10]	It considers both convergence and feasibility	The constraint bias is greater than the objective bias, which makes it difficult to strike a balance between constraints and objectives
Multi-stage techniques	CMOEA-MS [9]	The population can reach the feasible region through the infeasible region and spread along the feasible region boundaries	It only adjusts the search of two stages by the proportion of feasible solutions, which leads to a lower convergence rate
	ToP [13]	It introduces the constrained single-objective optimization method	It is difficult to find suitable conditions that can transform the first stage into the second stage
	MOEA/D-DAE [14]	It helps the population get rid of the local optimal situation	The local optimal value is easily misjudged as the global optimal value
	MSCMO [15]	CMOPs with complex constraints can be solved by adding constraints one by one	It leads to the existence of some infeasible solutions in the later stages of evolution
	PDTP-MDE [17]	It reduces the optimization difficulty	The setting of weights affects the quality of the solutions generated
	PPS [24]	It can pass through the infeasible region quickly	It is difficult to achieve the crossing of populations from UPF to CPF when the region of the decision space is small
Co-evolutionary techniques	CTAEA [19]	It can solve CMOPs with discontinuous or inner and outer layer feasible regions well	Strong association between the two populations can result in poor optimization effect on the CMOPs with complex constraints
	CCMODE [19]	It utilizes knowledge transfer among all single-objective subpopulations to promote the use of objective information by the archive population	It is time-consuming and a waste of computing resources
	CCMO [20]	It can achieve a balance of constraints and objectives through collaboration between two populations	Some promising infeasible solutions are not considered
	MTCMO [25]	It provides a complement to the evolutionary direction	It is ignored that different CMOPs have different preferences for objectives and constraints during evolution

single-objective optimization problem, and the second stage uses a specific CMOEA to find the final solutions. Although it employs the constrained single-objective optimization method, the problem's difficulty is not reduced due to the lack of simplification of constraints. Additionally, it is challenging to find suitable conditions for transitioning from the first stage to the second stage. Zhu et al. [14] proposed the MOEA/D-DAE algorithm, which includes a detect-and-escape strategy to guide the search out of stagnant states caused by constraints. Although this method can help the population escape local optima, when the infeasible region of the constraint problem is complex, local optima can be easily misjudged as global optima. Ma et al. [15] proposed a multi-stage evolutionary algorithm that incrementally adds constraints at different stages of evolution to solve CMOPs with complex constraints. However, this approach results in the presence of some infeasible solutions in the later stages of evolution, reducing the optimization effectiveness. Yu et al. [16] proposed a dynamic selection preference assisted constrained multi-objective differential evolution (DE) algorithm, which switches the selection preference between the objective function and constraints during the evolution process. Yu et al. [17] introduced the purpose-directed two-phase multiobjective differential evolution (PDTP-MDE) algorithm, which divides the evolutionary process into two stages based on the intended purpose of each stage. Although the method reduces the optimization difficulty by decomposing the optimization problem into simpler subproblems, the decomposition relies on human experience, and the setting of weights also affects the quality of solutions generated based on the decomposition method. Ming et al. [26] proposed a dual-stage dual-population evolutionary algorithm that efficiently explores and exploits both infeasible and feasible solutions. Fan et al. [24] proposed the push-pull search (PPS) algorithm, which divides the search process into a push search phase and a pull search phase. Although this method can quickly traverse the infeasible region, it struggles to achieve the crossing of populations from UPF to CPF when the decision space region is small.

The third category is co-evolutionary techniques. Li et al. [18] proposed a parameter-free technique called the two-archive evolutionary algorithm. This method involves maintaining two archives: a convergence-oriented archive (CA) and a diversity-oriented archive (DA). The purpose of the two-archive technique is to explore under-exploited areas of the CA through the DA in order to complement the behavior of the CA and provide as much diversity information as possible. To achieve this, the method utilizes a constrained mating selection mechanism which adaptively selects suitable mating parents based on the evolutionary state. This mechanism takes advantage of the complementary effects of the two archives. Although utilizing two archives can effectively solve CMOPs with discontinuous or inner and outer layer feasible regions, the strong association between the two populations can negatively impact the evolution of their populations, resulting in suboptimal optimization performance on CMOPs with complex constraints. Wang et al. [27] proposed a decomposition-based multi-objective optimization method for solving constrained optimization problems (COPs). This method involves transforming the COP into a biobjective optimization problem (BOP) and then optimizing the transformed BOP in a decomposition-based framework. The method also employs a restart strategy to address complex constraints. Wang et al. [19] introduced the cooperative differential evolution framework (CCMODE), which involves M constrained single-objective subpopulations and an archived population that cooperate to optimize all objectives and constraints of CMOPs. Although this approach utilizes knowledge transfer among all single-objective subpopulations to promote the use of objective information by the archive population, it is time-consuming and resource-intensive due to the simultaneous optimization of the $M + 1$ populations. Tian et al. [20] proposed a co-evolutionary framework for constrained multi-objective optimization that involves sharing information between two populations to solve CMOPs. Although the approach can achieve a balance between constraints and objectives through collaboration between two populations, some promising infeasible solutions are not considered. Wang et al. [28] introduced the cooperative multi-objective evolutionary algorithm with propulsive population (CMOEA-PP), which uses normal and propulsive populations to work together to solve CMOPs. Sun et al. [29] proposed the evolutionary algorithm with constraint relaxation strategy based on the differential evolution algorithm (CRS-DE), which divides the population into three subpopulations according to the overall constraint violation in order to relax constraints. Qiao et al. [25] introduced the multi-task constrained multi-objective optimization (MTCMO) framework, which involves the use of an auxiliary task with a dynamically decreasing constraint bound to assist in solving the main task. While the method makes the auxiliary problem highly correlated with the main problem by dynamically reducing the constraint boundary and provides a complement to the evolutionary direction, it adopts a fixed constraint boundary reduction method. This approach overlooks the fact that different CMOPs have distinct preferences for objectives and constraints during evolution.

In addition to the three techniques mentioned above, there have been some recent studies that have explored alternative approaches to solving CMOPs. Yuan et al. [30] introduced a cost value-based distance into the target space and used this distance, along with constraints, to define a metric for evaluating each individual's contribution to exploring promising areas. This led to the development of a metric-based constrained multi-objective algorithm. Xiang et al. [31] proposed

a new CHT (tradeoff model) that takes into account the underlying problem types, as determined by the relationship between the CPF and the UPF. Wu et al. [32] introduced the voting mechanism-based ensemble constraint handling framework (VMCH), which involves the use of multiple CHTs as voters to compare and select solutions. Peng et al. [33] proposed a constraint processing technique based on directed weights for CMOPs, in which infeasible individuals with good objective values and small constraint violations are retained in the population to improve the performance of the algorithm.

Although numerous CMOEAs have been employed to solve CMOPs with some success, most of them face difficulties when addressing CMOPs with high constraint complexity. This is primarily due to constraints causing the feasible region to become discrete, narrow, or challenging to traverse through the infeasible region. Multi-stage based CMOEAs are effective algorithms for handling complex CMOPs. Their principle involves dividing the process of solving the optimal feasible PF into multiple tasks, with each task processed at different stages. However, multi-stage based CMOEAs generally cannot achieve satisfactory results on all CMOPs and exhibit some limitations. The main reason is that it is impossible to accurately determine when to switch from one stage to another for CMOPs with varying characteristics, leading to insufficient efficiency and universality.

Existing multi-stage based CMOEAs mostly adopt a two-stage framework. For instance, the PPS approach [24] targets the unconstrained PF in the push stage and considers constraints in the pull stage, without considering solution feasibility in the early optimization stage. However, the convergence and the performance of searching feasible regions of PPS on solving CMOPs with discontinuous or disjoint feasible regions are not satisfied. In addition, the use of the MOEA/D framework in PPS results in a serious performance degradation when solving CMOPs with an irregular PF. On the other hand, ToP [13] considers all constraints initially, introducing a single objective before considering all constraints and objectives later. However, because ToP uses a reference vector to transform a CPF into a constrained single-objective optimization problem, the convergence rate is slow, giving rise to difficulties for the algorithm to converge to the CPF. Moreover, the solutions found in the first stage are relatively clustered, which affects the performance of optimization in the later stage. Similarly, CMOEA-MS [9] prioritizes objectives over constraints in the early stage, focuses on achieving a good solution diversity in the later stage. However, the convergence effect is not satisfactory when solving the CMOPs with disconnected CPF. MSCMO [15] initially considers fewer constraints, allowing populations to converge to promising feasible regions with a good degree of diversity, and searches for optimal solutions by considering additional constraints later. However, when the number of evaluations is limited, the solutions explored by the algorithm cannot satisfy all constraints, and there may be some infeasible solutions.

This work distinguishes itself with the existing work in three significant ways. Firstly, unlike traditional CMOEAs that do not account for the preference differences of CMOPs for objective optimization and constraint satisfaction at the onset of optimization, this study incorporates these preferences at the beginning of the search process. This is achieved by assessing the preferences of different CMOPs for objective optimization and constraint satisfaction, and subsequently adaptively assigning suitable algorithms to different stages based on their unique search characteristics. Secondly, this study diverges from previous algorithms that determine the number of executions at each optimization stage based on either user-specified conditions or the ratio of feasible solutions. Instead, the proposed algorithm facilitates an adaptive transition from one stage to another by assigning corresponding execution stages predicated on the effectiveness of solutions selected by each algorithm. Lastly, this research departs from the convention in existing CMOEAs of randomly selecting initial populations. Rather, it optimizes the initial populations during the early phase, thereby accelerating the initiation of the optimization process.

Building upon the aforementioned motivations, we propose an adaptive multi-stage evolutionary search for constrained multi-objective optimization to satisfy different CMOPs having different preferences for feasibility, diversity, and convergence. The algorithm comprises two phases. Initially, we must assess the preferences of CMOPs with different characteristics for feasibility, diversity, and convergence in the early stage. Consequently, two stages are introduced in the early phase. One stage is utilized to optimize competitive infeasible solutions with good objective values and low constraint violations, enabling a transition between diversity and feasibility. The other stage is designed to enhance the convergence or diversity performance of CMOPs with small feasible regions. The initial optimized populations and corresponding execution order are determined based on the effectiveness of the individuals selected for CMOPs and MOPs in these two stages. Additionally, to make all stages cooperate better with each other, the execution stages are adaptively assigned to each algorithm based on the accumulated effectiveness of the individuals selected in multiple optimizations. In the later phase, a third stage is introduced to ensure that the algorithm can solve as many CMOPs as possible, mainly because the relationship between the CPF and the UPF can enable high-quality information transfer between populations.

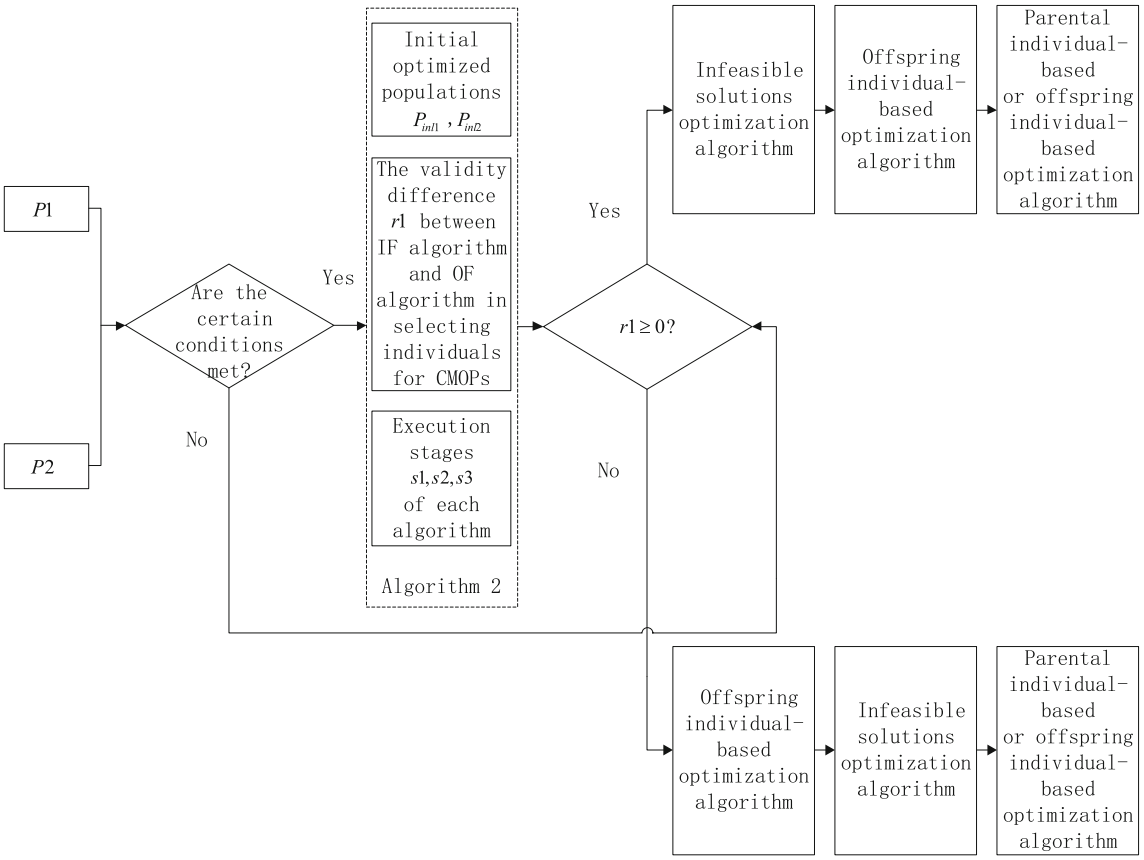


Fig. 1 The overall framework of the proposed algorithm, where P_1 and P_2 are used to solve the original CMOPs and the unconstrained MOPs, respectively, $r1$ is the validity difference between the IF and the OF algorithms. $s2$, $s2$ and $s3$ are three stages defined in Algorithm 2.

Proposed MSEFAS framework

The overall framework of MSEFAS is shown in Algorithm 1. MSEFAS first initializes two populations, P_1 and P_2, where P_1 is used to solve the original CMOPs and P_2 is used to solve unconstrained MOPs.

Since different problems exhibit varying characteristics, it is essential to determine at the beginning of the optimization process whether introducing infeasible solutions to increase diversity is necessary, how to select the initial optimized populations to enhance the optimization effect, and how to establish the execution stage of each algorithm in the optimization. To answer these three questions, we first need to evaluate the introducing infeasible solutions optimization algorithm (IF algorithm) and the offspring individual-based optimization algorithm (OF algorithm) separately to determine which of the two algorithms is more effective in selecting solutions. If the effectiveness of these two algorithms is determined by only one iteration at the beginning of the optimization, it may result in inaccurate judgments. Conversely, too many iterations can not only increase computational costs but also reduce the effectiveness of the optimization. In this work, we use the satisfaction of the number of iterations T_c being less than a given value as the determining condition. If this condition is satisfied, initial optimized populations, P_{inl1} and P_{inl2}, along with the parameter $r1$ of the validity difference between the IF algorithm and the OF algorithm in selecting individuals for CMOPs, and the execution stages $s1$, $s2$, and $s3$ of each algorithm are defined using Algorithm 2. The early phase contains two stages, IF algorithm and OF algorithm. Based on the value of $r1$, it is determined whether the IF algorithm or OF algorithm should be used as the first stage of optimization in early phase for the CMOPs. The second stage consists of the algorithm that was not selected for the first stage. In order to efficiently deal with CMOPs with different characteristics based on the relationship between CPF and UPF, we apply the parental individual-based or offspring individual-based optimization algorithm (PO algorithm) in the late phase. The optimization process is completed by combining the order of operation of the algorithms and the execution stages of each algorithm as calculated using Algorithm 2. If the number of iterations T_c no longer meets a predefined condition, Algorithm 2 is no longer executed and the process is repeated based on the value of $r1$ until the termination condition is reached. The

overall framework of the proposed algorithm is shown in Fig. 1.

Algorithm 2 has aims to generate the initial optimized populations and provide the execution stage of each algorithm.

To accomplish these goals, the algorithm first uses the IF and OF algorithms to solve the CMOPs and MOPs, respectively, and obtains the corresponding selected individuals P_1^{IF}, P_2^{IF}, P_1^{OF}, and P_2^{OF}. It then calculates the effectiveness parameters $a_{P_1^{IF}}$, $a_{P_2^{IF}}$, $a_{P_1^{OF}}$, $a_{P_2^{OF}}$ for these individuals using Eq. (3). These parameters represent the ratio of the number of individuals selected by each algorithm to the best N individuals selected for the CMOPs and MOPs.

Algorithm 1 The overall framework of MSEFAS

Input: N (population size), P_1 (the population for CMOPs), P_2 (the population for MOPs), $MaxFEs$ (maximum function evaluations).
Output: P_1.

$FEs = 0$;
Initialize P_1 and P_2 of size N;
Evaluate P_1 on the CMOP;
Evaluate P_2 on the MOP;
while termination criterion not satisfied **do**
  if $FEs/MaxFEs$= the given value **then**
    $P_{inl1}, P_{inl2}, r1, s1, s2, s3 \leftarrow$ **Algorithm2**;
  end if
  if $r1 \geq 0$ **then**
    if $FEs \leq s1$ **then**
      $P_1^{IF}, P_2^{IF} \leftarrow$ **IF Algorithm** ;
    else if $FEs > s1$ and $FEs \leq s1 + s2$ **then**
      $P_1^{OF}, P_2^{OF} \leftarrow$ **OF Algorithm**;
    else $FEs > s1 + s2$
      $P_1^{PO}, P_2^{PO} \leftarrow$ **PO Algorithm**;
    end if
  end if
  if $r1 < 0$ **then**
    if $FEs \leq s2$ **then**
      $P_1^{OF}, P_2^{OF} \leftarrow$ **OF Algorithm**;
    else if $FEs > s2$ and $FEs \leq s1 + s2$ **then**
      $P_1^{IF}, P_2^{IF} \leftarrow$ **IF Algorithm**;
    else $FEs > s1 + s2$
      $P_1^{PO}, P_2^{PO} \leftarrow$ **PO Algorithm**;
    end if
  end if
end while

$$a_{P_i^k} = numP_i^k/N(i = 1, 2, k = IF, OF) \tag{3}$$

$$ri = a_{P_i^{IF}} - a_{P_i^{OF}}(i = 1, 2) \tag{4}$$

Algorithm 2 Initial optimized populations and execution stage of each algorithm

Input: N, P_1, P_2, $MaxFEs$, **IF Algorithm**, **OF Algorithm**.
Output: P_{inl1}, P_{inl2}, $r1$, $s1$, $s2$, $s3$.

$P_1^{IF}, P_2^{IF} \leftarrow$ **IF Algorithm**;
$P_1^{OF}, P_2^{OF} \leftarrow$ **OF Algorithm**;
Calculate the efficiency of selecting individuals using two algorithms: $a_{P_1^{IF}}, a_{P_2^{IF}}, a_{P_1^{OF}}, a_{P_2^{OF}} \leftarrow Eq.(3)$;
Calculate $r1$ and $r2$ based on Eq.(4);
$A1 = 0, B1 = 0$;
if $r1 \geq 0$ and $r2 \geq 0$
  $P_{inl1} = P_1^{IF}$;
  $P_{inl2} = P_2^{IF}$;
else if $r1 \geq 0$ and $r2 < 0$
  $P_{inl1} = P_1^{IF}$;
  $P_{inl2} = P_2^{OF}$;
else if $r1 < 0$ and $r2 \geq 0$
  $P_{inl1} = P_1^{OF}$;
  $P_{inl2} = P_2^{IF}$;
else $r1 < 0$ and $r2 < 0$
  $P_{inl1} = P_1^{OF}$;
  $P_{inl2} = P_2^{OF}$;
end if
$A1 = a_{P_1^{IF}} + A1$;
$B1 = a_{P_1^{OF}} + B1$;
if $A1 - B1 \geq 0$ or $a_{P_1^{IF}} - a_{P_1^{OF}} \geq 0.2$ or $(a_{P_2^{OF}} - a_{P_2^{IF}} \geq 0.2$ and $B1 - A1 < 0.1)$
  $s1 = \left\| a_{P_1^{IF}} + a_{P_2^{OF}} \right\| * MaxFEs/4$;
  $s2 = \left\| a_{P_2^{OF}} - a_{P_1^{OF}} \right\| * MaxFEs/2 + 0.15 * MaxFEs$;
else if $B1 - A1 \geq 0.1$ or $a_{P_1^{OF}} - a_{P_1^{IF}} \geq 0.2$
  $s1 = \left\| a_{P_1^{IF}} - a_{P_2^{IF}} \right\| * MaxFEs/2 + 0.15 * MaxFEs$;
  $s2 = \left\| a_{P_1^{OF}} + a_{P_2^{OF}} \right\| * MaxFEs/4$;
else
  $s1 = a_{P_1^{IF}} * MaxFEs/2$;
  $s2 = a_{P_1^{OF}} * MaxFEs/2$;
end if
$s3 = MaxFEs - (s1 + s2)$;

Next, the algorithm uses Eq. (4) to calculate the difference $r1$ and $r2$ of the validity of the individuals selected by the two algorithms for the CMOPs and MOPs, respectively. Based on $r1$ and $r2$, it determines the initial population for the subsequent optimization process. The principle it follows is to select the population with the highest effectiveness, as determined by which algorithm selected the individuals.

The algorithm then gives the execution stage of each algorithm based on the effectiveness $a_{P_1^{IF}}, a_{P_2^{IF}}, a_{P_1^{OF}}, a_{P_2^{OF}}$ of

the individuals selected by the two algorithms. It accumulates the effectiveness of the two algorithms for the CMOPs in each cycle, denoted as $A1$ and $B1$. When one of three cases is satisfied, the algorithm gives the execution stage for each algorithm: (1) for the CMOPs, the cumulative validity of individuals obtained by the IF algorithm is higher than that of the OF algorithm in multiple evaluations; (2) in one evaluation, the validity of individuals obtained by the IF algorithm is 0.2 higher than that of the OF algorithm for the CMOPs; or (3) for the MOPs, the validity of individuals obtained by the OF algorithm in one evaluation is 0.2 higher than that of the IF algorithm.

In the $s3$ stage, we apply the PO algorithm as described in Algorithm 3 (with main reference [34]). To improve convergence, we use the binary tournament selection method to select mating parents from P_i based on their quality, and generate $N/2$ offspring individuals OP_i using evolutionary operators. These offspring are combined with the parents to form COP_i, which is then evaluated on the corresponding problem (lines 1–7).

Algorithm 3 PO algorithm

Input: N, P_1, P_2.
Output: P_1, P_2
1: **for** $i = 1 : 2$
2: $MatingParent_i \leftarrow$ Select $N/2$ individuals from P_i based on their quality using binary tournament selection method;
3: $OP_i \leftarrow$ Generate $N/2$ offspring based on the evolutionary operators by the $MatingParent_i$;
4: $COP_i \leftarrow P_i \cup OP_i$;
5: **end for**
6: Evaluate COP_1 on the MOP;
7: Evaluate COP_2 on the CMOP;
8: Calculate d_i according to Eq.(5);
9: **for** $i = 1 : 2$
10: **if** $d_i \geq 0$
11: //****PA Algorithm****//
12: $CP_i \leftarrow P_i \cup OP_i \cup P_{2/i}$;
13: **else if**
14: //****OF Algorithm****//
15: $CP_i \leftarrow P_i \cup OP_i \cup OP_{2/i}$;
16: **end if**
17: **end for**
18: Evaluate CP_1 on the CMOP;
19: Evaluate CP_2 on the MOP;
20: $P_1 \leftarrow$ Select N individuals from CP_1 using the environmental selection
21: $P_2 \leftarrow$ Select N individuals from CP_2 using the environmental selection;

Next, we calculate the difference between the success rates of the parent and offspring individuals using Eq. (5). If $d_i \geq 0$, it indicates that the parent individuals make up a higher percentage of the optimal N individuals compared to the offspring. In this case, we select the parents for optimization, merge P_i, OP_i and the selected $P_{2/i}$ to obtain CP_i, and refer to this as the PA Algorithm. On the other hand, if $d_i < 0$, it indicates that the offspring make up a higher percentage of the optimal N individuals. In this case, we select the offspring for optimization, merge P_i, OP_i and the selected $OP_{2/i}$ to obtain CP_i, and refer to this as the OF Algorithm. Both versions of CP_i are then evaluated on the corresponding problem separately (lines 8–19). Finally, we select N individuals from CP_i according to the environmental selection to form the new P_i (lines 20–21). The process of PO algorithm is shown in Fig. 2.

$$d_i = numP_{2/i}/N - numOP_{2/i}/(N/2)(i = 1, 2) \tag{5}$$

The details of the OF algorithm are described in Algorithm 4 (with main reference [34]). First, we randomly select the mating parents from P_i and then use the mating parents to generate offspring individuals OP_i of size $N/2$ by the evolutionary operators. P_i, OP_i, and $OP_{2/i}$ are combined to obtain the COP_i. After that, COP_i is evaluated on the corresponding problem. Finally, N individuals are selected from COP_i to form a new P_i using environmental selection.

Regarding the IF algorithm, in this paper, we embed the c-DPEA algorithm and Ship algorithm [35] into the IF algorithm, respectively, and discuss them in the later experiments.

Algorithm 4 OF algorithm

Input: N, P_1, P_2.
Output: P_1, P_2
1: **for** $i = 1 : 2$
2: $MatingParent_i \leftarrow$ Select $N/2$ individuals from P_i randomly;
3: $OP_i \leftarrow$ Generate $N/2$ offspring based on the evolutionary operators by the $MatingParent_i$;
4: $COP_i \leftarrow P_i \cup OP_i \cup OP_{2/i}$;
5: **end for**
6: Evaluate COP_1 on the CMOP;
7: Evaluate COP_2 on the MOP;
8: $P_1 \leftarrow$ Select N individuals from COP_1 using the environmental selection;
9: $P_2 \leftarrow$ Select N individuals from COP_2 using the environmental selection;

Experimental studies

In this section, we perform comprehensive experimental studies to assess the performance of the proposed MSEFAS. First, we compare the proposed MSEFAS with several state-of-the-art algorithms via benchmark tests: MOEADDAE [14], CMOEA-MS [9], CCMO [20], c-DPEA [23], EMCMO [34], TiGE2 [11], CTAEA [18], PPS [24], ToP [13], DSPCMDE [16], and MSCMO [15] on several test suites: constrained DTLZ [36], DOC [13], MW [8], LIRCMOP [37], and DASCMOP [38]. Then we conduct a series

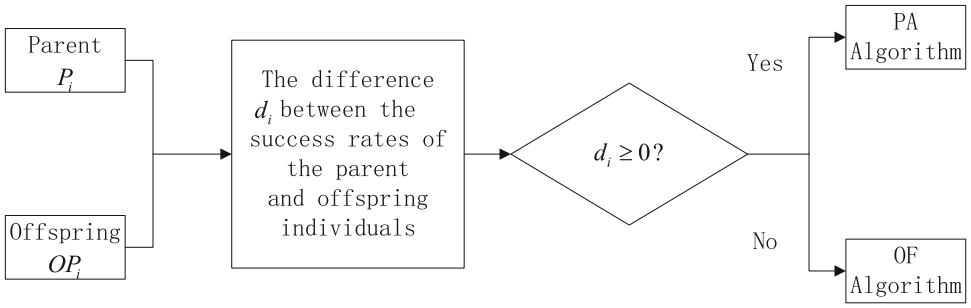


Fig. 2 The process of PO algorithm

of experiments to validate the effectiveness of the key functionalities of MSEFAS, including the adaptive assignment of execution stage, the adaptive selection of the initial optimized population, the algorithm selection in the initial stage, as well as the flexibility of the entire framework. Finally, we test the proposed MSEFAS on nine real-world CMOPs.

Performance Indicators

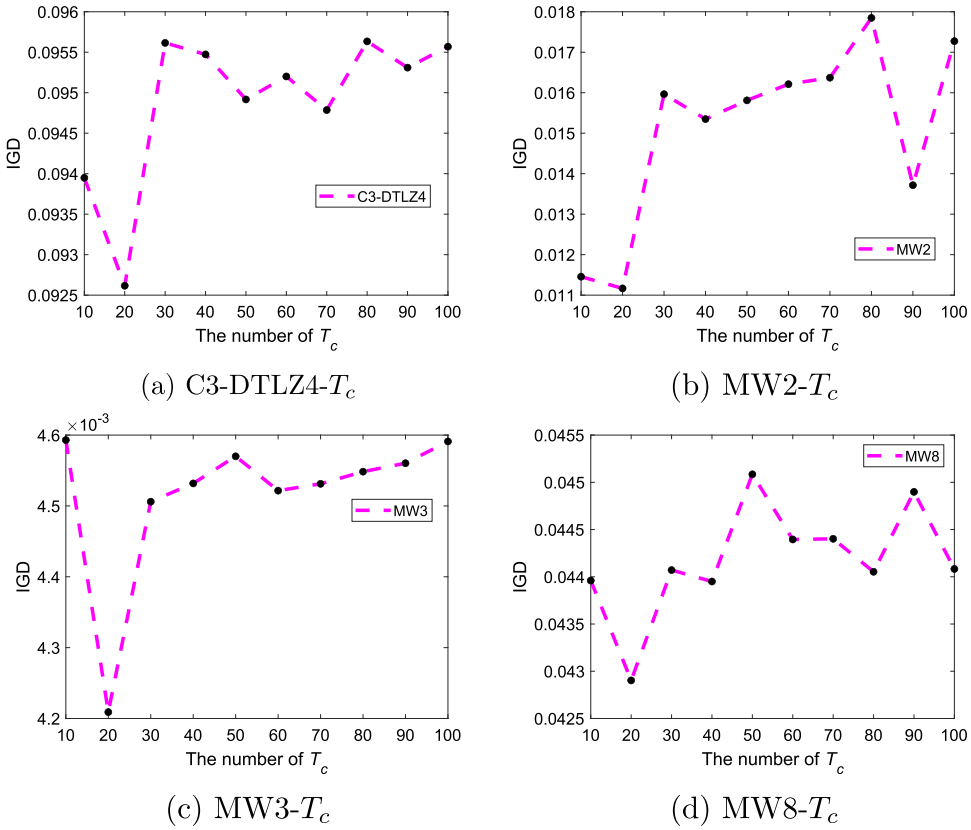
To evaluate the quality of the obtained solution sets, we use two widely-recognized performance indicators that consider both convergence and diversity: hypervolume (HV) [39] and inverted generational distance (IGD) [40]. For HV calculation, we use the maximum and minimum target values of the non-dominated solutions found by all compared algorithms as the reference point, with $\delta = 0.01$ (as described in [41]). For IGD calculation, we use 10000 evenly distributed Pareto optimal solutions sampled from the theoretical PF as the reference points.

To test for statistical significance, we employ the Wilcoxon rank sum test with a significance level of 5%. In the comparison results, "+" indicates that the proposed method is statistically better than the compared algorithm, "−" indicates that it is worse, and "=" means it is similar.

Parameter settings

- *Problems* : The number of decision variables V and the number of objectives M for each benchmark problem are set as follows. For the constrained DTLZ problems, C1-DTLZ1, DC1-DTLZ1, DC2-DTLIZ1, and DC3-DTLZ1, $M = 3$, $V = 7$, and for the remaining 6 problems, $M = 3$, $V = 12$. For DOC problems, DOC8 and DOC9, $M = 3$, for the remaining 7 problems, $M = 2$, the settings of V for different problems can refer to [13]. For the MW problems, V is set to 15 for all problems, $M = 3$ for MW4, MW8, and MW14, and $M = 2$ for the remaining 11 problems. For the LIR-CMOP problem, V is set to 10 for all problems, $M = 3$ for LIR-CMOP13 and LIRCMOP14, and $M = 2$ for the remaining 12 problems. For the DASCMOP problems, V is set to 10 for all problems, $M = 3$ for DAS-CMOP7, DAS-CMOP8 and DAS-CMOP9, and $M = 2$ for the remaining six problems.
- *Algorithms* : MOEADDAE adopts a detection and escape strategy to guide the search out of a stagnant state, and embeds this strategy into the decomposition-based multi-objective evolutionary framework. For MOEADDAE, the parameter α is set to 0.95. CMOEA-MS is a two-stage evolutionary algorithm that adjusts the fitness evaluation strategy during evolution to adaptively balance objective optimization and constraint satisfaction, the parameter for determining the current stage $\lambda = 0.5$. CCMO is a dual-population-based CMOEA embedded with a constrained NSGA-II, which adopts the truncation strategy in SPEA2 [42] in environmental selection and utilizes the constrained dominance principle method to solve constraints. c-DPEA is a cooperative co-evolutionary algorithm in which two populations cooperate by exchanging information during offspring generation. EMCMO adopts an evolutionary multitasking-based constrained multi-objective optimization framework to solve CMOPs by transforming the optimization of the CMOPs into two related tasks. For EMCMO, the parameter $\beta = 0.2$. The parameter settings of these five comparison algorithms are the same as those in the original literature. For the proposed MSEFAS, the c-DPEA algorithm is adopted as the IF (Introducing Infeasible Solutions) algorithm, and its parameter settings are the same as those used in the c-DPEA algorithm. The certain condition is whether the number of iterations T_c is less than 16. The analysis of parameter T_c is given in "Analysis of parameter settings". Detailed algorithm parameter settings can be found in Table SI of the Supplementary Material.

In addition to PPS, ToP and DSPCMDE adopt the differential evolution (DE) [43] operator and polynomial mutation (PM) [44]to generate offspring solutions, other algorithms all in this study use genetic algorithms with simulated binary crossover (SBX) [45] and PM as evolutionary operators to generate offspring solutions. The SBX operator has a crossover probability of $p_c = 1$ and a distribution index of $\eta_c = 20$. The PM operator has a mutation probability of



(a) C3-DTLZ4-T_c (b) MW2-T_c

(c) MW3-T_c (d) MW8-T_c

Fig. 3 Change in IGD values on C3-DTLZ4T, MW2, MW3 and MW8 test problems when T_c is set to 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100.

$p_m = 1/V$ and a distribution index of $\eta_m = 20$. The parameters crossover rate CR and scale factor F in the DE operator are set to 1.0 and 0.5, respectively. To ensure a fair comparison, we set the initial population size to 100 and the maximum number of function evaluations ($MaxFEs$) to 200,000 for all algorithms.

All experiments in this paper are implemented on PlatEMO [46] using MATLAB R2021a on an Intel Corei7 with 3.6 GHz CPU, running on Microsoft Window 10 Enterprize SP1 64-bit operating system.

Analysis of parameter settings

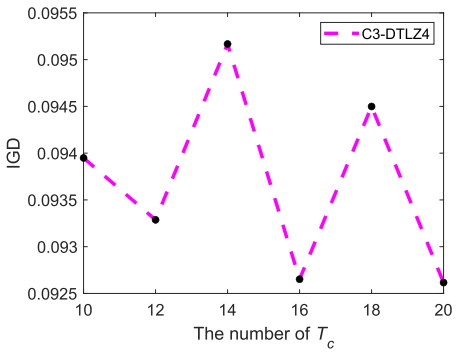
To test the influence of parameter T_c in the proposed MSEFAS on the experimental results, an experiment for sensitivity analysis is performed on the number of iterations T_c. We perform experiments for $T_c = 10, 20, 30, 40, 50, 60, 70, 80, 90$ and 100 iterations to solve C3-DTLZ4T, MW2, MW3 and MW8. Their IGD values are shown in Fig. 3. These results show that the algorithm works best when T_c is in the range of 10–20.

To explore the optimal setting of T_c, we divided T_c between 10 and 20 more finely. Figure 4 shows the IGD values of the MSEFAS for $T_c = 10, 12, 14, 16, 18$ and 20 over 20 independent runs on C3-DTLZ4T, MW2, MW3 and MW8, respectively, from which it can be seen that the optimization performance is the best when $T_c = 16$. Generally, the optimal T_c value is problem-dependent. For fair comparisons, however, we set T_c to 16 for all test problems.

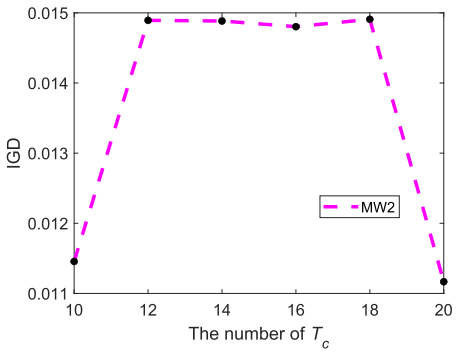
To investigate the influence of the four parameters p_c, η_c, p_m and η_m in the embedded c-DPEA algorithm on the optimization effect of MSEFAS, Fig. 5 shows the optimization effect of c-DPEA and MSEFAS algorithms on the MW2 when the four parameters change. As can be seen in Fig. 5, these four parameters have different degrees of influence on the optimization effect of both algorithms, but the MSEFAS consistently outperforms the c-DPEA.

Benchmark comparisons

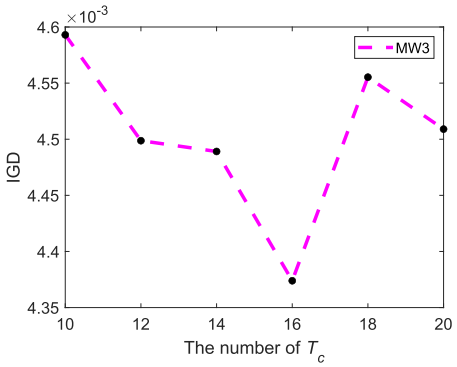
Tables 2 and 3 show the mean and standard deviation of IGD for MSEFAS and the comparison algorithms on five benchmark test suites, based on 30 independent runs. Note that NAN in all Tables indicates that the algorithm does not find a feasible solution. From the results in Table 2, we can see that MSEFAS obtains the best results on 25 out of 56 problems. It outperforms MOEADDAE, CMOEA-MS, CCMO, c-DPEA, and EMCMO on 47, 46, 40, 41, and 36 problems, respectively, and is comparable to these algorithms on 4, 9, 16, 13, and 20 problems, respectively. In addition to the mean values of MSEFAS presented, we also provide the best values of MSEFAS, namely MSEFAS(best). The mean and stan-



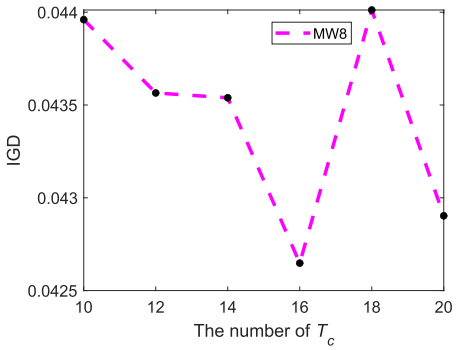
(a) C3-DTLZ4-T_c



(b) MW2-T_c

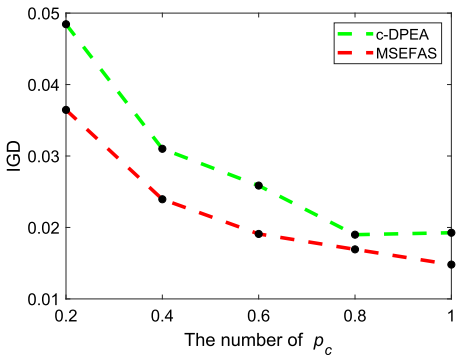


(c) MW3-T_c

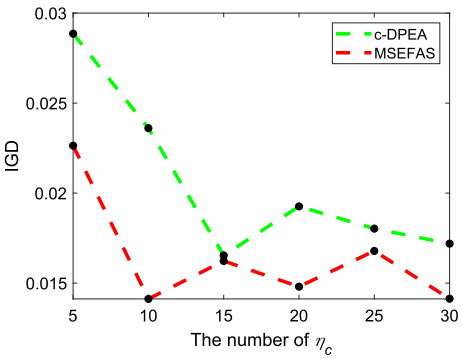


(d) MW8-T_c

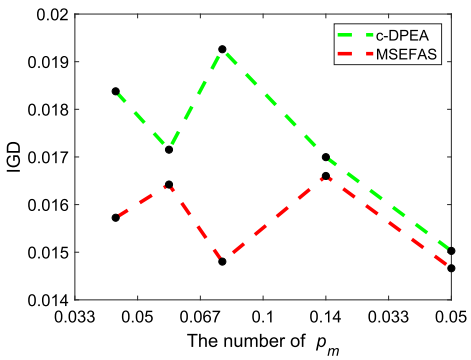
Fig. 4 Change in IGD values on C3-DTLZ4T, MW2, MW3 and MW8 test problems when T_c is set to 10, 12, 14, 16, 18 and 20



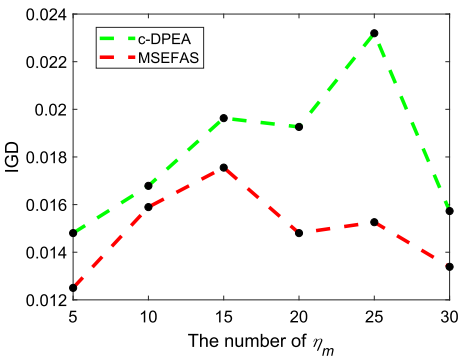
(a) MW2-p_c



(b) MW2-η_c



(c) MW3-p_m



(d) MW2-η_m

Fig. 5 The performance of c-DPEA and MSEFAS on MW2 when the four parameters in simulated binary crossover and polynomial mutation change

Table 2 Mean (standard deviation) of IGD obtained by MSEFAS and five compared algorithms on five benchmark test suites for 30 independent runs

Problem	MOEADDAE	CMOEA-MS	CCMO	c-DPEA	EMCMO	MSEFAS	MSEFAS (best)
C1-DTLZ1	2.1068e−2 (1.32e−4) −	2.0687e−2 (2.43e−4) −	2.0072e−2 (1.30e−4) −	2.0046e−2 (1.49e−4) −	1.9750e−2 (8.89e−5)−	**1.9677e–2 (6.04e–5)**	**1.9564e–2(3.89e–5)**
C1-DTLZ3	6.5273e−2 (4.80e−3) −	5.3225e−2 (5.70e−4) −	5.3523e−2 (5.98e−4) −	5.3563e−2 (6.81e−4) −	5.2894e−2 (5.95e−4) −	**5.2589e–2 (4.31e–4)**	**5.1634e–2 (2.63e–4)**
C2-DTLZ2	4.4195e−2 (4.58e−4) −	4.2392e−2 (4.41e−4) =	4.2874e−2 (4.36e−4) −	4.2809e−2 (6.32e−4) −	4.2360e−2 (4.80e−4) =	4.2352e−2 (3.85e−4)	4.1738e−2 (9.82e−6)
C3-DTLZ4	9.4998e−1 (3.62e−1) −	6.5057e−1 (5.11e−2) −	9.5283e−2 (1.25e−3) −	9.7113e−2 (1.37e−3) −	1.1985e−1 (1.36e−1) −	**9.2652e–2 (8.49e–4)**	**9.1366e–2 (2.32e–4)**
DC1-DTLZ1	1.2098e−2 (9.86e−5) −	1.8218e−2 (6.14e−3) −	1.1373e−2 (8.36e−5) =	1.1386e−2 (6.60e−5) =	1.1388e−2 (5.72e−5) =	1.1364e−2 (7.96e−5)	1.1220e−2(6.84e−5)
DC1-DTLZ3	3.7260e−2 (1.37e−3) −	3.3757e−2 (4.52e−4) =	3.3728e−2 (2.56e−4) =	3.3889e−2 (3.53e−4) −	3.3723e−2 (2.64e−4) =	3.3680e−2 (2.97e−4)	3.3094e−2(2.11e−4)
DC2-DTLZ1	2.1203e−2 (1.27e−4) −	2.0667e−2 (2.02e−4) −	2.0163e−2 (1.50e−4) −	2.0129e−2 (1.77e−4) −	1.9805e−2 (9.46e−5) =	1.9768e−2 (6.50e−5)	1.9640e−2(1.50e−5)
DC2-DTLZ3	3.7691e−1 (2.48e−1) −	1.7132e−1 (2.18e−1) −	5.3043e−2 (4.52e−4) −	5.2854e−2 (4.11e−4) −	5.2255e−2 (3.55e−4) −	**5.1702e–2 (1.58e–4)**	**5.1335e–2 (3.24e–4)**
DC3-DTLZ1	8.7975e−3 (7.98e−3) −	3.3577e−2 (6.05e−3) −	6.8156e−3 (5.05e−5) −	6.8487e−3 (5.32e−5) −	6.8166e−3 (5.43e−5) −	**6.7729e−3 (3.16e−5)**	**6.7117e−3(7.49e-6)**
DC3-DTLZ3	2.1293e−2 (5.47e−4) −	8.2046e−2 (1.93e−1) −	1.9912e−2 (2.25e−4) =	1.9974e−2 (2.64e−4) =	1.9912e−2 (2.08e−4) =	1.9952e−2 (2.56e−4)	1.9455e−2(6.43e−5)
DOC1	**1.4278e–1 (7.67e–2) +**	3.6553e+0 (3.39e+0) −	5.9111e+0 (5.83e+0) −	4.6963e−1 (4.43e−1) +	4.0679e+0 (4.22e+0) −	1.1658e+0 (1.71e+0)	**1.7700e−2(6.22e−3)**
DOC2	NaN (NaN)=	NaN (NaN)=	NaN (NaN)=	NaN (NaN)=	NaN (NaN)=	NaN (NaN)	NaN (NaN)
DOC3	NaN (NaN) −	6.9409e+2 (3.22e+2) −	7.4993e+2 (2.22e+2) −	7.3037e+2 (2.75e+2) −	6.3359e+2 (2.41e+2) −	**3.2916e+2 (1.20e+2)**	**1.8360e+1(3.68e+1)**
DOC4	1.5491e+0 (7.12e−1) −	8.5749e−1 (9.65e−1) −	1.3905e+0 (1.42e+0) −	5.0469e−1 (2.33e−1) −	1.1965e+0 (1.11e+0) −	**2.6558e−1 (1.44e−1)**	**8.1010e−2(4.66e−3)**
DOC5	NaN (NaN)=	**8.8563e+1 (3.27e+1)+**	NaN (NaN)=	NaN (NaN)=	NaN (NaN)=	NaN (NaN)	NaN (NaN)
DOC6	2.4702e+0 (8.56e−1) −	1.7062e+0 (1.32e+0) −	3.2490e+0 (3.93e+0) −	3.5357e+0 (4.65e+0) −	2.3713e+0 (2.34e+0) −	**9.7086e–1 (7.49e–1)**	**1.4610e–2 (8.05e–3)**
DOC7	**2.7304e+0 (1.54e+0) +**	4.7114e+0 (2.18e+0) =	6.2338e+0 (2.87e+0) −	7.1658e+0 (2.17e+0) −	5.3369e+0 (1.83e+0) −	4.3841e+0 (1.56e+0)	**1.9333e+0 (4.33e–1)**
DOC8	1.2715e+2 (7.67e+1) −	1.4655e+2 (7.20e+1) −	6.8208e+1 (5.38e+1) −	9.9605e+1 (6.92e+1) −	8.7843e+1 (5.44e+1) −	**8.4426e+0 (5.15e+0)**	**2.5547e+0 (4.96e–1)**
DOC9	1.5982e−1 (1.36e−1) −	1.3613e−1 (2.30e−1) −	1.7550e−1 (9.51e−2) −	1.9875e−1 (1.06e−1) −	1.5267e−1 (1.19e−1) −	**6.7183e–2 (5.13e–2)**	**3.0158e–2 (3.10e–3)**
MW1	1.8539e−3 (4.41e−5)−	1.6719e−3 (1.73e−5) −	1.6150e−3 (1.04e−5) −	1.6170e−3 (1.27e−5) −	1.5982e−3 (1.10e−5) =	1.5944e−3 (7.25e-6)	1.5838e−3(2.56e-6)
MW2	4.7158e−2 (2.80e−2) −	1.8916e−2 (7.63e−3) −	2.0745e−2 (8.73e−3) −	1.9261e−2 (1.77e−2) −	1.9302e−2 (9.57e−3) −	**1.4803e–2 (5.74e–3)**	**3.6100e–3 (2.56e–5)**
MW3	4.9975e−3 (1.58e−4) −	5.0266e−3 (2.69e−4) −	4.6427e−3 (1.68e−4) −	4.7412e−3 (1.76e−4) −	4.4749e−3 (1.50e−4) −	**4.3739e–3 (9.74e–5)**	**4.2084e–3 (6.14e–7)**
MW4	4.3044e−2 (3.14e−4) −	4.1744e−2 (5.33e−4) −	4.0734e−2 (3.35e−4) −	4.0631e−2 (3.30e−4) −	3.9748e−2 (2.72e−4) =	3.9682e−2 (1.72e−4)	3.9459e−2(2.97e−5)
MW5	**7.8361e–5 (2.92e–5) +**	1.7830e−2 (1.83e−2) −	5.6977e−4 (7.09e−4) −	3.5718e−4 (1.86e−4) −	1.0264e−3 (1.03e−3) −	2.2992e−4 (6.82e−5)	1.5514e−4(3.31e−5)
MW6	1.0806e−1 (1.55e−1) −	3.7107e−2 (8.86e−2) −	1.9412e−2 (1.41e−2) −	1.2383e−2 (8.07e−3) −	3.8373e−2 (8.24e−2) −	**6.9134e–3 (4.48e–3)**	**2.7113e–3 (1.01e–7)**
MW7	4.4784e−3 (1.37e−4) −	1.4865e−2 (2.02e−2)−	4.1135e−3 (2.32e−4) =	3.9606e−3 (1.27e−4) =	4.0428e−3 (1.80e−4) =	4.0353e−3 (1.36e−4)	3.8674e−3 (3.65e-6)
MW8	8.1825e−2 (5.04e−2) −	4.8347e−2 (4.73e−3) −	4.9073e−2 (1.43e−2) −	4.3995e−2 (2.47e−3) −	4.7121e−2 (5.73e−3) −	**4.2648e–2 (4.03e–4)**	**4.1905e–2 (5.13e–4)**
MW9	4.4306e−3 (1.04e−4) −	4.0813e−2 (1.41e−1) −	4.1716e−3 (9.36e−5) =	4.2585e−3 (1.18e−4) −	4.1764e−3 (7.43e−5) =	4.1490e−3 (4.61e−5)	4.0586e−3(1.67e−5)
MW10	1.4026e−1 (1.79e−1) −	5.4826e−2 (4.89e−2) −	4.6329e−2 (4.21e−2) −	2.0730e−2 (1.56e−2) −	4.2607e−2 (4.59e−2) −	**7.1128e–3 (3.53e–3)**	**3.3112e–3 (1.65e–5)**
MW11	6.3028e−3 (2.54e−4) −	2.8984e−2 (1.26e−1) −	5.8439e−3 (1.21e−4) −	6.0326e−3 (1.61e−4) −	5.8887e−3 (1.07e−4) −	**5.5924e−3 (4.87e−4)**	**4.8273e−3(3.41e−5)**
MW12	5.2534e−3 (1.79e−4) −	5.1715e−3 (1.65e−4) −	4.6549e−3 (7.02e−5) −	4.7725e−3 (1.34e−4) −	4.6671e−3 (1.37e−4) −	**4.5676e−3 (4.47e−5)**	**4.4975e−3(2.56e−5)**
MW13	1.2282e−1 (4.53e−2) −	8.0699e−2 (4.33e−2) −	6.1210e−2 (3.56e−2) −	3.4672e−2 (2.56e−2) −	8.1790e−2 (3.95e−2) −	**1.9538e–2 (8.14e–3)**	**1.0397e–2 (3.93e–5)**
MW14	1.0290e−1 (2.25e−2) −	1.2461e−1 (4.10e−3) −	9.7085e−2 (1.69e−3) −	9.6933e−2 (1.71e−3) −	9.5525e−2 (1.45e−3) −	**9.4550e–2 (1.06e–3)**	**9.2841e–2 (1.24e–4)**
LIRCMOP1	2.7275e−1 (2.06e−1) −	3.8373e−1 (1.45e−1) −	1.6786e−1 (7.57e−2) −	1.3937e−1 (9.18e−2) =	2.0377e−1 (1.05e−1) −	1.1579e−1 (4.27e−2)	**6.2341e–2 (2.97e–3)**

Table 2 continued

Problem	MOEADDAE	CMOEA-MS	CCMO	c-DPEA	EMCMO	MSEFAS	MSEFAS (best)
LIRCMOP2	9.1070e−2 (4.70e−2) −	3.0712e−1 (9.01e−2) −	7.5101e−2 (3.02e−2) =	8.9558e−2 (5.02e−2) −	7.8293e−2 (3.21e−2) −	6.4338e−2 (2.78e−2)	**2.5623e−2(1.17e−3)**
LIRCMOP3	1.2565e−1 (1.07e−1) =	3.5161e−1 (1.03e−1) −	1.5359e−1 (6.80e−2) =	1.5406e−1 (1.72e−1) =	2.0427e−1 (1.10e−1) −	1.3362e−1 (6.42e−2)	**6.1547e−2(1.68e−2)**
LIRCMOP4	**7.2807e−2 (8.79e−2) +**	3.0801e−1 (7.29e−2) −	1.6133e−1 (8.38e−2) =	1.5265e−1 (1.23e−1) =	1.7493e−1 (7.48e−2) −	1.3253e−1 (4.16e−2)	**6.5346e−2(8.32e−3)**
LIRCMOP5	5.0682e−2 (2.12e−2) −	1.9001e−2 (2.03e−2) −	7.0367e−3 (2.02e−3) −	8.7005e−3 (4.30e−3) −	7.3017e−3 (2.19e−3) −	**6.7290e−3 (3.93e−4)**	**6.0161e−3(2.80e−4)**
LIRCMOP6	6.3432e−2 (3.64e−2) −	1.7370e−2 (3.47e−2) −	6.0541e−3 (3.02e−4)=	7.9594e−3 (4.88e−3) −	6.1677e−3 (4.20e−4)=	6.4305e−3 (2.46e−4)	6.0602e−3(4.96e-6)
LIRCMOP7	3.6250e−2 (1.17e−2) −	7.9432e−3 (1.84e−3) −	7.8575e−3 (1.02e−3) −	7.5719e−3 (7.00e−4) −	7.7270e−3 (5.88e−4) −	**7.1214e−3 (7.47e−5)**	**6.9488e−3(5.27e-6)**
LIRCMOP8	3.1317e−2 (1.54e−2) −	9.1023e−3 (1.18e−2) −	7.2343e−3 (2.68e−4) −	7.2160e−3 (5.16e−4) −	7.2244e−3 (2.11e−4) −	**7.0374e−3 (1.49e−4)**	**6.8187e−3 (1.39e−5)**
LIRCMOP9	**3.2418e−3 (1.03e−4) +**	2.4700e−1 (1.55e−1) −	5.1032e−3 (1.66e−3) −	5.3977e−2 (3.46e−2) −	5.6304e−3 (1.95e−3) −	4.6463e−3 (1.25e−3)	**3.1311e−3(2.53e−5)**
LIRCMOP10	4.7575e−3 (1.31e−4)=	7.3710e−2 (4.52e−2) −	4.6943e−3 (1.79e−4) =	4.8127e−3 (1.39e−4) −	4.7352e−3 (1.59e−4) =	4.6248e−3 (1.06e−4)	**4.4465e−3 (3.23e−5)**
LIRCMOP11	2.5397e−3 (2.05e−4) −	7.7422e−2 (6.37e−2) −	2.3976e−3 (6.68e−5) =	2.3961e−3 (6.56e−5) =	2.3897e−3 (5.60e−5) =	2.3796e−3 (5.29e−5)	2.2788e−3(2.24e−5)
LIRCMOP12	3.2096e−3 (1.38e−4) −	6.5993e−2 (1.11e−1) −	3.0504e−3 (7.38e−5) =	3.0757e−3 (9.01e−5) =	3.0527e−3 (8.88e−5) =	3.0482e−3 (1.31e−4)	**2.7180e−3(1.25e−4)**
LIRCMOP13	9.7280e−2 (9.43e−4) −	9.2639e−2 (7.98e−4) −	9.3733e−2 (1.16e−3) −	9.3270e−2 (1.00e−3) −	9.1068e−2 (6.11e−4) =	9.0649e−2 (7.98e−4)	8.9225e−2(8.53e−5)
LIRCMOP14	9.9484e−2 (7.22e−4) −	9.5422e−2 (9.23e−4) =	9.6393e−2 (9.86e−4) −	9.5440e−2 (9.07e−4)=	9.5785e−2 (8.79e−4) −	9.4857e−2 (6.11e−4)	9.3985e−2(8.42e-6)
DASCMOP1	8.0037e−2 (1.55e−1) −	7.7480e−2 (1.32e−1) −	7.1166e−3 (5.05e−3) −	5.5960e−3 (4.28e−3) −	7.7094e−3 (4.28e−3) −	**3.8255e−3 (5.72e−4)**	**3.0331e−3 (4.52e−5)**
DASCMOP2	5.0545e−2 (4.36e−2) −	7.9459e−3 (4.04e−3) −	1.1475e−2 (1.89e−2) −	4.5916e−3 (1.22e−3) −	1.0609e−2 (1.04e−2) −	**4.4884e−3 (3.33e−4)**	**4.0628e−3(8.83e-6)**
DASCMOP3	8.8202e−2 (5.90e−2) −	1.2694e−1 (4.70e−2) −	5.3725e−2 (3.72e−2) −	**2.4785e−2 (1.12e−2) +**	4.8402e−2 (3.43e−2) −	3.3482e−2 (9.57e−3)	**1.3085e−2(5.37e−4)**
DASCMOP4	1.3136e−3 (4.32e−5) −	1.1434e−3 (1.18e−5) =	1.1391e−3 (1.16e−5) =	1.2481e−3 (5.26e−4) −	1.1380e−3 (1.32e−5) =	1.1342e−3 (1.06e−5)	1.1175e−3(2.71e-6)
DASCMOP5	3.0540e−3 (1.06e−4) −	**2.6777e−3 (6.43e−5) =**	2.8004e−3 (7.73e−5) −	3.0658e−3 (2.12e−4) −	2.7462e−3 (4.79e−5) =	2.7794e−3 (5.23e−5)	2.7310e−3(7.65e-6)
DASCMOP6	2.4387e−2 (1.83e−2) −	2.8828e−2 (3.31e−2) −	1.9236e−2 (4.86e−3) −	1.8408e−2 (2.67e−3) −	1.8868e−2 (4.47e−3) −	**1.2952e−2 (4.19e−4)**	**1.2413e−2 (1.28e−4)**
DASCMOP7	3.5659e−2 (1.56e−3) −	3.0909e−2 (4.89e−4) =	3.1159e−2 (5.24e−4) −	3.0853e−2 (6.30e−4) =	3.0811e−2 (5.49e−4) =	3.0700e−2 (6.18e−4)	2.9828e−2(3.08e-6)
DASCMOP8	4.6334e−2 (2.15e−3) −	3.9401e−2 (8.85e−4) −	3.9539e−2 (7.29e−4) −	3.9053e−2 (6.67e−4) =	3.9707e−2 (8.35e−4) −	3.8942e−2 (7.41e−4)	**3.7821e−2(1.04e−3)**
DASCMOP9	6.9856e−2 (1.39e−2)−	4.0713e−2 (1.09e−3) =	4.0416e−2 (1.08e−3) =	4.1726e−2 (1.07e−3) −	4.0209e−2 (9.63e−4) =	4.0076e−2 (1.17e−3)	**3.7994e−2 (8.07e−4)**
win/lose/tie	5/47/4	1/46/9	0/40/16	2/41/13	0/36/20		

Table 3 mean (standard deviation) of IGD obtained by MSEFAS and six compared algorithms on five benchmark test suites for 30 independent runs

Problem	TiGE2	CTAEA	PPS	ToP	DSPCMDE	MSCMO	MSEFAS
C1-DTLZ1	3.3283e−1 (1.21e−1)−	2.3408e−2 (2.33e−4)−	2.5937e−2 (7.95e−4)−	2.1026e−1 (1.67e−1)−	2.7925e−2 (2.25e−3)−	2.0065e−2 (1.53e−4)−	**1.9677e–2 (6.04e–5)**
C1-DTLZ3	5.0112e+0 (3.60e+0)−	7.1005e−2 (9.61e−3)−	3.6490e+0 (4.05e+0)−	1.2795e+0 (2.92e+0)−	6.1526e−1 (7.36e−1) −	5.3477e−2 (4.34e−4)−	**5.2589e–2 (4.31e–4)**
C2-DTLZ2	1.3862e−1 (1.21e−1) −	5.6306e−2 (1.12e−3)−	5.5687e−2 (1.89e−3)−	5.8325e−2 (2.29e−3)−	5.4603e−2 (2.12e−3) −	4.2565e−2 (5.33e−4)=	4.2352e−2 (3.85e−4)
C3-DTLZ4	1.9085e−1 (1.54e−2)−	1.1169e−1 (1.88e−3)−	1.2840e−1 (1.13e−2)−	1.4261e−1 (6.58e−3)−	1.2964e−1 (3.27e−3)−	1.3695e−1 (1.76e−1)−	**9.2652e–2 (8.49e–4)**
DC1-DTLZ1	4.2634e−1 (1.23e−1)−	1.5254e−2 (3.12e−4)−	3.8138e−2 (7.26e−2)−	1.8100e−2 (2.63e−3)−	3.0299e−1 (2.65e−1)−	1.2376e−2 (4.41e−3)−	**1.1364e–2 (7.96e–5)**
DC1-DTLZ3	1.2592e+0 (3.81e−1)−	4.3424e−2 (1.19e−3)−	2.3048e−1 (1.69e−1)−	5.4011e−1 (1.35e+0)−	4.7918e−1 (2.89e−1)−	3.3954e−2 (2.64e−4)=	3.3680e−2 (2.97e−4)
DC2-DTLZ1	3.5448e−1 (5.63e−2)−	2.3356e−2 (2.25e−4)−	4.1906e−2 (4.37e−2)−	NaN (NaN)−	2.9927e−2 (1.57e−3)−	2.0174e−2 (1.76e−4)−	**1.9768e−2 (6.50e−5)**
DC2-DTLZ3	9.2182e−1 (0.00e+0)−	6.3423e−2 (9.08e−4)−	4.6718e−1 (2.10e−1)−	NaN (NaN)−	4.7797e−1(2.11e−1)−	5.2860e−2(3.04e−4)−	**5.1702e−2 (1.58e−4)**
DC3-DTLZ1	1.0385e+0 (6.60e−1) −	9.5979e−3 (3.04e−4)−	1.9335e−1 (1.34e−1)−	3.8954e−1 (3.26e−1)−	5.1091e−1 (5.36e−1) −	1.9529e−2 (3.70e−2)−	**6.7729e−3 (3.16e−5)**
DC3-DTLZ3	2.6254e+0 (7.35e−1)−	2.6986e−2 (8.33e−4)−	1.5694e+0 (9.33e−1)−	5.5402e+0 (2.98e+0)−	5.5341e−1 (1.44e−1)−	1.5398e−1 (2.10e−1)−	**1.9952e–2 (2.56e–4)**
DOC1	3.0348e+0 (4.12e+0) −	5.1889e+2 (2.75e+2) −	8.4318e−2 (5.63e−2)+	**6.0354e−3 (3.79e−4)+**	3.1933e+2 (4.45e+2)−	1.3156e+0 (1.16e+0)−	1.1658e+0 (1.71e+0)
DOC2	NaN (NaN)=	NaN (NaN)=	**1.6070e–1 (9.22e–2)+**	NaN (NaN)=	4.1381e−1 (1.11e−1)+	NaN (NaN)=	NaN (NaN)
DOC3	5.6126e+2 (1.90e+2)−	NaN (NaN)−	2.7690e+2 (3.24e+2)−	1.6635e+2 (1.39e+2)+	**1.5318e+2 (2.08e+2)+**	NaN (NaN)−	3.2916e+2 (1.20e+2)
DOC4	2.2335e+0 (9.83e−1)−	2.4818e+2 (2.69e+2)−	2.8149e−1 (6.43e−2)−	9.2128e−2 (7.19e−2)+	**2.6003e−2 (2.54e−3)+**	8.6691e−1 (4.93e−1)−	2.6558e−1 (1.44e−1)
DOC5	8.6274e+1 (1.84e+1)+	NaN (NaN)=	3.5468e+1 (9.28e+1)+	**2.1831e+1 (4.83e+1)+**	NaN (NaN)=	NaN (NaN)=	NaN (NaN)
DOC6	1.2063e+0 (5.91e−1)−	2.5178e+1 (1.77e+1)−	5.0544e−1 (7.41e−2)+	5.2077e+0 (1.97e+0)−	**3.0214e−3 (1.63e−4)+**	9.7086e−1 (1.19e+0)=	9.7086e−1 (7.49e−1)
DOC7	4.0436e+0 (1.54e+0)=	NaN (NaN)−	4.7336e−1 (2.12e−1) +	9.7712e−1 (4.45e−1)+	**2.2399e−1 (9.14e−1)+**	4.3439+0 (1.80e+0)=	4.3841e+0 (1.56e+0)
DOC8	1.7050e+2 (7.42e+1) −	5.6581e+2 (0.00e+0) −	1.0510e+2 (4.34e+1)−	4.5687e+1(1.93e+1)−	**1.8855e−1 (3.59e−2)+**	9.8776e+1 (6.44e+1)−	8.4426e+0 (5.15e+0)
DOC9	**5.8123e−2 (1.13e−2)+**	6.7845e−1 (1.73e−1) −	3.1244e−1 (2.82e−2)−	2.1477e−1(4.57e−2)−	1.2033e−1 (1.22e−2)−	2.2331e−1 (6.58e−2)−	6.7183e−2 (5.13e−2)
MW1	1.3533e−2 (4.23e−3)−	2.0179e−3 (7.92e−5)−	2.6935e−3 (7.22e−5)−	3.3984e−2 (0.00e+0)−	2.0690e−3 (4.66e−5)−	1.6153e−3 (1.58e−5)−	**1.5944e−3 (7.25e-6)**
MW2	6.2814e−1 (7.84e−2)−	2.1925e−2 (1.91e−2)−	1.4970e−1 (9.28e−2)−	1.0916e−1(7.88e−3)−	5.3876e−2 (2.68e−2)−	2.5256e−2 (2.06e−2)−	**1.4803e–2 (5.74e–3)**
MW3	2.2277e–2 (3.14e–3)−	4.7708e−3 (1.70e−4)−	6.4329e−3 (6.32e−4)−	6.0050e−1 (4.45e−1)−	5.2092e−3 (1.78e−4)−	4.6882e−3 (2.15e−4)−	**4.3739e−3 (9.74e−5)**
MW4	1.1392e−1 (4.19e−2)−	4.6651e−2 (2.39e−4)−	5.4875e−2 (1.59e−3)−	1.3689e−1 (5.76e−2)−	6.1141e−2 (2.16e−3)−	4.0457e−2 (3.15e−4)−	**3.9682e−2 (1.72e−4)**
MW5	3.4607e−2 (9.08e−3)−	7.4394e−3 (1.65e−3)−	3.7730e−1 (3.75e−1)−	5.9609e−1 (3.00e−1)−	4.2424e−2 (2.85e−2)−	9.7238e−4 (8.74e−4)−	**2.2992e−4 (6.82e−5)**
MW6	2.3830e−1 (3.33e−1)−	1.0635e−2 (5.68e−3)−	5.1619e−1 (3.41e−1)−	5.9419e−1 (3.49e−1)−	1.6848e−1 (1.62e−1)−	2.1817e−2 (1.29e−2)−	**6.9134e−3 (4.48e−3)**
MW7	4.5101e−2 (2.87e−2)−	6.1739e−3 (3.40e−4)−	5.4064e−3 (3.90e−4)−	3.5879e−2 (9.82e−2)−	5.9061e−3 (7.03e−4)−	4.0111e−3 (1.90e−4)=	4.0353e−3 (1.36e−4)
MW8	6.8824e−1 (1.12e−1)−	5.4392e−2 (3.67e−3)−	2.1082e−1 (2.65e−1)−	4.3988e−1 (3.89e−1)−	9.1679e−2 (1.66e−2)−	4.6002e−2 (3.30e−3)−	**4.2648e–2 (4.03e–4)**
MW9	1.1860e−1 (1.99e−1)−	8.1999e−3 (6.15e−4)−	4.4377e−2 (1.57e−1)−	5.7794e−1 (2.86e−1)−	2.1328e−1 (2.72e−1)−	4.4112e−3 (2.27e−4)−	**4.1490e–3 (4.61e–5)**
MW10	6.1765e−2 (3.22e−2)−	1.6298e−2 (1.41e−2)−	6.1223e−1 (1.17e−1)−	7.1057e−1(1.57e−2)−	2.0737e−1 (1.38e−1)−	4.0442e−2 (3.26e−2)−	**7.1128e−3 (3.53e−3)**
MW11	3.5395e−2 (6.36e−3)−	1.2209e−2 (1.22e−3)−	7.5721e−3 (4.05e−4)−	2.1788e−1 (2.36e−1)−	7.6944e−1 (2.28e−16)−	5.9009e−3 (1.33e−4)−	**5.5924e−3 (4.87e−4)**
MW12	7.1684e−2 (1.67e−1)−	7.6166e−3 (6.24e−4)−	8.3372e−3 (6.55e−3)−	7.0758e−1 (2.76e−1)−	4.9630e−3 (1.26e−4)−	4.6610e−3 (6.39e−5)=	4.5676e−3 (4.47e−5)
MW13	1.2756e+0 (5.56e−1)−	3.7620e−2 (3.16e−2)−	4.6480e−1 (3.96e−1)−	5.8064e−1 (5.04e−1)−	1.7201e−1 (3.61e−2)−	7.9479e−2 (3.33e−2)−	**1.9538e−2 (8.14e−3)**
MW14	1.5746e−1 (7.73e−3)−	1.1268e−1 (4.28e−3)−	1.2914e−1 (7.15e−3)−	3.0642e−1 (2.08e−1)−	1.2491e−1 (6.28e−3)−	9.6839e−2 (1.60e−3)−	**9.4550e−2 (1.06e−3)**
LIRCMOP1	1.0601e−1(3.26e−2)=	2.6992e−1(1.42e−1)−	**1.0829e−2(5.16e−3)+**	1.4317e−1(1.10e−1)−	4.3084e−2 (2.17e−2)+	1.3658e−1 (7.81e−2)−	1.1579e−1 (4.27e−2)

Table 3 continued

Problem	TiGE2	CTAEA	PPS	ToP	DSPCMDE	MSCMO	MSEFAS
LIRCMOP2	1.1267e−1(2.50e−2)−	7.5337e−2(1.35e−2)−	**6.2635e−3(8.24e−4)+**	1.1764e−1(8.34e−2)−	3.3913e−2 (1.31e−2)+	8.9862e−2 (7.16e−2)−	6.4338e−2 (2.78e−2)
LIRCMOP3	1.2535e−1(3.32e−2)=	3.1562e−1(2.03e−1)−	**7.2799e−3(5.00e−3)+**	3.7670e−1(1.05e−1)−	7.0231e−2 (4.94e−2)+	1.2293e−1 (6.81e−2)=	1.3362e−1 (6.42e−2)
LIRCMOP4	1.2864e−1(1.22e−2)=	1.5175e−1(6.94e−2)−	**3.7804e−3(5.85e−4)+**	3.0167e−1(6.14e−2)−	8.2577e−2 (5.36e−2)+	8.3774e−2 (7.93e−2)+	1.3253e−1 (4.16e−2)
LIRCMOP5	3.0643e−1(1.09e−1)−	9.5888e−2(2.05e−2)−	7.0113e−3(8.77e−4)−	2.3690e−1(4.67e−1)−	**5.9282e−3 (1.82e−4)+**	1.0036e−2 (9.00e−3)−	6.7290e−3 (3.93e−4)
LIRCMOP6	4.2171e−1 (1.91e−1)−	1.5273e−1 (1.34e−1)−	7.8208e−3 (6.87e−4)−	5.3967e−2 (1.00e−1)−	**6.1448e−3 (2.58e−4)+**	7.4018e−3 (1.77e−3)−	6.4305e−3 (2.46e−4)
LIRCMOP7	2.0256e−1 (7.42e−2)−	1.7976e−2 (2.05e−3)−	9.9402e−3 (6.41e−4)−	8.7422e−3 (4.12e−4)−	7.9932e−3 (2.24e−4)−	1.3056e−2 (1.36e−2)−	**7.1214e−3 (7.47e−5)**
LIRCMOP8	2.9251e−1 (1.37e−1)−	1.9842e−2 (1.51e−2)−	1.0281e−2 (1.24e−3)−	8.6326e−3 (3.76e−4)−	8.1226e−3 (1.95e−4)−	1.2064e−2 (1.34e−2)−	**7.0374e−3 (1.49e−4)**
LIRCMOP9	6.9755e−1 (1.85e−1)−	6.3556e−2 (2.45e−2)−	**3.2220e−3 (1.34e−4)+**	3.1314e−1 (1.33e−1)−	4.1458e−2 (1.18e−1)−	8.9385e−2 (6.75e−2)−	4.6463e−3 (1.25e−3)
LIRCMOP10	4.9211e−1 (5.90e−2)−	5.8149e−2 (5.86e−2)−	5.3684e−3 (4.11e−4)−	5.5856e−3 (2.57e−4)−	5.0846e−3 (1.96e−4)−	4.8172e−3 (2.06e−4)−	**4.6248e−3 (1.06e−4)**
LIRCMOP11	6.4242e−1 (3.55e−1)−	1.2251e−1 (3.66e−2)−	2.3903e−3 (1.54e−4)=	1.1870e−1 (6.95e−2)−	2.3625e−3 (6.65e−5) =	2.3608e−3 (3.98e−5)=	2.3796e−3 (5.29e−5)
LIRCMOP12	4.3848e−1 (2.47e−1)−	1.9482e−2 (8.67e−3)−	3.1386e−3 (4.85e−5)=	1.6178e−2 (3.78e−2)−	3.4468e−3 (9.61e−4) −	3.4312e−3 (1.25e−3)−	3.0482e−3 (1.31e−4)
LIRCMOP13	3.5973e−1 (9.27e−2)−	1.0820e−1 (2.25e−3)−	1.2353e−1 (2.73e−3)−	1.2583e−1 (3.70e−3)−	1.2015e−1 (4.04e−3)−	9.3635e−2 (9.65e−4)−	**9.0649e−2 (7.98e−4)**
LIRCMOP14	6.3741e−1 (2.87e−1)−	1.1113e−1 (1.07e−3)−	1.1851e−1 (2.62e−3)−	1.2029e−1 (4.08e−3)−	1.1513e−1 (2.81e−3)−	9.5692e−2 (6.82e−4)−	**9.4857e−2 (6.11e−4)**
DASCMOP1	9.8741e−2 (7.22e−2)−	1.1135e−2 (1.29e−3)−	**3.2248e−3 (1.78e−4)+**	3.6265e−2 (1.48e−1)−	3.8419e−3 (5.36e−4)=	3.8001e−3 (7.19e−4)=	3.8255e−3 (5.72e−4)
DASCMOP2	2.8674e−2 (5.20e−3)−	7.6771e−3 (7.14e−4)−	5.0530e−3 (1.36e−4)−	5.1224e−3 (1.62e−4)−	4.8000e−3 (1.36e−4)−	**3.9294e−3 (6.90e−5)+**	4.4884e−3 (3.33e−4)
DASCMOP3	1.3074e−1 (5.83e−2)−	3.0142e−2 (7.57e−3)=	**1.9289e−2 (1.23e−4)+**	1.0824e−1 (6.35e−2)−	1.9545e−2 (2.86e−4)+	7.6172e−2 (5.51e−2)−	3.3482e−2 (9.57e−3)
DASCMOP4	2.2517e−1 (3.02e−1)−	1.1269e−2 (1.96e−3)−	3.7646e−2 (7.38e−2)−	6.2516e−1 (2.35e−1)−	1.6259e−3 (1.71e−4)−	1.1407e−3 (1.00e−5)=	1.1342e−3 (1.06e−5)
DASCMOP5	2.4652e−2 (3.87e−3)−	7.7964e−3 (7.14e−4)−	4.0705e−3 (1.46e−4)−	5.0213e−1 (2.60e−1)−	3.0935e−3 (1.57e−4)−	2.7473e−3 (6.82e−5)=	2.7794e−3 (5.23e−5)
DASCMOP6	9.1773e−2 (1.12e−1)−	2.6203e−2 (5.29e−3)−	2.3528e−2 (1.27e−2)−	6.6473e−1 (2.02e−1)−	1.7971e−2 (2.84e−3)−	1.6411e−2 (3.59e−3)−	**1.2952e−2 (4.19e−4)**
DASCMOP7	8.9545e−2 (7.46e−3)−	6.0661e−2 (5.22e−3)−	6.5817e−2 (1.12e−2)−	4.1658e−1 (1.44e−1)−	4.4965e−2 (2.89e−3)−	3.0367e−2 (6.34e−4)=	3.0700e−2 (6.18e−4)
DASCMOP8	2.4583e−1 (1.99e−1)−	9.2330e−2 (7.93e−3)−	8.6480e−2 (1.29e−2)−	5.6805e−1 (2.55e−1)−	5.7217e−2 (3.20e−3)−	3.9742e−2 (1.04e−3)=	3.8942e−2 (7.41e−4)
DASCMOP9	1.3894e−1 (2.15e−2)−	9.6088e−2 (8.02e−3)−	6.6657e−2 (3.50e−3)−	6.0053e−2 (2.04e−3)−	5.7478e−2 (3.06e−3)−	4.0364e−2 (1.04e−3)=	4.0076e−2 (1.17e−3)
Win/lose/tie	2/49/5	0/53/3	12/42/2	5/50/1	13/40/3	2/38/16	

dard deviation of HV for MSEFAS and the five comparison algorithms on the five benchmark test suites, based on 30 independent runs, can be found in Table SII of the Supplementary material. From the results in Table 3, we can see that MSEFAS obtains the best results on 26 out of 56 problems. It outperforms TiGE2, CTAEA, PPS, ToP, DSPCMDE, and MSCMO on 49, 53, 42, 50, 40, and 38 problems, respectively, and is comparable to these algorithms on 5, 3, 2, 1, 3, and 16 problems, respectively. These results show that MSEFAS performs better on most of the tested instances compared to the other algorithms.

Figures 6, 7, 8 and 9 show the feasible and non-dominated solution sets, as well as the running IGD values for all CMOEAs on LIRCMOP1, DOC8, MW14, and DASCMOP8, respectively. From these figures, we can observe that MSEFAS has better diversity than the other five CMOEAs on LIRCMOP1 (a type IV CMOP with many large infeasible regions and small feasible regions). The Pareto front of DOC8 is linear, disconnected, and has a small, nonlinear feasible region. MW14 has a disconnected geometry due to Pareto domination and is a type I CMOP.

The Pareto front of DASCMOP8 is composed of several disjoint segments blocked by large infeasible regions, and has multiple disconnected feasible regions. Figures 7, 8 and 9 show that MSEFAS is able to reach all non-dominated feasible regions and obtain well-distributed solutions on the CPF for DOC8, MW14, and DASCMOP8, especially when the feasible region is small or has multiple disconnected feasible regions, compared to the other five comparison algorithms.

Adaptive assignment of execution stage

To verify the effectiveness of the proposed MSEFAS in adaptively assigning the corresponding execution stage to each algorithm, we created five variants of MSEFAS by fixing the number of execution stages and denoted them as follows:

- MSEFAS(0,1/2,1/2): the first stage is not executed and the second and third stages are both executed for 1/2 of the total process.
- MSEFAS(1/4,1/4,1/2): the first and second stages are executed for 1/4 of the total process, and the third stage is executed for 1/2 of the total process.
- MSEFAS(1/3,1/3,1/3): the three stages are all executed for 1/3 of the total process.
- MSEFAS(1/2,1/4,1/4): the first stage is executed for 1/2 of the total process, and the second and third stages are both executed for 1/4 of the total process.
- MSEFAS(1/2,1/2,0): the first and second stages are executed for 1/2 of the total process, and the third stage is not executed.

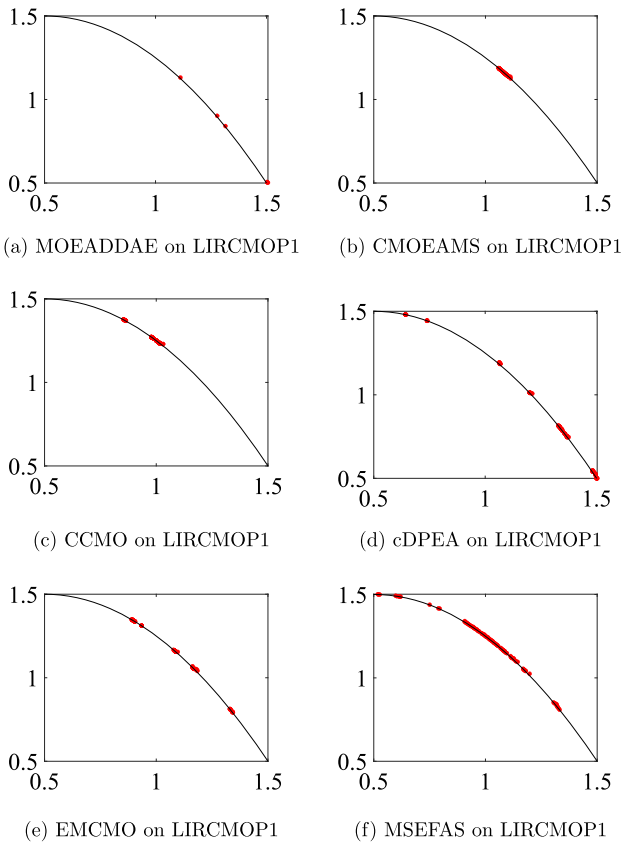


Fig. 6 Feasible and non-dominated solutions on LIRCMOP1 with the median IGD value among 30 runs

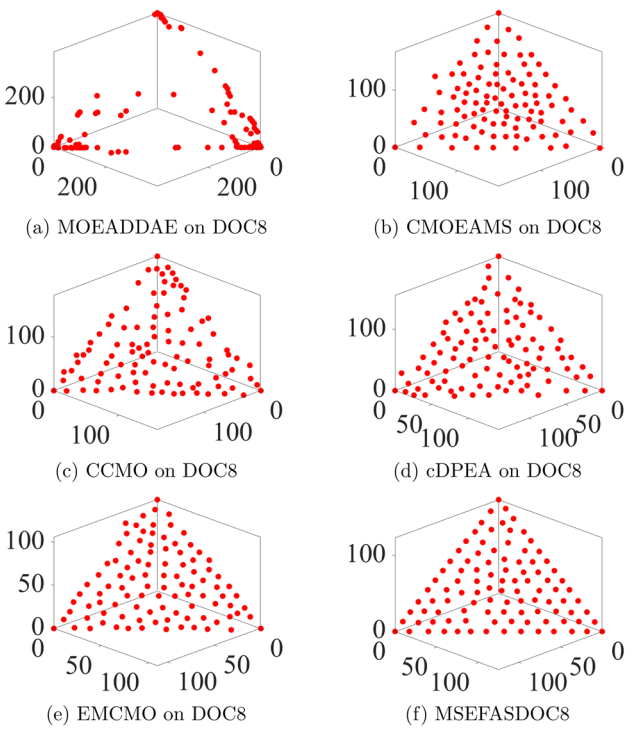


Fig. 7 Feasible and non-dominated solutions on DOC8 with the median IGD value among 30 runs

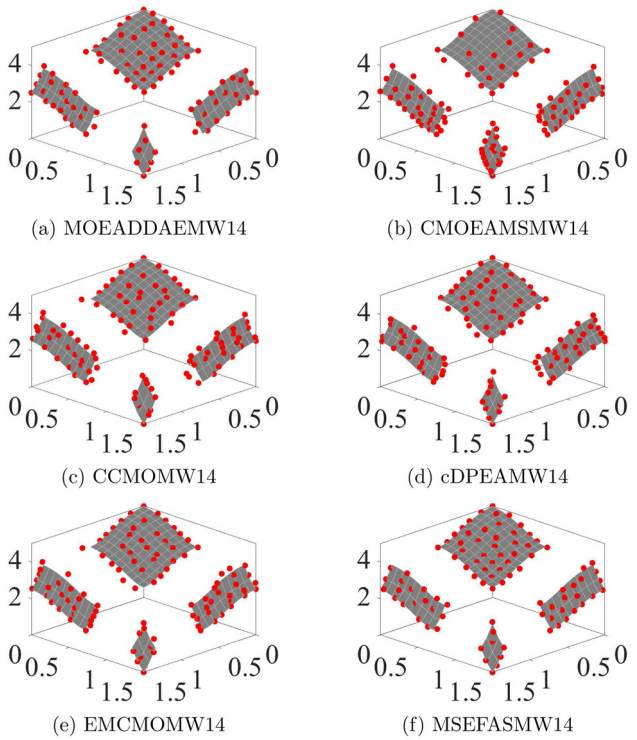


Fig. 8 Feasible and non-dominated solutions on MW14 with the median IGD value among 30 runs

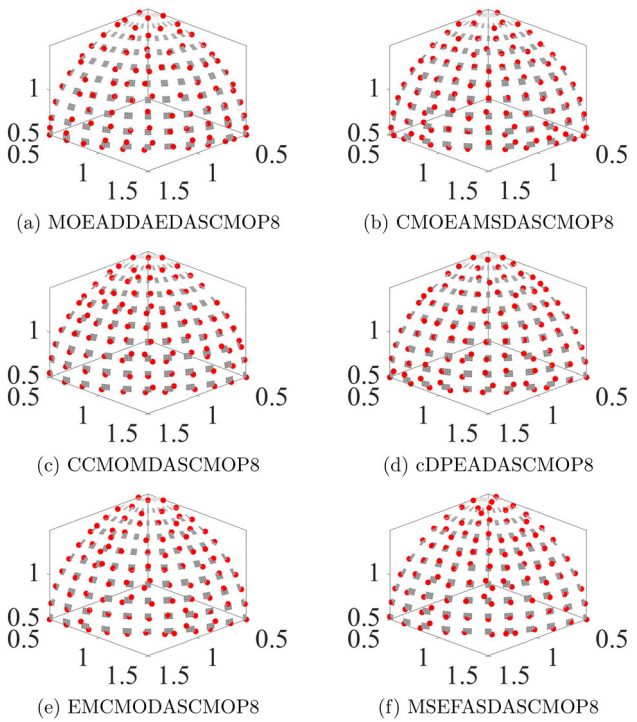


Fig. 9 Feasible and non-dominated solutions on DASCMOP8 with the median IGD value among 30 runs

We compared MSEFAS with its five variants and the results for IGD and HV are shown in Table 4 and Table SIII. The results show that MSEFAS outperforms all five variants, demonstrating the effectiveness of the proposed method in adaptively assigning the execution stage to each algorithm.

In order to verify which of the three parts of the proposed MSEFAS plays an effective role, we also design three variants of the proposed algorithm, namely the algorithm with only the IF and OF parts (MSEFAS(IFOF)), the algorithm with only the IF and PO parts (MSEFAS(IFPO)), and the algorithm with only the OF and PO parts (MSEFAS(OFPO)). Table 5 shows the mean values of IGD obtained by MSEFAS and the three variant algorithms on five benchmark test suites for 30 independent runs.

As can be seen from Table 5, MSEFAS is superior to the three variant algorithms. Among the three variant algorithms, MSEFAS(IFPO) and MSEFAS(OFPO) perform better than MSEFAS(IFOF), and the common point of these two algorithms is that they both contain the PO algorithm, so it can be concluded that the PO algorithm plays an important role in MSEFAS. Although the effectiveness of the PO algorithm for MSEFAS is indispensable, it cannot exceed the MSEFAS algorithm, which also indicates that the effectiveness of MSEFAS is not only caused by the simple superposition of the IF algorithm, OF algorithm, and PO algorithm, but is mainly attributed to the fact that there is a strategy in MSEFAS that can adaptively provide suitable initial optimized populations and the execution stage of each algorithm for different test problems.

Adaptive selection of the initial optimized population

To verify the effectiveness of the proposed adaptive selection of the initial optimized population, we compared it to an optimization algorithm without adaptive selection of the initial population. The initial optimized populations without adaptive selection can be divided into two types: those given only by c-DPEA (denoted as MSEFAS(cDPEAPOP)) and those given only by CCMO (denoted as MSEFAS(CCMOPOP)). To ensure fairness, we made sure that the parameter settings in MSEFAS(cDPEAPOP) and MSEFAS(CCMOPOP) were the same as in MSEFAS. The results of the mean and standard deviation of IGD for MSEFAS and the two comparison algorithms on the five benchmark test suites, based on 30 independent runs, are shown in Table 6. The HV test results can be found in Table S-III in the Supplementary material. From these results, we can see that MSEFAS performs better than both MSEFAS(cDPEAPOP) and MSEFAS(CCMOPOP), demonstrating the effectiveness of the proposed adaptive selection of the initial optimized population.

Table 4 Mean (standard deviation) of IGD obtained by MSEFAS and five compared algorithms of its variants on five benchmark test suites for 30 independent runs

Problem	MSEFAS(0,1/2,1/2)	MSEFAS(1/4,1/4,1/2)	MSEFAS(1/3,1/3,1/3)	MSEFAS(1/2,1/4,1/4)	MSEFAS(1/2,1/2,0)	MSEFAS
C1-DTLZ1	1.9774e−2 (8.36e−5)−	1.9797e−2 (9.26e−5)−	1.9776e−2 (1.03e−4)−	1.9762e−2 (1.15e−4)−	1.9745e−2 (9.22e−5)−	**1.9677e−2 (6.04e−5)**
C1-DTLZ3	5.2811e−2 (6.52e−4)=	5.2968e−2 (6.17e−4)−	5.3048e−2 (6.64e−4)−	5.2908e−2 (7.06e−4)−	5.2822e−2 (5.62e−4)=	5.2589e−2 (4.31e−4)
C2-DTLZ2	4.2490e−2 (4.67e−4)=	4.2439e−2 (5.68e−4)=	4.2397e−2 (5.49e−4)=	4.2380e−2 (4.11e−4)=	4.2422e−2 (4.22e−4)=	4.2352e−2 (3.85e−4)
C3-DTLZ4	9.4942e−2 (1.60e−3)−	9.5033e−2 (1.26e−3)−	9.5115e−2 (1.23e−3)−	9.5314e−2 (1.24e−3)−	1.1950e−1 (1.36e−1)−	**9.2652e−2 (8.49e−4)**
DC1-DTLZ1	1.1393e−2 (6.88e−5)=	1.1376e−2 (7.18e−5)=	1.1374e−2 (7.67e−5)=	1.1386e−2 (6.87e−5)=	1.1339e−2 (6.21e−5)=	1.1364e−2 (7.96e−5)
DC1-DTLZ3	3.3715e−2 (2.98e−4)=	3.3829e−2 (3.81e−4)−	3.3787e−2 (3.31e−4)−	3.3891e−2 (3.78e−4)−	3.3707e−2 (2.98e−4)=	3.3680e−2 (2.97e−4)
DC2-DTLZ1	1.9816e−2 (8.73e−5)−	1.9842e−2 (1.23e−4)−	1.9805e−2 (9.28e−5)=	1.9836e−2 (7.89e−5)−	1.9835e−2 (1.11e−4)−	1.9768e−2 (6.50e−5)
DC2-DTLZ3	5.2233e−2 (3.38e−4)−	5.2254e−2 (3.84e−4)−	5.2164e−2 (3.43e−4)−	5.2407e−2 (3.88e−4)−	5.2221e−2 (4.09e−4)−	**5.1702e−2 (1.58e−4)**
DC3-DTLZ1	6.8311e−3 (5.33e−5)−	6.8237e−3 (5.51e−5)−	6.7967e−3 (4.67e−5)−	6.8298e−3 (4.97e−5)−	6.8526e−3 (5.47e−5)−	**6.7729e–3 (3.16e–5)**
DC3-DTLZ3	1.9958e−2 (2.11e−4)=	1.9989e−2 (2.25e−4)=	1.9955e−2 (1.89e−4)=	1.9993e−2 (3.05e−4)=	2.0044e−2 (3.11e−4)=	1.9952e−2 (2.56e−4)
DOC1	6.2021e+0 (4.67e+0)−	3.2183e+0 (4.22e+0)−	3.4514e+0 (4.65e+0)−	9.5714e+0 (1.17e+1)−	7.5254e+0 (5.28e+0)−	**1.1658e+0 (1.71e+0)**
DOC2	NaN (NaN)=	NaN (NaN)=	NaN (NaN)=	NaN (NaN)=	NaN (NaN)=	NaN (NaN)
DOC3	7.3584e+2 (1.86e+2)−	5.8426e+2 (1.02e+2)−	5.8021e+2 (2.77e+2)−	6.8208e+2 (1.68e+2)−	5.7600e+2 (2.18e+2)−	**3.2916e+2 (1.20e+2)**
DOC4	8.8737e−1 (8.93e−1)−	7.8464e−1 (6.80e−1)−	8.9947e−1 (9.65e−1)−	1.3538e+0 (1.15e+0)−	1.4041e+0 (1.10e+0)−	**2.6558e−1 (1.44e−1)**
DOC5	NaN (NaN)=	NaN (NaN)=	NaN (NaN)=	NaN (NaN)=	NaN (NaN)=	NaN (NaN)
DOC6	1.4820e+0 (1.94e+0)−	2.2547e+0 (2.86e+0)−	2.5197e+0 (2.74e+0)−	2.5947e+0 (2.06e+0)−	3.3832e+0 (2.82e+0)−	**9.7086e−1 (7.49e−1)**
DOC7	5.7965e+0 (2.21e+0)−	5.6604e+0 (1.86e+0)−	5.8449e+0 (1.98e+0)−	6.0362e+0 (2.32e+0)−	5.9217e+0 (2.43e+0)−	**4.3841e+0 (1.56e+0)**
DOC8	7.2975e+1 (6.59e+1)−	8.0020e+1 (6.51e+1)−	7.4962e+1 (6.29e+1)−	9.9813e+1 (7.12e+1)−	1.1047e+2 (9.92e+1)−	**8.4426e+0 (5.15e+0)**
DOC9	1.5596e−1 (9.98e−2)−	1.5348e−1 (9.81e−2)−	1.5613e−1 (1.10e−1)−	1.6774e−1 (1.14e−1)−	2.0532e−1 (1.08e−1)−	**6.7183e–2 (5.13e–2)**
MW1	1.6006e−3 (8.63e-6)−	1.5979e−3 (1.08e−5)=	1.5990e−3 (1.13e−5)=	1.5990e−3 (1.15e−5)=	1.5951e−3 (1.03e−5)=	1.5944e−3 (7.25e-6)
MW2	1.4705e−2 (7.12e−3)=	1.3914e−2 (5.34e−3)=	1.4233e−2 (6.90e−3)=	2.1839e−2 (7.55e−3)−	2.1417e−2 (1.52e−2)−	1.4803e−2 (5.74e−3)
MW3	4.4270e−3 (1.41e−4)=	4.5109e−3 (1.74e−4)−	4.5118e−3 (1.43e−4)−	4.8697e−3 (3.50e−4)−	4.7832e−3 (3.54e−4)−	4.3739e−3 (9.74e−5)
MW4	3.9778e−2 (2.91e−4)=	3.9747e−2 (2.78e−4)=	3.9797e−2 (2.43e−4)=	3.9919e−2 (2.48e−4)−	4.0159e−2 (3.82e−4)−	3.9682e−2 (1.72e−4)
MW5	3.7067e−4 (4.58e−4)−	2.9265e−4 (2.10e−4)=	7.0051e−4 (1.01e−3)−	2.3964e−3 (3.99e−3)−	2.1580e−3 (4.00e−3)−	2.2992e−4 (6.82e−5)
MW6	9.8856e−3 (6.39e−3)−	1.2288e−2 (6.65e−3)−	1.3480e−2 (7.67e−3)−	2.4387e−2 (2.16e−2)−	2.0634e−2 (1.19e−2)−	**6.9134e−3 (4.48e−3)**
MW7	4.1403e−3 (2.60e−4)−	4.1545e−3 (1.55e−4)−	4.2255e−3 (3.04e−4)−	4.5100e−3 (4.40e−4)−	4.3675e−3 (4.62e−4)−	**4.0353e−3 (1.36e−4)**
MW8	4.3911e−2 (1.81e−3)−	4.4337e−2 (1.74e−3)−	4.4852e−2 (2.20e−3)−	4.4166e−2 (2.35e−3)−	4.5540e−2 (4.71e−3)−	**4.2648e−2 (4.03e−4)**
MW9	4.1813e−3 (1.06e−4)=	4.2844e−3 (1.81e−4)−	4.3540e−3 (2.77e−4)−	4.9763e−3 (2.16e−3)−	5.1356e−2 (1.78e−1)−	4.1490e−3 (4.61e−5)
MW10	1.7801e−2 (1.50e−2)−	2.2045e−2 (1.54e−2)−	2.0487e−2 (2.08e−2)−	3.2514e−2 (2.06e−2)−	3.1356e−2 (2.19e−2)−	**7.1128e−3 (3.53e−3)**
MW11	5.8997e−3 (1.49e−4)−	5.9526e−3 (1.37e−4)−	5.9278e−3 (2.39e−4)−	6.0842e−3 (3.73e−4)−	6.0291e−3 (2.65e−4)−	**5.5924e−3 (4.87e−4)**
MW12	4.6533e−3 (8.17e−5)−	4.7324e−3 (1.23e−4)−	4.7467e−3 (1.23e−4)−	2.7162e−2 (1.22e−1)−	5.2890e−2 (1.83e−1)−	**4.5676e−3 (4.47e−5)**
MW13	5.6906e−2 (3.80e−2)−	4.5312e−2 (2.64e−2)−	3.8702e−2 (2.69e−2)−	5.8068e−2 (4.46e−2)−	6.9589e−2 (3.57e−2)−	**1.9538e−2 (8.14e−3)**
MW14	9.5853e−2 (1.67e−3)−	9.6182e−2 (1.38e−3)−	9.6283e−2 (2.03e−3)−	9.6716e−2 (1.66e−3)−	9.6536e−2 (1.80e−3)−	**9.4550e−2 (1.06e−3)**
LIRCMOP1	1.0761e−1 (3.98e−2)=	1.7052e−1 (7.10e−2)−	1.4091e−1 (6.64e−2)=	2.0974e−1 (8.62e−2)−	2.3207e−1 (8.50e−2)−	1.1579e−1 (4.27e−2)

Table 4 continued

Problem	MSEFAS(0,1/2,1/2)	MSEFAS(1/4,1/4,1/2)	MSEFAS(1/3,1/3,1/3)	MSEFAS(1/2,1/4,1/4)	MSEFAS(1/2,1/2,0)	MSEFAS
LIRCMOP2	5.1908e−2 (2.13e−2)=	8.5970e−2 (3.76e−2)−	8.2560e−2 (3.20e−2)−	1.1069e−1 (5.42e−2)−	1.0287e−1 (5.10e−2)−	6.4338e−2 (2.78e−2)
LIRCMOP3	1.0975e−1 (4.70e−2)=	1.7808e−1 (6.91e−2)−	1.4425e−1 (4.71e−2)=	2.1272e−1 (8.29e−2)−	2.2949e−1 (9.92e−2)−	1.3362e−1 (6.42e−2)
LIRCMOP4	1.0878e−1 (4.46e−2)=	1.3379e−1 (4.83e−2)=	1.4954e−1 (5.67e−2)=	2.0274e−1 (7.38e−2)−	1.9637e−1 (8.42e−2)−	1.3253e−1 (4.16e−2)
LIRCMOP5	8.6919e−3 (4.48e−3)−	8.1733e−3 (3.79e−3)−	9.5804e−3 (8.20e−3)−	7.3699e−3 (3.48e−3)−	7.6209e−3 (1.94e−3)−	**6.7290e–3 (3.93e–4)**
LIRCMOP6	6.3796e−3 (4.17e−4)=	6.8605e−3 (8.50e−4)−	7.1912e−3 (3.03e−3)−	6.7484e−3 (2.55e−3)=	7.2378e−3 (1.84e−3)−	6.4305e−3 (2.46e−4)
LIRCMOP7	8.2342e−3 (1.33e−3)−	7.9887e−3 (9.42e−4)−	7.7915e−3 (7.46e−4)−	7.7612e−3 (6.10e−4)−	8.5836e−3 (4.23e−3)−	**7.1214e−3 (7.47e−5)**
LIRCMOP8	7.2990e−3 (4.65e−4)−	7.3911e−3 (1.69e−4)−	7.4246e−3 (3.99e−4)−	7.2558e−3 (2.07e−4)−	7.3917e−3 (5.53e−4)−	**7.0374e–3 (1.49e–4)**
LIRCMOP9	9.9791e−3 (1.35e−2)−	5.7516e−3 (2.16e−3)−	3.0690e−2 (3.44e−2)−	6.3276e−3 (4.60e−3)−	8.3775e−3 (1.36e−2)−	**4.6463e–3 (1.25e–3)**
LIRCMOP10	4.8019e−3 (2.62e−4)−	5.0302e−3 (2.07e−4)−	4.8612e−3 (1.95e−4)−	4.7412e−3 (2.00e−4)=	4.7530e−3 (1.86e−4)=	4.6248e−3 (1.06e−4)
LIRCMOP11	2.4034e−3 (7.71e−5)=	2.3863e−3 (4.97e−5)=	2.3880e−3 (4.76e−5)=	2.3925e−3 (7.94e−5)=	2.4166e−3 (6.19e−5)=	2.3796e−3 (5.29e−5)
LIRCMOP12	3.0481e−3 (1.07e−4)=	3.1894e−3 (4.31e−4)−	3.0447e−3 (8.66e−5)=	3.0615e−3 (1.03e−4)=	3.0718e−3 (9.73e−5)=	3.0482e−3 (1.31e−4)
LIRCMOP13	9.0978e−2 (9.01e−4)=	9.0868e−2 (8.18e−4)=	9.1007e−2 (7.67e−4)=	9.0666e−2 (6.74e−4)=	9.0957e−2 (8.32e−4)=	9.0649e−2 (7.98e−4)
LIRCMOP14	9.5728e−2 (7.92e−4)−	9.5656e−2 (8.01e−4)−	9.5791e−2 (9.08e−4)−	9.5549e−2 (7.37e−4)=	9.5451e−2 (9.99e−4)=	**9.4857e−2 (6.11e−4)**
DASCMOP1	1.2847e−2 (2.74e−2)−	8.5212e−3 (3.78e−3)−	1.4090e−2 (2.68e−2)−	1.0214e−2 (5.89e−3)−	1.5146e−2 (2.68e−2)−	**3.8255e–3 (5.72e–4)**
DASCMOP2	1.5005e−2 (1.38e−2)−	1.9875e−2 (1.90e−2)−	2.2532e−2 (1.82e−2)−	3.0853e−2 (2.45e−2)−	2.8754e−2 (2.20e−2)−	**4.4884e−3 (3.33e−4)**
DASCMOP3	4.7036e−2 (3.60e−2)−	5.4347e−2 (4.10e−2)−	4.3925e−2 (2.95e−2)−	7.3076e−2 (5.05e−2)−	5.6243e−2 (4.24e−2)−	**3.3482e−2 (9.57e−3)**
DASCMOP4	1.2691e−3 (5.49e−4)−	1.1654e−3 (8.50e−5)=	1.3432e−3 (7.33e−4)−	1.1415e−3 (1.26e−5)=	1.1409e−3 (1.19e−5)=	1.1342e−3 (1.06e−5)
DASCMOP5	2.9283e−3 (1.31e−4)−	2.9985e−3 (1.27e−4)−	3.0134e−3 (1.74e−4)−	2.8384e−3 (1.29e−4)−	2.8210e−3 (1.22e−4)−	**2.7794e–3 (5.23e–5)**
DASCMOP6	1.9269e−2 (1.92e−3)−	1.9636e−2 (1.79e−3)−	1.9016e−2 (2.19e−3)−	2.0729e−2 (6.53e−3)−	2.0078e−2 (4.80e−3)−	**1.2952e–2 (4.19e–4)**
DASCMOP7	3.1222e–2 (9.11e–4)−	3.1363e−2 (1.06e−3)−	3.1052e−2 (7.35e−4)−	3.0697e−2 (7.07e−4)=	3.0742e−2 (7.58e−4)=	3.0700e−2 (6.18e−4)
DASCMOP8	3.9517e−2 (9.56e−4)−	3.9532e−2 (9.43e−4)−	3.9446e−2 (1.13e−3)−	3.9192e−2 (9.26e−4)=	3.9486e−2 (9.21e−4)−	3.8942e−2 (7.41e−4)
DASCMOP9	4.0266e−2 (7.47e−4)=	3.9879e−2 (8.90e−4)=	4.0526e−2 (1.64e−3)=	4.0482e−2 (1.29e−3)=	4.0649e−2 (1.70e−3)=	4.0076e−2 (1.17e−3)
Win/lose/tie	0/36/20	0/42/14	0/40/16	0/40/16	0/40/16	

Table 5 mean (standard deviation) of IGD obtained by MSEFAS and three compared algorithms of its variants on five benchmark test suites for 30 independent runs

Problem	MSEFAS (IFOF)	MSEFAS (IFPO)	MSEFAS (OFPO)	MSEFAS
C1-DTLZ1	2.0012e−2 (9.68e−5)−	1.9761e−2 (9.14e−5) =	1.9724e−2 (1.18e−4)=	1.9677e−2 (6.04e−5)
C1-DTLZ3	5.3707e−2 (5.44e−4)−	5.3118e−2 (5.89e−4) =	5.2926e−2(5.38e−4)=	5.2589e−2 (4.31e−4)
C2-DTLZ2	4.2827e−2 (5.64e−4)−	4.2407e−2 (4.29e−4) =	4.2520e−2(5.10e−4)=	4.2352e−2 (3.85e−4)
C3-DTLZ4	9.6934e−2 (1.49e−3)−	9.5392e−2 (1.07e−3)−	9.4967e−2(1.50e−3)=	9.2652e−2 (8.49e−4)
DC1-DTLZ1	1.1413e−2 (7.08e−5) =	1.1385e−2 (4.75e−5) =	1.1378e−2 (8.98e−5)=	1.1364e−2 (7.96e−5)
DC1-DTLZ3	3.3853e−2 (4.92e−4) =	3.3874e−2 (4.27e−4) =	3.3601e−2(3.08e−4)=	3.3680e−2 (2.97e−4)
DC2-DTLZ1	2.0112e−2 (1.39e−4)−	1.9846e−2 (7.41e−5) =	1.9833e−2(9.86e−5)=	1.9768e−2 (6.50e−5)
DC2-DTLZ3	5.2990e−2 (3.43e−4)−	5.2177e−2 (2.81e−4) =	5.2234e−2 (3.08e−4)−	5.1702e−2 (1.58e−4)
DC3-DTLZ1	6.8200e−3 (5.33e−5)−	6.8409e−3 (5.61e−5)−	6.8328e−3(7.26e−5)−	**6.7729e−3 (3.16e−5)**
DC3-DTLZ3	1.9907e−2 (2.34e−4) =	1.9966e−2 (1.76e−4) =	1.9948e−2(2.84e−4)=	1.9952e−2 (2.56e−4)
DOC1	2.5295e+0 (3.19e+0)−	4.9291e+0 (8.96e+0)−	4.9933e+0(3.33e+0)−	**1.1658e+0 (1.71e+0)**
DOC2	NaN (NaN)=	NaN (NaN)=	NaN (NaN)=	NaN (NaN)
DOC3	6.6173e+2 (1.93e+2)−	7.0093e+2 (1.87e+2)−	6.4723e+2(2.30e+2)−	**3.2916e+2 (1.20e+2)**
DOC4	9.3145e−1 (9.04e−1)−	9.3102e−1 (5.65e−1)−	6.9421e−1(4.47e−1)−	**2.6558e–1 (1.44e–1)**
DOC5	NaN (NaN)=	NaN (NaN)=	NaN (NaN)=	NaN (NaN)
DOC6	2.4127e+0 (1.54e+0)−	2.1126e+0 (1.98e+0)−	1.9936e+0(2.36e+0)−	**9.7086e–1 (7.49e–1)**
DOC7	6.1733e+0 (2.49e+0)−	6.1663e+0 (2.86e+0)−	5.5005e+0(1.88e+0)−	**4.3841e+0 (1.56e+0)**
DOC8	1.0282e+2 (7.20e+1)−	8.4927e+1 (7.32e+1)−	1.0557e+2(7.47e+1)−	**8.4426e+0 (5.15e+0)**
DOC9	1.4664e−1 (1.01e−1)−	1.5180e−1 (1.04e−1)−	1.6181e−1 (1.16e−1)−	**6.7183e–2 (5.13e–2)**
MW1	1.6138e−3 (9.11e-6)−	1.5969e−3 (6.89e-6)=	1.5986e−3 (7.85e-6)=	1.5944e−3 (7.25e-6)
MW2	1.4298e−2 (6.22e−3) =	2.1780e−2 (5.43e−3)−	1.3954e−2 (6.92e−3)=	1.4803e−2 (5.74e−3)
MW3	4.6331e−3 (1.64e−4)−	4.5836e−3 (2.66e−4)−	4.4561e−3 (1.54e−4)=	4.3739e−3 (9.74e−5)
MW4	4.0733e−2 (3.44e−4)−	3.9722e−2 (2.23e−4) =	3.9814e−2 (2.51e−4)=	3.9682e−2 (1.72e−4)
MW5	9.0864e−4 (1.79e−3)−	7.6771e−4 (1.21e−3)−	2.6839e−4 (2.69e−4)−	**2.2992e–4 (6.82e–5)**
MW6	1.4735e−2 (1.12e−2)−	9.8957e−3 (7.87e−3)−	1.0549e−2 (6.97e−3)−	**6.9134e–3 (4.48e–3)**
MW7	4.3974e−3 (3.93e−4)−	4.0574e−3 (2.72e−4)=	4.1210e−3 (1.85e−4)=	4.0353e−3 (1.36e−4)
MW8	4.6613e−2 (4.90e−3)−	4.4224e−2 (1.51e−3)−	4.4131e−2 (1.46e−3)−	**4.2648e–2 (4.03e–4)**
MW9	4.3125e−3 (1.31e−4)−	4.3042e−3 (1.58e−4)−	4.1744e−3 (7.32e−5)=	4.1490e−3 (4.61e−5)
MW10	2.4419e−2 (2.95e−2)−	3.7950e−2 (2.67e−2)−	1.7264e−2 (1.35e−2)−	**7.1128e–3 (3.53e–3)**
MW11	6.0107e−3 (2.43e−4)−	5.8546e−3 (3.58e−4)−	5.9621e−3 (1.13e−4)−	**5.5924e–3 (4.87e–4)**
MW12	4.6965e−3 (7.27e−5)−	4.7122e−3 (1.07e−4)−	4.6664e−3 (9.46e−5)−	**4.5676e–3 (4.47e–5)**
MW13	4.5959e−2 (3.06e−2)−	4.3017e−2 (2.97e−2)−	4.3493e−2 (2.49e−2)−	**1.9538e–2 (8.14e–3)**
MW14	9.7401e−2 (1.90e−3)−	9.6261e−2 (1.46e−3)−	9.5400e−2 (1.35e−3)−	**9.4550e–2 (1.06e–3)**
LIRCMOP1	1.6554e−1 (7.01e−2)−	1.9555e−1 (8.95e−2)−	9.0926e−2 (3.49e−2)=	1.1579e−1 (4.27e−2)
LIRCMOP2	8.1469e−2 (2.03e−2)−	9.2970e−2 (5.48e−2)−	5.7436e−2 (2.06e−2)=	6.4338e−2 (2.78e−2)
LIRCMOP3	1.8255e−1 (5.90e−2)−	1.7630e−1 (8.73e−2)−	1.1153e−1 (4.08e−2)=	1.3362e−1 (6.42e−2)
LIRCMOP4	1.8219e−1 (6.79e−2)−	2.0331e−1 (5.18e−2)−	1.2625e−1 (4.05e−2)=	1.3253e−1 (4.16e−2)
LIRCMOP5	8.1333e−3 (2.68e−3)−	8.2975e−3 (3.15e−3)−	7.1366e−3 (5.95e−4)−	**6.7290e–3 (3.93e–4)**
LIRCMOP6	6.7590e−3 (3.24e−4)−	6.7649e−3 (2.45e−3)−	6.4162e−3 (7.47e−4)=	6.4305e−3 (2.46e−4)
LIRCMOP7	8.4080e−3 (3.90e−3)−	7.8871e−3 (8.33e−4)−	7.8683e−3 (1.17e−3)−	**7.1214e−3 (7.47e−5)**

Table 5 continued

Problem	MSEFAS (IFOF)	MSEFAS (IFPO)	MSEFAS (OFPO)	MSEFAS
LIRCMOP8	7.2627e−3 (2.78e−4)−	7.4004e−3 (2.19e−4)−	7.3740e−3 (5.88e−4)−	**7.0374e–3 (1.49e–4)**
LIRCMOP9	1.0063e−1 (5.39e−2)−	4.5923e−3 (1.54e−3) =	8.8884e−3 (1.65e−2)−	4.6463e−3 (1.25e−3)
LIRCMOP10	4.7499e−3 (1.73e−4) =	4.8709e−3 (1.96e−4)−	4.7440e−3 (2.20e−4)=	4.6248e−3 (1.06e−4)
LIRCMOP11	2.4063e−3 (4.09e−5)=	2.3762e−3 (6.12e−5)=	2.3851e−3 (7.90e−5)=	2.3796e−3 (5.29e−5)
LIRCMOP12	3.3468e−3 (1.35e−3)−	3.0290e−3 (9.92e−5) =	3.0383e−3 (9.52e−5)=	3.0482e−3 (1.31e−4)
LIRCMOP13	9.3861e−2 (9.38e−4)−	9.0627e−2 (6.78e−4) =	9.1050e−2 (7.89e−4)=	9.0649e−2 (7.98e−4)
LIRCMOP14	9.5353e−2 (6.24e−4)=	9.5778e−2 (8.10e−4)−	9.5934e−2 (1.31e−3)−	9.4857e−2 (6.11e−4)
DASCMOP1	9.3940e−3 (5.17e−3)−	7.4458e−3 (5.94e−3)−	7.6552e−3 (4.59e−3)−	**3.8255e–3 (5.72e–4)**
DASCMOP2	3.2971e−2 (2.09e−2)−	1.8071e−2 (1.81e−2)−	1.2905e−2 (1.66e−2)−	**4.4884e–3 (3.33e–4)**
DASCMOP3	6.9617e−2 (5.05e−2)−	3.4954e−2 (1.32e−2) =	3.8755e−2 (2.41e−2)=	3.3482e−2 (9.57e−3)
DASCMOP4	1.2835e–3 (6.42e–4)−	1.1393e−3 (1.75e−5)=	1.2840e−3 (6.47e−4)−	1.1342e−3 (1.06e−5)
DASCMOP5	3.1795e−3 (2.59e−4)−	2.7718e−3 (5.12e−5)=	2.9230e−3 (1.41e−4)−	**2.7794e–3 (5.23e–5)**
DASCMOP6	2.0735e−2 (7.83e−3)−	1.9941e−2 (6.08e−3)−	1.7552e−2 (3.10e−3)−	**1.2952e–2 (4.19e–4)**
DASCMOP7	3.1400e−2 (7.06e−4)−	3.0869e−2 (6.84e−4) =	3.1303e−2 (5.96e−4)−	3.0700e−2 (6.18e−4)
DASCMOP8	3.9262e−2 (9.45e−4)=	3.9530e−2 (6.98e−4)−	3.9679e−2 (8.51e−4)−	3.8942e−2 (7.41e−4)
DASCMOP9	4.0026e−2 (1.52e−3) =	4.0282e−2 (9.61e−4)=	4.0220e−2 (1.04e−3)=	4.0076e−2 (1.17e−3)
Win/lose/tie	0/45/11	0/34/22	0/29/27	

Algorithm selection in the initial stage

To analyze the relationship between the test problems and the algorithms used in the initial stage, we created Table 7 which shows the algorithms used in the initial stage for different test problems. From Table 7, we can see that 34 out of 56 test problems use the IF algorithm in the initial stage, with most of them being type II and type IV problems, except for MW2, MW4, MW14. For type I problems of MW2, MW4 MW14, their UPF and CPF are either both linear or both disconnected and have small feasible regions. This suggests that when the CPF is part of the UPF or the UPF does not contain any part of the CPF, or has the characteristics of MW2, MW4, MW14 for some problems, in the initial optimization stage, the optimal solutions of the unconstrained MOPs may not help much to find the optimal solutions of the CMOPs, and may make it fall into the local optimal. However, the IF algorithm can introduce some potential infeasible solutions in the initial stage to increase the diversity of the solutions, so as to get rid of local optimization, so it is more effective to use the IF algorithm.

There are 22 test problems with the OF algorithm in the initial stage, most of which are type I, type II, and type III problems, except for LIRCMOP7, LIRCMOP8, and DASCMOP3. For type IV problems of LIRCMOP7 and LIRCMOP8, their unconstrained PFs are located in the infeasible region, and their true PFs are located on their constraint boundaries. This also suggests that when the CPF is the same as the UPF or the CPF is part of the UPF or the CPF and UPF partially overlap, in the initial optimization stage, the optimal solutions of the unconstrained MOPs are great helpful to find the optimal solutions of the CMOPs. In this case, the OF algorithm can be used to focus on speeding up the convergence to improve the optimization performance, so the OF algorithm is more effective.

In order to analyze the relationship between different test problems and the selected initial populations, the algorithms used to select P_{inl1} and P_{inl2} respectively for different test problems are given in Table 8. It can be seen from Table 8 that when P_{inl1} is obtained by IF or OF algorithm, if there are DAS-CMOPs with diversity, feasibility and convergence difficulties and partial DOC problems that contain many complex properties in both the decision space and the objective space, P_{inl2} mostly adopts IF algorithm, which indicates that compared with the OF algorithm for such problems, IF algorithm is more efficient to select P_{inl2} by increasing the diversity of solutions. In contrast, P_{inl2} tends to adopt OF algorithm when the CMOPs belongs to type I, type II, type III or partial type IV problems, which indicates that OF algorithm is more effective in select P_{inl2} by focusing on the convergence of solutions compared to the IF algorithm.

To more specifically understand the ratio of each phase, Fig. 10 gives the ratio of the three execution phases of the constrained DTLZ test problems in the optimization process. As can be seen in Fig. 10, for C1-DTLZ1, DC1-DTLZ1, DC1-DTLZ3, DC3-DTLZ1, and DC3-DTLZ3, the ranking of the ratio of execution phases is $OF < IF < PO$, indicating that, except for the final PO stage, the number of optimizations performed by the IF algorithm is higher than the number of optimizations performed by the OF algorithm.

Table 6 Mean (standard deviation) of IGD obtained by MSEFAS and two compared algorithms on five benchmark test suites for 30 independent runs

Problem	MSEFAS (cDPEAPOP)	MSEFAS (CCMOPOP)	MSEFAS
C1-DTLZ1	1.9795e−2 (1.03e−4)−	1.9797e−2 (8.29e−5)−	**1.9677e–2 (6.04e–5)**
C1-DTLZ3	5.3065e−2 (6.38e−4)−	5.2974e−2 (6.37e−4)−	**5.2589e–2 (4.31e–4)**
C2-DTLZ2	4.2237e−2 (4.10e−4) =	4.2329e−2 (4.32e−4) =	4.2352e−2 (3.85e−4)
C3-DTLZ4	9.5124e−2 (1.27e−3)−	9.5256e−2 (1.27e−3)−	**9.2652e–2 (8.49e–4)**
DC1-DTLZ1	1.1367e−2 (6.62e−5) =	1.1368e−2 (6.83e−5) =	1.1364e−2 (7.96e−5)
DC1-DTLZ3	3.3812e−2 (3.78e−4)−	3.3716e−2 (3.19e−4) =	3.3680e−2 (2.97e−4)
DC2-DTLZ1	1.9812e−2 (7.91e−5)−	1.9800e−2 (9.34e−5) =	1.9768e−2 (6.50e−5)
DC2-DTLZ3	5.2288e−2 (3.53e−4)−	5.2264e−2 (3.69e−4)−	**5.1702e–2 (1.58e–4)**
DC3-DTLZ1	6.8247e−3 (4.41e−5)−	6.8389e−3 (4.16e−5)−	**6.7729e–3 (3.16e–5)**
DC3-DTLZ3	1.9979e−2 (2.24e−4) =	1.9958e−2 (1.54e−4) =	1.9952e−2 (2.56e−4)
DOC1	3.5999e+0 (3.84e+0)−	3.2302e+0 (3.66e+0)−	**1.1658e+0 (1.71e+0)**
DOC2	NaN (NaN)=	NaN (NaN)=	NaN (NaN)
DOC3	5.5100e+2 (1.49e+2)−	5.7091e+2 (2.23e+2)−	**3.2916e+2 (1.20e+2)**
DOC4	1.5644e+0 (3.03e+0)−	9.1373e−1 (6.11e−1)−	**2.6558e–1 (1.44e–1)**
DOC5	NaN (NaN)=	NaN (NaN)=	NaN (NaN)
DOC6	1.7635e+0 (1.94e+0)−	1.8771e+0 (1.20e+0)−	**9.7086e–1 (7.49e–1)**
DOC7	6.0736e+0 (2.23e+0)−	5.9970e+0 (2.13e+0)−	**4.3841e+0 (1.56e+0)**
DOC8	8.3043e+1 (7.47e+1)−	8.2155e+1 (6.74e+1)−	**8.4426e+0 (5.15e+0)**
DOC9	1.2848e−1 (9.66e−2)−	1.7533e−1 (1.15e−1)−	**6.7183e–2 (5.13e–2)**
MW1	1.5975e−3 (1.10e−5) =	1.6000e−3 (8.69e-6)−	1.5944e−3 (7.25e-6)
MW2	1.4700e−2 (6.72e−3) =	1.4052e−2 (6.61e−3) =	1.4803e−2 (5.74e−3)
MW3	4.5282e−3 (1.44e−4)−	4.5293e−3 (1.82e−4)−	**4.3739e–3 (9.74e–5)**
MW4	3.9964e−2 (2.86e−4)−	3.9844e−2 (2.20e−4)=	3.9682e−2 (1.72e−4)
MW5	4.7428e−4 (7.05e−4)−	5.2268e−4 (6.86e−4)−	**2.2992e–4 (6.82e–5)**
MW6	1.0725e−2 (6.57e−3)−	1.3630e−2 (9.69e−3)−	**6.9134e–3 (4.48e–3)**
MW7	4.2513e−3 (2.60e−4)−	4.1848e−3 (2.33e−4)−	**4.0353e–3 (1.36e–4)**
MW8	4.4490e−2 (2.12e−3)−	4.4729e−2 (2.53e−3)−	**4.2648e–2 (4.03e–4)**
MW9	4.4393e−3 (7.72e−4)−	2.7828e−2 (1.29e−1)−	**4.1490e–3 (4.61e–5)**
MW10	2.0194e−2 (1.44e−2)−	1.7510e−2 (1.54e−2)−	**7.1128e–3 (3.53e–3)**
MW11	5.9296e−3 (2.17e−4)−	5.9126e−3 (1.36e−4)−	**5.5924e–3 (4.87e–4)**
MW12	4.6754e−3 (8.36e−5)−	4.7287e−3 (1.06e−4)−	**4.5676e–3 (4.47e–5)**
MW13	3.5280e−2 (2.47e−2)−	3.8134e−2 (2.89e−2)−	**1.9538e–2 (8.14e–3)**
MW14	9.5833e−2 (1.36e−3)−	9.5809e−2 (1.71e−3)−	**9.4550e–2 (1.06e–3)**
LIRCMOP1	1.3730e−1 (5.68e−2) =	1.3907e−1 (6.60e−2) =	1.1579e−1 (4.27e−2)
LIRCMOP2	6.7906e−2 (3.01e−2) =	7.3161e−2 (2.35e−2) =	6.4338e−2 (2.78e−2)
LIRCMOP3	1.6762e−1 (5.99e−2)−	1.4924e−1 (7.78e−2) =	1.3362e−1 (6.42e−2)
LIRCMOP4	1.6390e−1 (6.39e−2)−	1.7067e−1 (6.23e−2)−	**1.3253e–1 (4.16e–2)**
LIRCMOP5	7.4450e−3 (2.40e−3)−	7.0167e−3 (4.92e−4)−	**6.7290e–3 (3.93e–4)**
LIRCMOP6	6.6734e−3 (5.77e−4) =	6.8325e−3 (1.60e−3)−	6.4305e−3 (2.46e−4)
LIRCMOP7	7.9197e−3 (8.17e−4)−	7.8387e−3 (8.02e−4)−	**7.1214e–3 (7.47e–5)**
LIRCMOP8	7.3389e−3 (2.12e−4)−	7.3808e−3 (2.50e−4)−	**7.0374e−3 (1.49e−4)**

Table 6 continued

Problem	MSEFAS (cDPEAPOP)	MSEFAS (CCMOPOP)	MSEFAS
LIRCMOP9	5.3045e−3 (1.77e−3)−	8.3321e−3 (1.34e−2)−	**4.6463e–3 (1.25e–3)**
LIRCMOP10	5.0208e−3 (2.17e−4)−	4.9957e−3 (1.83e−4)−	**4.6248e–3 (1.06e–4)**
LIRCMOP11	2.3851e−3 (5.30e−5) =	2.3842e−3 (4.28e−5) =	2.3796e−3 (5.29e−5)
LIRCMOP12	3.0744e−3 (9.50e−5) =	3.0867e−3 (1.12e−4) =	3.0482e−3 (1.31e−4)
LIRCMOP13	9.0935e−2 (7.42e−4) =	9.0634e−2 (7.94e−4) =	9.0649e−2 (7.98e−4)
LIRCMOP14	9.5416e−2 (9.08e−4) =	9.5479e−2 (9.46e−4)=	9.4857e−2 (6.11e−4)
DASCMOP1	9.6307e−3 (5.30e−3)−	7.7671e−3 (4.09e−3)−	**3.8255e–3 (5.72e–4)**
DASCMOP2	1.9881e−2 (2.30e−2)−	2.2934e−2 (2.37e−2)−	**4.4884e–3 (3.33e–4)**
DASCMOP3	4.3992e−2 (2.84e−2)−	4.4319e−2 (3.35e−2)−	**3.3482e–2 (9.57e–3)**
DASCMOP4	1.1436e−3 (1.61e−5) =	1.1476e−3 (2.22e−5) =	1.1342e−3 (1.06e−5)
DASCMOP5	3.0494e−3 (2.13e−4)−	2.9892e−3 (1.98e−4)−	**2.7794e–3 (5.23e–5)**
DASCMOP6	2.0012e−2 (4.31e−3)−	1.9896e−2 (4.51e−3)−	**1.2952e–2 (4.19e–4)**
DASCMOP7	3.1100e−2 (8.16e−4)−	3.1144e−2 (6.79e−4)−	**3.0700e–2 (6.18e–4)**
DASCMOP8	3.9478e−2 (1.05e−3)−	3.9243e−2 (9.66e−4)−	3.8942e−2 (7.41e−4)
DASCMOP9	4.0218e−2 (1.09e−3) =	4.0389e−2 (1.10e−3) =	4.0076e−2 (1.17e−3)
Win/lose/tie	0/40/16	0/38/18	

Table 7 The algorithms used in the initial stage for different test problems

The algorithms used in the initial stage	Problem
IF algorithm	Type I: MW2, MW4, MW14
	Type II: C2-DTLZ2, DC2-DTLZ1, DC2-DTLZ3, MW1, MW5, MW6, MW8
	Type IV: C3-DTLZ4, DC3-DTLZ1, DC3-DTLZ3, MW9, MW11, MW12, LIRCMOP1, LIRCMOP2, LIRCMOP3
	LIRCMOP4, DASCMOP1, DASCMOP2, DASCMOP4, DASCMOP5, DASCMOP6, DASCMOP7, DASCMOP8
	DASCMOP9
	DOC2, DOC3, DOC5, DOC6, DOC8, DOC9
OF algorithm	Type I: C1-DTLZ1, C1-DTLZ3, DC1-DTLZ1, DC1-DTLZ3, LIRCMOP5, LIRCMOP6, LIRCMOP13, LIRCMOP14
	Type II: LIRCMOP9, LIRCMOP10
	Type III: MW3, MW7, MW10, MW13, LIRCMOP11, LIRCMOP12
	Type IV: LIRCMOP7, LIRCMOP8, DASCMOP3
	DOC1, DOC4, DOC7

This suggests that the introducing infeasible solutions stage has a greater effect on the overall optimization compared to the optimization stage based on offspring individuals. For C1-DTLZ3, C2-DTLZ2, C3-DTLZ4, DC2-DTLZ1, and DC2-DTLZ3, the ranking of the ratio of execution phases is $IF < OF < PO$, which indicates that the number of optimizations performed by the OF algorithm is higher than the number of optimizations performed by the IF algorithm. This suggests that the optimization stage based on offspring individuals has a greater effect on the overall optimization compared to the introducing infeasible solutions optimization stage.

Flexibility analysis

To further confirm the flexibility of the proposed MSEFAS framework, we embedded the Ship algorithm [35] into the IF stage and denoted the resulting algorithm as MSEFAS(Ship). The results of mean and standard deviation of the IGD obtained by MSEFAS(Ship) and five comparison algorithms on five benchmark test suites for 30 independent runs are presented in Table 9. As can be seen, MSEFAS(Ship) significantly outperformed MOEADDAE, CMOEA-MS, CCMO, c-DPEA, and EMCMO on 46, 47, 39, 42, and 32 problems, respectively, and was comparable to them on 7, 7, 16, 12, and 23 problems. These results demonstrate the flexibility

Table 8 The algorithms used to select P_{inl1} and P_{inl2} for different test problems

Initial optimization populations	Problem
P_{inl1}(IF), P_{inl2}(IF)	Type IV: DASCMOP2, DASCMOP4, DASCMOP5, DASCMOP6, DASCMOP7, DASCMOP8, DASCMOP9
	DOC2, DOC5, DOC9.
P_{inl1}(IF), P_{inl2}(OF)	Type I: MW2, MW4, MW14
	Type II: C2-DTLZ2, DC2-DTLZ1, DC2-DTLZ3, MW1, MW5, MW6, MW8
	Type IV: C3-DTLZ4, DC3-DTLZ1, DC3-DTLZ3, MW9, MW11, MW12, LIRCMOP1, LIRCMOP2, LIRCMOP3
	LIRCMOP4, DASCMOP1
	DOC3, DOC6, DOC8
P_{inl1}(OF), P_{inl2}(IF)	Type IV: DASCMOP3
	DOC1, DOC4
P_{inl1}(OF), P_{inl2}(OF)	Type I: C1-DTLZ1, C1-DTLZ3, DC1-DTLZ1, DC1-DTLZ3, LIRCMOP5, LIRCMOP6, LIRCMOP13, LIRCMOP14
	Type II: LIRCMOP9, LIRCMOP10
	Type III: MW3, MW7, MW10, MW13, LIRCMOP11, LIRCMOP12
	Type IV: LIRCMOP7, LIRCMOP8
	DOC7

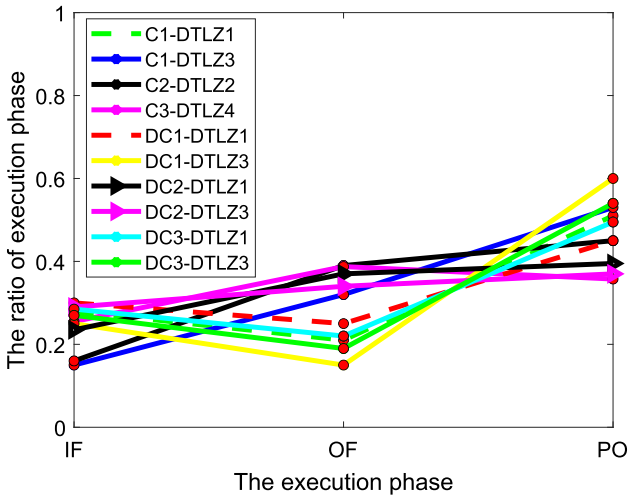


Fig. 10 The ratio of execution phase

of the MSEFAS framework in allowing the incorporation of different algorithms into its different stages.

Results on real-world CMOPs

To verify the effectiveness of the proposed MSEFAS for real-world applications, we tested the proposed MSEFAS together with TiGE2, CTAEA, MOEADDAE, ToP, DSPCMDE and MSCMO on real-world problems of RMWCMOP. RWCMOP1 represents the pressure vessel design problem [47], RWCMOP2 is a vibrating platform design problem [48], RWCMOP3 is the two bar truss design problem [49], RWCMOP4 is the welded beam design problem [50], RWCMOP5 represents the disc brake design problem [51], RWCMOP6 is the speed reducer design problem [52], RWCMOP7 represents the gear train design problem [53], RWCMOP8 represents the gar side impact design problem [54], and RWCMOP9 represents the four bar plane truss problem [55]. For RWCMOP2, $V = 5$, $M = 2$. For RWCMOP3, $V = 3$, $M = 2$. For RWCMOP6, $V = 7$, $M = 2$. For RWCMOP8, $V = 7$, $M = 3$. For the remaining five problems, $V = 4$, $M = 2$. The initial population of all problems is 100. In addition to $MaxFEs = 26250$ for RWCMOP8, for other problems, $MaxFEs = 20000$. Table 10 lists the HV results obtained by MSEFAS and six compared algorithms on nine RWMOP test problems over 30 independent runs.

As we can see from the comparative results in Table 10, MSEFAS is superior to six algorithms under comparison on nine RWMOPs, confirming the effectiveness of MSEFAS on practical problems.

Conclusion

This paper has presented an adaptive selection multi-stage framework, MSEFAS, for effectively handling constrained multi-objective optimization problems (CMOPs). Generally, MSEFAS includes three tailored optimization stages and adaptively assigns the execution order and initial optimized populations to different algorithms based on their effectiveness in selecting promising individuals. Specifically, one stage optimizes competitive infeasible solutions with good objective values and low constraint violations, enabling the transition between diversity and feasibility; another stage

Table 9 Mean (standard deviation) of IGD obtained by MSEFAS and five compared algorithms on five benchmark test suites for 30 independent runs

Problem	MOEADDAE	CMOEA-MS	CCMO	c-DPEA	EMCMO	MSEFAS (ship)
C1-DTLZ1	2.1068e−2 (1.32e−4)−	2.0687e−2 (2.43e−4)−	2.0072e−2 (1.30e−4)−	2.0046e−2 (1.49e−4)−	1.9750e−2 (8.89e−5) =	1.9763e−2 (7.31e−5)
C1-DTLZ3	6.5273e−2 (4.80e−3)−	5.3225e−2 (5.70e−4)−	5.3523e−2 (5.98e−4)−	5.3563e−2 (6.81e−4)−	5.2894e−2 (5.95e−4) =	5.2789e−2 (4.88e−4)
C2-DTLZ2	4.4195e−2 (4.58e−4)−	4.2392e−2 (4.41e−4) =	4.2874e−2 (4.36e−4)−	4.2809e−2 (6.32e−4)−	4.2360e−2 (4.80e−4) =	4.2322e−2 (4.17e−4)
C3-DTLZ4	9.4998e−1 (3.62e−1)−	6.5057e−1 (5.11e−2)−	9.5283e−2 (1.25e−3)=	9.7113e−2 (1.37e−3)−	1.1985e−1 (1.36e−1)−	9.4593e−2 (1.15e−3)
DC1-DTLZ1	1.2098e−2 (9.86e−5)−	1.8218e−2 (6.14e−3)−	1.1373e−2 (8.36e−5) =	1.1386e−2 (6.60e−5) =	1.1388e−2 (5.72e−5) =	1.1350e−2 (6.22e−5)
DC1-DTLZ3	3.7260e−2 (1.37e−3)−	3.3757e−2 (4.52e−4)−	3.3728e−2 (2.56e−4)−	3.3889e−2 (3.53e−4)−	3.3723e−2 (2.64e−4)−	**3.3526e–2 (3.24e–4)**
DC2-DTLZ1	2.1203e−2 (1.27e−4)−	2.0667e−2 (2.02e−4)−	2.0163e−2 (1.50e−4)−	2.0129e−2 (1.77e−4)−	1.9805e−2 (9.46e−5) =	1.9853e−2 (9.49e−5)
DC2-DTLZ3	3.7691e−1 (2.48e−1)−	1.7132e−1 (2.18e−1)−	5.3043e−2 (4.52e−4)−	5.2854e−2 (4.11e−4)−	5.2255e−2 (3.55e−4)−	**5.1866e–2 (1.21e–4)**
DC3-DTLZ1	8.7975e−3 (7.98e−3)−	3.3577e−2 (6.05e−3)−	6.8156e−3 (5.05e−5)−	6.8487e−3 (5.32e−5)−	6.8166e−3 (5.43e−5)−	**6.7784e–3 (2.93e–5)**
DC3-DTLZ3	2.1293e−2 (5.47e−4)−	8.2046e−2 (1.93e−1)−	**1.9912e–2 (2.25e–4) =**	1.9974e−2 (2.64e−4) =	1.9912e−2 (2.08e−4) =	1.9927e−2 (2.20e−4)
DOC1	**1.4278e−1 (7.67e−2) +**	3.6553e+0 (3.39e+0)−	5.9111e+0 (5.83e+0)−	4.6963e–1 (4.43e–1) +	4.0679e+0 (4.22e+0)−	1.0944e+0 (1.29e+0)
DOC2	NaN (NaN)=	NaN (NaN)=	NaN (NaN)=	NaN (NaN)=	NaN (NaN)=	NaN (NaN)
DOC3	NaN (NaN)=	**6.9409e+2 (3.22e+2)+**	**7.4993e+2 (2.22e+2)+**	**7.3037e+2 (2.75e+2)+**	**6.3359e+2 (2.41e+2)+**	NaN (NaN)
DOC4	1.5491e+0 (7.12e−1)−	8.5749e−1 (9.65e−1)−	1.3905e+0 (1.42e+0)−	5.0469e−1 (2.33e−1)−	1.1965e+0 (1.11e+0)−	**1.4847e–1 (1.23e–1)**
DOC5	NaN (NaN)=	**8.8563e+1 (3.27e+1)+**	NaN (NaN)=	NaN (NaN)=	NaN (NaN)=	NaN (NaN)
DOC6	2.4702e+0 (8.56e−1)−	1.7062e+0 (1.32e+0)−	3.2490e+0 (3.93e+0)−	3.5357e+0 (4.65e+0)−	2.3713e+0 (2.34e+0)−	**8.2650e–1 (5.94e–1)**
DOC7	2.7304e+0 (1.54e+0) =	4.7114e+0 (2.18e+0)−	6.2338e+0 (2.87e+0)−	7.1658e+0 (2.17e+0)−	5.3369e+0 (1.83e+0)−	3.5095e+0 (4.98e−1)
DOC8	1.2715e+2 (7.67e+1)−	1.4655e+2 (7.20e+1)−	6.8208e+1 (5.38e+1)−	9.9605e+1 (6.92e+1)−	8.7843e+1 (5.44e+1)−	**6.3398e+0 (3.55e+0)**
DOC9	1.5982e−1 (1.36e−1)−	1.3613e−1 (2.30e−1)−	1.7550e−1 (9.51e−2)−	1.9875e−1 (1.06e−1)−	1.5267e−1 (1.19e−1)−	**3.8861e–2 (1.05e–3)**
MW1	1.8539e−3 (4.41e−5)−	1.6719e−3 (1.73e−5)−	1.6150e−3 (1.04e−5)−	1.6170e−3 (1.27e−5)−	1.5982e−3 (1.10e−5) =	1.5948e−3 (4.96e-6)
MW2	4.7158e−2 (2.80e−2)−	1.8916e−2 (7.63e−3)−	2.0745e−2 (8.73e−3)−	1.9261e−2 (1.77e−2)−	1.9302e−2 (9.57e−3)−	**6.2115e–3 (4.26e–3)**
MW3	4.9975e−3 (1.58e−4)−	5.0266e−3 (2.69e−4)−	4.6427e−3 (1.68e−4)−	4.7412e−3 (1.76e−4)−	4.4749e−3 (1.50e−4)−	**4.3598e–3 (9.49e–5)**
MW4	4.3044e−2 (3.14e−4)−	4.1744e−2 (5.33e−4)−	4.0734e−2 (3.35e−4)−	4.0631e−2 (3.30e−4)−	3.9748e−2 (2.72e−4) =	**3.9609e–2 (6.68e–5)**
MW5	**7.8361e–5 (2.92e–5) +**	1.7830e−2 (1.83e−2)−	5.6977e−4 (7.09e−4)−	3.5718e−4 (1.86e−4)−	1.0264e−3 (1.03e−3)−	2.4070e−4 (6.35e−5)
MW6	1.0806e−1 (1.55e−1)−	3.7107e−2 (8.86e−2)−	1.9412e−2 (1.41e−2)−	1.2383e−2 (8.07e−3)−	3.8373e−2 (8.24e−2)−	**2.9031e–3 (9.60e–4)**
MW7	4.4784e−3 (1.37e−4)−	1.4865e−2 (2.02e−2)−	4.1135e−3 (2.32e−4) =	3.9606e−3 (1.27e−4) =	4.0428e−3 (1.80e−4) =	4.0317e−3 (1.32e−4)
MW8	8.1825e−2 (5.04e−2)−	4.8347e−2 (4.73e−3)−	4.9073e−2 (1.43e−2)−	4.3995e−2 (2.47e−3)−	4.7121e−2 (5.73e−3)−	**4.2877e–2 (6.77e–4)**
MW9	4.4306e−3 (1.04e−4)−	4.0813e−2 (1.41e−1)−	4.1716e−3 (9.36e−5) =	4.2585e−3 (1.18e−4)−	4.1764e−3 (7.43e−5) =	4.1507e−3 (6.48e−5)
MW10	1.4026e−1 (1.79e−1)−	5.4826e−2 (4.89e−2)−	4.6329e−2 (4.21e−2)−	2.0730e−2 (1.56e−2)−	4.2607e−2 (4.59e−2)−	**4.5906e–3 (2.73e–3)**
MW11	6.3028e−3 (2.54e−4)−	2.8984e−2 (1.26e−1)−	5.8439e−3 (1.21e−4)=	6.0326e−3 (1.61e−4)−	5.8887e−3 (1.07e−4)=	5.7604e−3 (1.35e−4)
MW12	5.2534e−3 (1.79e−4)−	5.1715e−3 (1.65e−4)−	4.6549e−3 (7.02e−5)−	4.7725e−3 (1.34e−4)−	4.6671e−3 (1.37e−4)−	**4.5975e–3 (9.25e–5)**
MW13	1.2282e−1 (4.53e−2)−	8.0699e−2 (4.33e−2)−	6.1210e−2 (3.56e−2)−	3.4672e−2 (2.56e−2)−	8.1790e−2 (3.95e−2)−	**1.3687e–2 (5.53e–3)**
MW14	1.0290e−1 (2.25e−2)−	1.2461e−1 (4.10e−3)−	9.7085e−2 (1.69e−3)−	9.6933e−2 (1.71e−3)−	9.5525e−2 (1.45e−3)−	**9.4369e–2 (8.14e–4)**
LIRCMOP1	2.7275e−1 (2.06e−1)−	3.8373e−1 (1.45e−1)−	1.6786e−1 (7.57e−2)−	1.3937e−1 (9.18e−2)−	2.0377e−1 (1.05e−1)−	**8.8215e–2 (2.25e–2)**
LIRCMOP2	9.1070e−2 (4.70e−2)−	3.0712e−1 (9.01e−2)−	7.5101e−2 (3.02e−2)−	8.9558e−2 (5.02e−2)−	7.8293e−2 (3.21e−2)−	**5.5357e–2 (1.07e–2)**

Table 9 continued

Problem	MOEADDAE	CMOEA-MS	CCMO	c-DPEA	EMCMO	MSEFAS (ship)
LIRCMOP3	1.2565e−1 (1.07e−1)=	3.5161e−1 (1.03e−1)−	1.5359e−1 (6.80e−2)−	1.5406e−1 (1.72e−1)−	2.0427e−1 (1.10e−1)−	**1.1760e–1 (4.25e–2)**
LIRCMOP4	7.2807e−2 (8.79e−2)=	3.0801e−1 (7.29e−2)−	1.6133e−1 (8.38e−2)−	1.5265e−1 (1.23e−1)−	1.7493e−1 (7.48e−2)−	1.0908e−1 (3.63e−2)
LIRCMOP5	5.0682e−2 (2.12e−2)−	1.9001e−2 (2.03e−2)−	7.0367e−3 (2.02e−3)=	8.7005e−3 (4.30e−3)−	7.3017e−3 (2.19e−3) =	7.2141e−3 (5.23e−4)
LIRCMOP6	6.3432e−2 (3.64e−2)−	1.7370e−2 (3.47e−2)−	6.0541e−3 (3.02e−4)=	7.9594e−3 (4.88e−3)−	6.1677e−3 (4.20e−4)=	6.7163e−3 (3.98e−4)
LIRCMOP7	3.6250e−2 (1.17e−2)−	7.9432e−3 (1.84e−3)−	7.8575e−3 (1.02e−3)−	7.5719e−3 (7.00e−4)−	7.7270e−3 (5.88e−4)−	**7.3507e–3 (2.04e–4)**
LIRCMOP8	3.1317e−2 (1.54e−2)−	9.1023e−3 (1.18e−2)−	7.2343e−3 (2.68e−4)−	7.2160e−3 (5.16e−4)−	7.2244e−3 (2.11e−4)−	**7.0123e–3 (1.25e–4)**
LIRCMOP9	**3.2418e−3 (1.03e−4) +**	2.4700e−1 (1.55e−1)−	5.1032e−3 (1.66e−3)−	5.3977e−2 (3.46e−2)−	5.6304e−3 (1.95e−3)−	3.8193e−3 (5.50e−4)
LIRCMOP10	4.7575e−3 (1.31e−4)=	7.3710e−2 (4.52e−2)−	4.6943e−3 (1.79e−4)=	4.8127e−3 (1.39e−4)=	4.7352e−3 (1.59e−4)=	4.7649e−3 (9.39e−5)
LIRCMOP11	2.5397e−3 (2.05e−4)−	7.7422e−2 (6.37e−2)−	2.3976e−3 (6.68e−5)=	2.3961e−3 (6.56e−5) =	2.3897e−3 (5.60e−5)=	2.3913e−3 (1.98e−5)
LIRCMOP12	3.2096e−3 (1.38e−4)−	6.5993e−2 (1.11e−1)−	3.0504e−3 (7.38e−5)=	3.0757e−3 (9.01e−5)=	3.0527e−3 (8.88e−5)=	3.0959e−3 (1.28e−4)
LIRCMOP13	9.7280e−2 (9.43e−4)−	9.2639e−2 (7.98e−4)−	9.3733e−2 (1.16e−3)−	9.3270e−2 (1.00e−3)−	9.1068e−2 (6.11e−4) =	9.0711e−2 (7.18e−4)
LIRCMOP14	9.9484e−2 (7.22e−4)−	9.5422e−2 (9.23e−4)=	9.6393e−2 (9.86e−4)−	9.5440e−2 (9.07e−4)=	9.5785e−2 (8.79e−4)−	**9.4806e–2 (1.97e–4)**
DASCMOP1	8.0037e−2 (1.55e−1)−	7.7480e−2 (1.32e−1)−	7.1166e−3 (5.05e−3)−	5.5960e−3 (4.28e−3)−	7.7094e−3 (4.28e−3)−	**3.9890e–3 (3.14e–4)**
DASCMOP2	5.0545e−2 (4.36e−2)−	7.9459e−3 (4.04e−3)−	1.1475e−2 (1.89e−2)−	4.5916e−3 (1.22e−3)−	1.0609e−2 (1.04e−2)−	**4.2407e–3 (1.36e–4)**
DASCMOP3	8.8202e−2 (5.90e−2)−	1.2694e−1 (4.70e−2)−	5.3725e−2 (3.72e−2)−	2.4785e−2 (1.12e−2)=	4.8402e−2 (3.43e−2)−	2.8725e−2 (1.68e−2)
DASCMOP4	1.3136e−3 (4.32e−5)−	1.1434e−3 (1.18e−5)=	1.1391e−3 (1.16e−5) =	1.2481e−3 (5.26e−4)−	1.1380e−3 (1.32e−5) =	1.1359e−3 (8.83e-6)
DASCMOP5	3.0540e−3 (1.06e−4)−	2.6777e−3 (6.43e−5) =	2.8004e−3 (7.73e−5)−	3.0658e−3 (2.12e−4)−	2.7462e−3 (4.79e−5)=	2.7682e−3 (7.07e−5)
DASCMOP6	2.4387e−2 (1.83e−2)−	2.8828e−2 (3.31e−2)−	1.9236e−2 (4.86e−3)−	1.8408e−2 (2.67e−3)−	1.8868e−2 (4.47e−3)−	**1.2606e−2 (9.27e−5)**
DASCMOP7	3.5659e−2 (1.56e−3)−	3.0909e−2 (4.89e−4) =	3.1159e−2 (5.24e−4)=	3.0853e−2 (6.30e−4) =	3.0811e−2 (5.49e−4) =	3.0921e−2 (9.75e−4)
DASCMOP8	4.6334e−2 (2.15e−3)−	3.9401e−2 (8.85e−4)−	3.9539e−2 (7.29e−4)−	3.9053e−2 (6.67e−4) =	3.9707e−2 (8.35e−4)−	3.8925e−2 (9.04e−4)
DASCMOP9	6.9856e−2 (1.39e−2)−	4.0713e−2 (1.09e−3) =	4.0416e−2 (1.08e−3) =	4.1726e−2 (1.07e−3)−	4.0209e−2 (9.63e−4) =	4.0167e−2 (1.12e−3)
Win/lose/tie	3/46/7	2/47/7	1/39/16	2/42/12	1/32/23	

Table 10 Mean (standard deviation) of HV obtained by MSEFAS and six compared algorithms on nine RWMOP test problems for 30 independent runs

Problem	TiGE2	CTAEA	MOEADDAE	ToP	DSPCMDE	MSCMO	MSEFAS
RWMOP1	5.2022e−1(1.92e−2)−	6.0539e−1(8.84e−4)≡	5.5825e-(2.01e−2)−	6.0778e−1(4.47e−4)≡	**6.5750e−1 (1.36e−1)+**	NaN(NaN)−	6.0517e−1 (3.54e−4)
RWMOP2	1.0928e−1 (1.03e−1)−	1.5941e−1 (1.72e−1)−	5.5761e−3 (2.67e−2)−	3.5488e−1 (1.02e−1)≡	2.4136e−1 (1.58e−1)−	NaN(NaN)−	3.4970e−1 (5.79e−3)
RWMOP3	7.7910e−1 (9.15e−2)−	8.6867e−1 (2.14e−2)−	9.0128e−1 (1.36e−4)≡	7.2210e−1 (2.53e−1)−	9.0168e−1 (2.25e−4)≡	2.9831e−1 (2.36e−1)−	9.0011e−1 (4.71e−4)
RWMOP4	4.1664e−1 (1.61e−1)−	8.5192e−1 (1.11e−2)≡	8.5492e−1 (2.42e−3)≡	8.6209e−1 (6.61e−4)≡	7.9205e−1 (2.22e−2)−	6.0781e−2 (1.30e−1)−	8.5704e−1 (2.45e−3)
RWMOP5	4.0080e−1 (8.06e−3)−	4.2806e−1 (5.03e−3)−	4.2929e−1 (3.05e−3)−	4.3456e−1 (1.21e−4)≡	2.7691e−1 (6.33e−5)−	3.9472e−1 (8.62e−3)−	4.3407e−1 (3.02e−4)
RWMOP6	2.7218e−1 (1.98e−3)≡	2.1878e−1 (5.11e−2)−	2.7659e−1 (2.54e−4)≡	2.7668e−1 (8.03e−4)≡	2.7435e−1 (9.91e−3)≡	NaN(NaN)−	2.7676e−1 (1.17e−4)
RWMOP7	4.7085e−1(1.03e−2)−	4.8206e−1(4.10e−3)≡	4.8367e−1(1.72e−4)≡	4.8438e−1(8.27e−5)≡	4.8434e−1(6.12e−5)≡	4.8122e−1(1.52e−3)≡	4.8429e−1(6.07e−5)
RWMOP8	2.0487e−2(3.03e−4)−	2.6051e−2(5.88e−5)≡	2.5736e−2(1.05e−4)−	2.5813e−2(1.06e−4)−	2.5859e−2(9.99e−5)−	2.4502e−2(5.64e−4)−	2.6038e−2(5.13e−5)
RWMOP9	3.9537e−1(3.83e−3)−	4.0858e−1(8.88e−4)≡	3.8539e−1(5.98e−3)−	4.0933e−1(1.45e−4)≡	4.0933e−1(1.70e−4)≡	4.0930e−1(2.80e−4)≡	4.0944e−1(1.58e−4)
Win/lose/tie	0/8/1	0/4/5	0/5/4	0/2/7	1/4/4	0/7/2	

is introduced to improve convergence or diversity performance on CMOPs with small feasible regions; the final stage considers the relationship between the constrained Pareto optimal front (CPF) and the unconstrained Pareto optimal front (UPF). In comparison with five state-of-the-art constrained multi-objective evolutionary algorithms on a variety of benchmark suites, we have demonstrated the effectiveness of the proposed framework for handling a wide range of CMOPs.

Although MSEFAS has demonstrated promising performance, there is still potential for future improvements. For different optimization problems, the quality of offspring during the optimization process directly affects the optimization results. Possible future work could explore the adaptive assignment of different evolutionary operators to various optimization problems based on their success rates in generating offspring individuals, or directly exploring more efficient evolutionary operators. Another possibility is to use machine learning to train deep network models for predicting potential offspring, thereby partially replacing the offspring individuals generated through evolutionary processes. Furthermore, based on the relationship between UPFs and CPFs, UPFs can significantly aid in finding CPFs, so we can learn knowledge from unconstrained MOPs and apply it to CMOPs. Considering that the dimensions of these two types of problems differ after accounting for constraints, another intriguing approach could involve using heterogeneous transfer learning methods to provide useful knowledge for transitioning across different optimization stages. This would further enhance the adaptability and effectiveness of the proposed framework in handling various CMOPs.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s40747-024-01542-9.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Kim HS, Kang JW (2012) Semi-active fuzzy control of a wind-excited tall building using multi-objective genetic algorithm. Eng Struct 41:242–257

2. Saravanan R, Ramabalan S, Ebenezer NGR, Dharmaraja C (2009) Evolutionary multi criteria design optimization of robot grippers. Appl Soft Comput 9(1):159–172
3. Omkar S, Senthilnath J, Khandelwal R, Naik GN, Gopalakrishnan S (2011) Artificial bee colony (ABC) for multi-objective design optimization of composite structures. Appl Soft Comput 11(1):489–499
4. Lee LH, Tan KC, Ou K, Chew YH (2003) Vehicle capacity planning system: a case study on vehicle routing problem with time windows. IEEE Trans Syst Man Cybern Part A Syst Hum 33(2):169–178
5. Amini F, Ghaderi P (2013) Hybridization of harmony search and ant colony optimization for optimal locating of structural dampers. Appl Soft Comput 13(5):2272–2280
6. Wang J, Ren W, Zhang Z, Huang H, Zhou Y (2020) A hybrid multiobjective memetic algorithm for multiobjective periodic vehicle routing problem with time windows. IEEE Trans Syst Man Cybern Syst 50(11):4732–4745. https://doi.org/10.1109/TSMC.2018.2861879
7. Tanabe R, Oyama A (2017) A note on constrained multi-objective optimization benchmark problems. In: 2017 IEEE congress on evolutionary computation (CEC), pp 1127–1134. https://doi.org/10.1109/CEC.2017.7969433
8. Ma Z, Wang Y (2019) Evolutionary constrained multiobjective optimization: test suite construction and performance comparisons. IEEE Trans Evolut Comput 23(6):972–986. https://doi.org/10.1109/TEVC.2019.2896967
9. Tian Y, Zhang Y, Su Y, Zhang X, Tan KC, Jin Y (2022) Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization. IEEE Trans Cybern 52(9):9559–9572. https://doi.org/10.1109/TCYB.2020.3021138
10. Ma Z, Wang Y, Song W (2021) A new fitness function with two rankings for evolutionary constrained multiobjective optimization. IEEE Trans Syst Man Cybern Syst 51(8):5005–5016. https://doi.org/10.1109/TSMC.2019.2943973
11. Zhou Y, Zhu M, Wang J, Zhang Z, Xiang Y, Zhang J (2018) Tri-goal evolution framework for constrained many-objective optimization. IEEE Trans Syst Man Cybern Syst 50(8):3086–3099
12. Fan Z, Wang Z, Li W, Yuan Y, You Y, Yang Z, Sun F, Ruan J (2020) Push and pull search embedded in an m2m framework for solving constrained multi-objective optimization problems. Swarm Evolut Comput 54:100651
13. Liu ZZ, Wang Y (2019) Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces. IEEE Trans Evolut Comput 23(5):870–884
14. Zhu Q, Zhang Q, Lin Q (2020) A constrained multiobjective evolutionary algorithm with detect-and-escape strategy. IEEE Trans Evolut Comput 24(5):938–947. https://doi.org/10.1109/TEVC.2020.2981949
15. Ma H, Wei H, Tian Y, Cheng R, Zhang X (2021) A multi-stage evolutionary algorithm for multi-objective optimization with complex constraints. Inf Sci 560:68–91
16. Yu K, Liang J, Qu B, Luo Y, Yue C (2022) Dynamic selection preference-assisted constrained multiobjective differential evolution. IEEE Trans Syst Man Cybern Syst 52(5):2954–2965. https://doi.org/10.1109/TSMC.2021.3061698
17. Yu K, Liang J, Qu B, Yue C (2021) Purpose-directed two-phase multiobjective differential evolution for constrained multiobjective optimization. Swarm Evolut Comput 60:100799
18. Li K, Chen R, Fu G, Yao X (2019) Two-archive evolutionary algorithm for constrained multiobjective optimization. IEEE Trans Evolut Comput 23(2):303–315. https://doi.org/10.1109/TEVC.2018.2855411
19. Wang J, Liang G, Zhang J (2019) Cooperative differential evolution framework for constrained multiobjective optimization. IEEE Trans Cybern 49(6):2060–2072. https://doi.org/10.1109/TCYB.2018.2819208
20. Tian Y, Zhang T, Xiao J, Zhang X, Jin Y (2021) A coevolutionary framework for constrained multiobjective optimization problems. IEEE Trans Evolut Comput 25(1):102–116. https://doi.org/10.1109/TEVC.2020.3004012
21. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evolut Comput 6(2):182–197
22. Liu HL, Chen L, Deb K, Goodman ED (2016) Investigating the effect of imbalance between convergence and diversity in evolutionary multiobjective algorithms. IEEE Trans Evolut Comput 21(3):408–425
23. Ming M, Trivedi A, Wang R, Srinivasan D, Zhang T (2021) A dual-population-based evolutionary algorithm for constrained multiobjective optimization. IEEE Trans Evolut Comput 25(4):739–753. https://doi.org/10.1109/TEVC.2021.3066301
24. Fan Z, Li W, Cai X, Li H, Wei C, Zhang Q, Deb K, Goodman E (2019) Push and pull search for solving constrained multi-objective optimization problems. Swarm Evolut Comput 44:665–679
25. Qiao K, Yu K, Qu B, Liang J, Song H, Yue C, Lin H, Tan KC (2022) Dynamic auxiliary task-based evolutionary multitasking for constrained multi-objective optimization. IEEE Trans Evolut Comput. https://doi.org/10.1109/TEVC.2022.3175065
26. Ming M, Wang R, Ishibuchi H, Zhang T (2021) A novel dual-stage dual-population evolutionary algorithm for constrained multi-objective optimization. IEEE Trans Evolut Comput. https://doi.org/10.1109/TEVC.2021.3131124
27. Wang BC, Li HX, Zhang Q, Wang Y (2021) Decomposition-based multiobjective optimization for constrained evolutionary optimization. IEEE Trans Syst Man Cybern Syst 51(1):574–587. https://doi.org/10.1109/TSMC.2018.2876335
28. Wang J, Li Y, Zhang Q, Zhang Z, Gao S (2022) Cooperative multiobjective evolutionary algorithm with propulsive population for constrained multiobjective optimization. IEEE Trans Syst Man Cybern Syst 52(6):3476–3491. https://doi.org/10.1109/TSMC.2021.3069986
29. Sun Z, Ren H, Yen GG, Chen T, Wu J, An H, Yang J (2022) An evolutionary algorithm with constraint relaxation strategy for highly constrained multiobjective optimization. In: IEEE transactions on cybernetics
30. Yuan J, Liu HL, Ong YS, He Z (2022) Indicator-based evolutionary algorithm for solving constrained multiobjective optimization problems. IEEE Trans Evolut Comput 26(2):379–391. https://doi.org/10.1109/TEVC.2021.3089155
31. Xiang Y, Yang X, Huang H, Wang J (2021) Balancing constraints and objectives by considering problem types in constrained multiobjective optimization. IEEE Trans Cybern. https://doi.org/10.1109/TCYB.2021.3089633
32. Wu G, Wen X, Wang L, Pedrycz W, Suganthan PN (2022) A voting-mechanism-based ensemble framework for constraint handling techniques. IEEE Trans Evolut Comput 26(4):646–660. https://doi.org/10.1109/TEVC.2021.3110130
33. Peng C, Liu HL, Gu F (2017) An evolutionary algorithm with directed weights for constrained multi-objective optimization. Appl Soft Comput 60:613–622
34. Qiao K, Yu K, Qu B, Liang J, Song H, Yue C (2022) An evolutionary multitasking optimization framework for constrained multiobjective optimization problems. IEEE Trans Evolut Comput 26(2):263–277. https://doi.org/10.1109/TEVC.2022.3145582
35. Ma Z, Wang Y (2021) Shift-based penalty for evolutionary constrained multiobjective optimization and its application. IEEE Trans Cybern. https://doi.org/10.1109/TCYB.2021.3069814
36. Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. IEEE Trans Evolut Comput 18(4):577–601. https://doi.org/10.1109/TEVC.2013.2281535

37. Fan Z, Li W, Cai X, Huang H, Fang Y, You Y, Mo J, Wei C, Goodman E (2019) An improved epsilon constraint-handling method in MOEA/D for CMOPS with large infeasible regions. Soft Comput 23(23):12491–12510
38. Fan Z, Li W, Cai X, Li H, Wei C, Zhang Q, Deb K, Goodman E (2020) Difficulty adjustable and scalable constrained multiobjective test problem toolkit. Evolut Comput 28(3):339–378
39. While L, Hingston P, Barone L, Huband S (2006) A faster algorithm for calculating hypervolume. IEEE Trans Evolut Comput 10(1):29–38
40. Bosman PA, Thierens D (2003) The balance between proximity and diversity in multiobjective evolutionary algorithms. IEEE Trans Evolut Comput 7(2):174–188
41. Knowles J (2006) Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Trans Evolut Comput 10(1):50–66
42. Zitzler E, Laumanns M, Thiele L (2001) Spea2: improving the strength pareto evolutionary algorithm. TIK-Rep **103**
43. Price K, Storn RM, Lampinen JA (2006) Differential evolution: a practical approach to global optimization. Springer, New York
44. Deb K, Goyal M et al (1996) A combined genetic adaptive search (GENEAS) for engineering design. Comput Sci Inform 26:30–45
45. Deb K, Agrawal RB et al (1995) Simulated binary crossover for continuous search space. Complex Syst 9(2):115–148
46. Tian Y, Cheng R, Zhang X, Jin Y (2017) Platemo: a MATLAB platform for evolutionary multi-objective optimization [educational forum]. IEEE Comput Intell Mag 12(4):73–87
47. Kannan B, Kramer SN (1994) An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design
48. Narayanan S, Azarm S (1999) On improving multiobjective genetic algorithms for design optimization. Struct Optim 18:146–155
49. Chiandussi G, Codegone M, Ferrero S, Varesio FE (2012) Comparison of multi-objective optimization methodologies for engineering applications. Comput Math Appl 63(5):912–942
50. Deb K et al (1999) Evolutionary algorithms for multi-criterion optimization in engineering design. Evolut Algorithms Eng Comput Sci 2:135–161
51. Osyczka A, Kundu S (1995) A genetic algorithm-based multicriteria optimization method. In: Proceedings of the 1st world congress structural and multidisciplinary optimization, pp 909–914 (1995)
52. Azarm S, Tits A, Fan M (1999) Tradeoff-driven optimization-based design of mechanical systems. In: 4th symposium on multidisciplinary analysis and optimization, pp 4758
53. Ray T, Liew K (2002) A swarm metaphor for multiobjective design optimization. Eng Optim 34(2):141–153
54. Jain H, Deb K (2013) 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 Evolut Comput 18(4):602–622
55. Cheng F, Li X (1999) Generalized center method for multiobjective engineering optimization. Eng Optim 31(5):641–661

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.