



VICTORIA UNIVERSITY
MELBOURNE AUSTRALIA

Multi-objective multi-criteria evolutionary algorithm for multi-objective multi-task optimization

This is the Published version of the following publication

Du, Ke-Jing, Li, Jian-Yu, Wang, Hua and Zhang, Jun (2022) Multi-objective multi-criteria evolutionary algorithm for multi-objective multi-task optimization. *Complex and Intelligent Systems*, 9. pp. 1211-1228. ISSN 2199-4536

The publisher's official version can be found at
<https://link.springer.com/article/10.1007/s40747-022-00650-8>
Note that access to this version may require subscription.

Downloaded from VU Research Repository <https://vuir.vu.edu.au/47100/>



Multi-objective multi-criteria evolutionary algorithm for multi-objective multi-task optimization

Ke-Jing Du¹ · Jian-Yu Li² · Hua Wang¹ · Jun Zhang¹

Received: 13 July 2021 / Accepted: 10 December 2021 / Published online: 22 February 2022
© The Author(s) 2022

Abstract

Evolutionary multi-objective multi-task optimization is an emerging paradigm for solving multi-objective multi-task optimization problem (MO-MTOP) using evolutionary computation. However, most existing methods tend to directly treat the multiple multi-objective tasks as different problems and optimize them by different populations, which face the difficulty in designing good knowledge transferring strategy among the tasks/populations. Different from existing methods that suffer from the difficult knowledge transfer, this paper proposes to treat the MO-MTOP as a multi-objective multi-criteria optimization problem (MO-MCOP), so that the knowledge of all the tasks can be inherited in a same population to be fully utilized for solving the MO-MTOP more efficiently. To be specific, the fitness evaluation function of each task in the MO-MTOP is treated as an evaluation criterion in the corresponding MO-MCOP, and therefore, the MO-MCOP has multiple relevant evaluation criteria to help the individual selection and evolution in different evolutionary stages. Furthermore, a probability-based criterion selection strategy and an adaptive parameter learning method are also proposed to better select the fitness evaluation function as the criterion. By doing so, the algorithm can use suitable evaluation criteria from different tasks at different evolutionary stages to guide the individual selection and population evolution, so as to find out the Pareto optimal solutions of all tasks. By integrating the above, this paper develops a multi-objective multi-criteria evolutionary algorithm framework for solving MO-MTOP. To investigate the proposed algorithm, extensive experiments are conducted on widely used MO-MTOPs to compare with some state-of-the-art and well-performing algorithms, which have verified the great effectiveness and efficiency of the proposed algorithm. Therefore, treating MO-MTOP as MO-MCOP is a potential and promising direction for solving MO-MTOP.

Keywords Multi-objective optimization · Multi-criteria optimization · Multi-task optimization · Evolutionary algorithm · Evolutionary computation

التطوري متعدد الأهداف التحسين · متعدد المعايير الأمثل · متعدد المهام الأمثل · الخوارزمية التطورية الإحتساب · "الرئيسيه الكلمات

Introduction

Multi-objective multi-task optimization (MO-MTO) [1–3] is a novel and emerging paradigm that aims to solve multiple multi-objective optimization tasks simultaneously. The core assumption of MO-MTO lies in that the knowledge and information gained from the optimization of one task

can be used to enhance the optimization of the other tasks [4–7]. For instance, when the optimal solutions of two tasks share similarities in some dimensions, the knowledge of the current best solution in one task (no matter single or multi-objective) can guide the evolutionary search for the other task. Therefore, MO-MTO that tackles the multiple tasks together during the evolutionary process can be more efficient than the traditional multi-objective optimization diagram that only considers one multi-objective optimization task during the evolutionary process. Moreover, the above assumption is usually supported in a variety of real-world optimization applications [8–12]. For instance, the vehicle routing problem [8, 9], with various constraints such as the vehicle capacity, vehicle number, and time constraints, usually has

✉ Jian-Yu Li
jianyulics@qq.com

¹ Institute for Sustainable Industries and Liveable Cities, Victoria University, Melbourne, VIC 8001, Australia

² School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

a lot of specific problem instances that are somewhat similar in their problem characteristics, function landscapes, and optimal solutions. As a result, the MO-MTO research has attracted increasing attention in recent years.

When dealing with the MO-MTO problem (MO-MTOP), evolutionary computation (EC) methods are usually adopted, which leads to a promising research topic, i.e., evolutionary MO-MTO (denoted as EMO-MTO in the following contents) [13]. This is due to the fact that EC algorithms are powerful and efficient in tackling various complex optimization problems with different characteristics and difficulties [14–17]. The widely used EC algorithms include genetic algorithm (GA) [18–20], particle swarm optimization (PSO) [21–23], differential evolution (DE) [24–27], estimation of distribution algorithm (EDA) [28–30], ant colony optimization (ACO) [31–33], and evolutionary search (ES) [34, 35]. Hence, by integrating EC algorithms with the MO-MTO paradigm, the EMO-MTO is potential for solving complex MO-MTOP more efficiently.

To date, some EMO-MTO algorithms have been proposed for solving MO-MTOP, which can be roughly classified into two categories. The first category is about the multifactorial-based approach [3, 13, 36–38], while the second category is about the non-multifactorial-based approach [39–41]. In the first category, the multi-objective multifactorial evolutionary algorithm (MO-MFEA) [3] is one of the most classical and representative algorithm frameworks, which evolves a single population in a unified search space with skill factors for solving multiple tasks. Due to the efficiency of the MO-MFEA, some enhanced MO-MFEA variants have been further studied and proposed for solving MO-MTOPs [13, 36, 37]. Differently, the second category mainly maintains multiple populations for solving multiple tasks [39–41]. For example, Lin et al. [39] proposed an incremental learning method to transfer common knowledge among populations to help solve relevant tasks. For another example, Feng et al. [41] proposed a novel autoencoder method to transfer the promising individuals among different populations targeted at different tasks.

However, existing EMO-MTO algorithms, no matter using a single population with multiple groups (based on the MO-MFEA framework) or using multiple populations, still treat the multiple tasks in the MO-MTOP as different problems when optimizing them during the evolution procedure. Therefore, the existing EMO-MTO algorithms have to work with a well-designed knowledge transferring strategy among the tasks. However, designing a good knowledge transferring strategy is a very difficult issue. Differently, a recent work in [42] has shown that it is beneficial to consider the multiple tasks as different evaluation criteria during the optimization process of the single-objective MTO. This way, the knowledge for optimizing different tasks can be reserved in the optimization process and be naturally shared among all the

tasks. Following this, we can use the multiple relevant criteria to evolve the population accordingly, so as to search for the optimal solutions for all different tasks in one run. Inspired by the idea of treating multi-tasks as multi-criteria [42], we attempt to treat the MO-MTOP as a multi-objective multi-criteria optimization problem (MO-MCOP), so that the knowledge for different tasks can be fully leveraged to solve the MO-MTOP more efficiently. That is, we treat the MO-MTOP with multiple multi-objective optimization tasks as an MO-MCOP with multiple evaluation criteria, where the multiple criteria are used for the environmental selection and population evolution. By doing so, the challenging issue in MO-MTOP, i.e., how to find the useful knowledge and then transfer them across different relevant multi-objective tasks, becomes a simpler issue: how to utilize the multiple evaluation criteria to guide the environmental selection and population evolution, so as to generate the optimal solutions that satisfy different criteria of all tasks. Therefore, this research direction has a great potential of both leading to a significant approach for dealing with MO-MTOPs and providing significant contributions to the developments of related research communities.

The major novelties and contributions of this paper are listed as follows:

First, this paper attempts to solve MO-MTOPs by treating them as MO-MCOPs, which provides a novel and potential way for handling MO-MTOPs. Moreover, to the best of our knowledge, this paper is the first that tries to tackle MO-MTOP by treating it as MO-MCOP. Besides, this paper also mathematically discusses why treating MO-MTOP as MO-MCOP can be effective and efficient.

Second, a probability-based criterion selection strategy (PCSS) is proposed to select and utilize the multiple evaluation criteria based on the corresponding probability, so that different criteria can have different corresponding chances to be selected to guide the environmental selection and population evolution.

Third, an adaptive parameter learning (APL) method is further proposed to learn the probability adaptively for choosing criteria in PCSS. By adopting the APL, the algorithm can learn the suitable probability to help determine which criterion should be used at the current generation, so as to guide the population evolution with different criteria more appropriately.

Fourth, by integrating the above, a multi-objective multi-criteria evolutionary algorithm (MO-MCEA) framework is developed for solving MO-MTOPs.

To evaluate the proposed methods, extensive experiments are conducted on widely used MO-MTOP benchmarks. Moreover, some state-of-the-art and well-performing EMO-MTO algorithms have also been adopted to compare and challenge the proposed MO-MCEA.

The rest contents are as follows: the next section briefly introduces background knowledge and related work. The following section gives the motivation and analysis of treating MO-MTOP as MO-MCOP. The next section describes the proposed methods. Experimental studies are provided in the next section. Finally, the last section concludes the paper.

Background and related work

Multi-objective multi-task optimization

MO-MTO is a diagram for solving multiple multi-objective optimization tasks together. Mathematically, the MO-MTOP can be defined as follows.

Given K multi-objective optimization tasks (assuming the objectives in every task are all minimization problems), denoted as T_1, T_2, \dots, T_K , where the k^{th} task has M_k ($M_k > 1$) objective functions $F_k(x) = [f_1(x), f_2(x), \dots, f_{M_k}(x)]$. The search space and the objective space of the k^{th} task are Ω_k and Ψ^M_k , respectively, and they satisfy that $F_k: \Omega_k \rightarrow \Psi^M_k$. The aim of a minimization MO-MTOP is to find the optimal solution set $\{x_k\}$ for each task T_k , such that $\{x_k\}$ satisfies

$$\{x_k\} = \operatorname{argmin} F_k(x_k | x_k \in \Omega_k), k = 1, 2, 3, \dots, K. \quad (1)$$

As each F_k has multiple objectives, we will have the following important concepts for each task T_k to determine whether a solution is optimal according to the related definitions in the literature of multi-objective optimization [43, 44].

Definition 1 *Pareto domination* Given any two objective fitness vectors $u = [u_1, u_2, \dots, u_M]$ and $w = [w_1, w_2, \dots, w_M]$ in the objective space Ψ^M , we say that u dominates w if $u_m \leq w_m$ for all $m = 1, 2, \dots, M$ and $u \neq w$, denoted as $u \preceq w$.

Definition 2 *Pareto optimal* A solution vector $x \in \Omega$ is Pareto optimal if there is no $x^* \in \Omega$, such that $F(x^*)$ dominates $F(x)$.

Definition 3 *Pareto set* The Pareto set (PS) is a set of the Pareto optimal solutions, which can be represented as

$$\text{PS} = \{x \in \Omega \text{ and } x \text{ is Pareto optimal}\}. \quad (2)$$

Definition 4 *Pareto front* The Pareto front (PF) is composed of the solutions in PS, as

$$\text{PF} = \{F(x) | x \in \text{PS}\}. \quad (3)$$

Based on the above definitions, the optimal solution set $\{x_k\}$ for each task T_k is actually the PS of the T_k .

Related work

To date, although the EMO-MTO is a newly emerged optimization diagram, some EMO-MTO works have been proposed and attracted increasing attention. Therefore, this part provides a brief review of the existing works.

As mentioned briefly in the Introduction part, existing works about EMO-MTO can be categorized into two major categories. The first category is about the multifactorial-based approach [3, 13, 36–38], while the second category is about the non-multifactorial-based approach [39–41].

In the first category, MO-MFEA is the most representative algorithm framework for solving MO-MTOPs [3]. When compared with other algorithms, the distinct feature of MO-MFEA is that the MO-MFEA evolves one population with skill factors to find the optimal solutions for multiple tasks together. In MO-MFEA, each individual in the population corresponds to one task according to their skill factors, and thereby individuals for different tasks can transfer common knowledge implicitly via genetic operations, e.g., crossover operation. By utilizing omnidirectional knowledge transfers, the MO-MFEA can achieve mutual knowledge sharing among different multi-objective optimization tasks, so that the optimization for each task can be benefited by the obtained knowledge from other tasks. To make the knowledge sharing more adaptively, Bali et al. [13] studied the intertask relationship learning and proposed an online adaptive genetic transfer method to develop the MO-MFEA-II, which can achieve better overall performance than the original MO-MFEA. Moreover, as MO-MFEA may convergence slowly if tasks are weakly relevant or even irrelevant, Zheng et al. [36] proposed to introduce additional helper tasks via the weighted sum of original tasks to improve the knowledge transfer and speed up the algorithm convergence. Furthermore, Yang et al. [37] considered not only the convergence but also the diversity of MO-MFEA and proposed a two-stage assortative mating method to enhance the knowledge transfer among diversity-related and convergence-related variables among related tasks. Besides, Binh et al. [38] proposed a reference-point-based approach and an enhanced random mating probability learning method to better exploit and transfer knowledge among individuals targeted at different tasks.

In the second category, algorithms maintain multiple populations with explicitly transfer information among the populations for solving MO-MTOPs. For example, Lin et al. [39] proposed an algorithm based on incremental learning to find suitable knowledge for the transfer among different tasks. Liang et al. [40] proposed a genetic transform strategy to transfer the individual genetic information from one task to relevant tasks. In addition, Feng et al. [41] proposed an explicit autoencoding method to achieve knowledge transfer

among the population targeted at different tasks to enhance the optimization results.

However, these existing methods and algorithms for MO-MTOPs still treat the multiple tasks in the MO-MTOP as different problems and optimize them simultaneously, which requires a well-designed knowledge transferring strategy to share knowledge among tasks. Differently from these methods, the MO-MCEA proposed in this paper treats all tasks in MO-MTOP as multiple criteria of an entire MO-MTOP to guide the evolution appropriately, so as to fully utilize the knowledge in different tasks during the optimization process and obtain promising solutions for all tasks. Therefore, the contributions and novelties of this paper are justified.

Motivation of treating MO-MTOP as MO-MCOP

Definition of MCOP and MO-MCOP

In this paper, the MCOP is defined as follows:

Given an optimization problem (assuming it is a minimization problem) with K available evaluation criteria (including objective/constraint functions) on the same search space Ω , which are denoted as F_1, F_2, \dots, F_K , the aim of a minimization MCOP is to find the optimal solution set $\{x\}$ that satisfies

$$\begin{aligned} \{x\} &= \arg \min F_0(x|x \in \Omega) \\ \text{with } F_0 &\in \{F_1, F_2, \dots, F_K\}, \end{aligned} \quad (4)$$

where F_0 can be arbitrary one in $\{F_1, F_2, \dots, F_K\}$ at different search stages. Note that the “argmin” should be “argmax” if the problem is a maximization optimization problem, and the $\{x\}$ will only have one element if the problem is single-objective. The key characteristic of MCOP is that when each evaluation function can guide the optimization to find acceptable optimal solutions for the problem, then the proper usage of multiple evaluation functions can obtain a satisfactory solution more efficiently. Moreover, if the F_1, F_2, \dots, F_K have different fidelities or accuracy scales, then the MCOP can be considered as a multifidelity optimization problem [45] or multi-scale optimization problem [46], respectively. In addition, the relationship between MCOP and MTOP is that both of them have multiple evaluation functions, while the main difference between them lies in that the MCOP requires the algorithm to optimize only one evaluation function each time, while the MTOP requires the algorithm to optimize multiple different evaluation functions together each time.

Based on the above, the MO-MCOP is the same as MCOP except that all evaluation functions in $\{F_1, F_2, \dots, F_K\}$ are multi-objective functions.

Relationship between MO-MTOP and MO-MCOP

This part discusses the relationship between MO-MTOP and MO-MCOP. To begin with, we consider a unified search space Ω for all different tasks in the MO-MTOP, where the search space of the k th task is Ω_k . Considering K one-to-one mapping functions $\varphi_1, \varphi_2, \dots$, and φ_K , where $\varphi_k: \Omega \rightarrow \Omega_k$, then Eq. (1) for MO-MTOP can be rewritten as

$$\begin{aligned} \{x_k\} &= \arg \min F_k(x_k|x_k \in \Omega_k) \\ &= \arg \min F_k(\varphi_k(x)|x \in \Omega) \\ &= \arg \min F_k \circ \varphi_k(x|x \in \Omega), \quad k = 1, 2, 3, \dots, K. \end{aligned} \quad (5)$$

Now, the MO-MTOP is with K tasks with the same search space (i.e., Ω), where the multi-objective evaluation function of the k th task is $F_k \circ \varphi_k$ and of all tasks are the same. Moreover, if considering $F_1 \circ \varphi_1, F_2 \circ \varphi_2, \dots$, and $F_K \circ \varphi_K$ as different evaluation functions in a MO-MCOP, then the aim of MO-MCOP, i.e., Eq. (4), can be rewritten as

$$\begin{aligned} \{x\} &= \arg \min F_0(x|x \in \Omega) \\ \text{with } F_0 &\in \{F_1 \circ \varphi_1, F_2 \circ \varphi_2, \dots, F_K \circ \varphi_K\}. \end{aligned} \quad (6)$$

Note that we have $\{\varphi_k(x)\} = \{x_k\}$ for $k = 1, 2, \dots, K$ as $\varphi_k: \Omega \rightarrow \Omega_k$. Therefore, comparing Eqs. (5) and (6), the main difference between MO-MTOP and MO-MCOP is that given a set of evaluation functions, MO-MTOP aims to optimize all evaluation functions by considering them together all the time during the optimization process, while MO-MCOP attempts to obtain the optimal solutions by considering one evaluation function every time during the optimization process, where the latter (i.e., MO-MCOP) can be an easier problem, because the optimization algorithms can select an appropriate function at different stages adaptively and flexibly to guide the evolution to obtain better results. Therefore, it would be better if MO-MTOP can be treated as MO-MCOP.

Treating MO-MTOP as MO-MCOP

Based on the above, MO-MTOP can be treated as MO-MCOP with multiple criteria (i.e., evaluation functions) and the algorithm can select the proper criterion in different stages to guide the evolution. Herein, this part further analyzes the rationality and the benefit of treating MO-MTOP as MO-MCOP in the following contents.

The rationality of treating MO-MTOP as MO-MCOP

In fact, the key issue of rationality depends on the difference between the optimal results obtained from Eq. (5) and those from Eq. (6), i.e., the equivalence degree of MO-MTOPs and MO-MCOPs. Without loss of generality, we consider an

MO-MTOP with two tasks T_i and T_j , where the evaluation functions and Pareto sets of these two tasks are $F_i \circ \varphi_i$ and $F_j \circ \varphi_j$ and PS_i and PS_j , respectively. Therefore, according to Eq. (5), the optimal results after solving the MO-MTOP should include PS_i and PS_j . Moreover, according to Eq. (6), if $F_i \circ \varphi_i$ is selected as the evaluation criterion all the time when solving the MO-MCOP, the optimal results after solving the MO-MCOP will be PS_i . That is, the difference between the optimal results obtained by Eq. (5) (i.e., solving MO-MTOP) and those by Eq. (6) (i.e., solving MO-MCOP) can be considered as the difference between $PS_i \cup PS_j$ and PS_i , i.e., the difference between PS_i and PS_j .

Based on the above, we can have three observations on the rationality of treating MO-MTOP as MO-MCOP.

First, if T_i and T_j in an MO-MTOP have high similarity and share much common knowledge in their Pareto sets PS_i and PS_j , e.g., $PS_i = PS_j$ or $PS_i \approx PS_j$, then the results obtained by Eq. (5) (i.e., by solving MO-MTOP) and those obtained by Eq. (6) (i.e., by solving MO-MCOP) are similar. In such a situation, treating the MO-MTOP as an MO-MCOP can find the Pareto sets for both tasks.

Second, if T_i and T_j share some similarities in their Pareto sets, i.e., $PS_i \cap PS_j \neq \emptyset$ but $PS_i \neq PS_j$, then the PS_i will contain some Pareto optimal solutions for T_j . As real-world multi-objective optimization tasks usually require enough Pareto optimal solutions rather than all optimal solutions, the number of Pareto optimal solutions in PS_i can be enough to constitute an acceptable Pareto set for T_j . In this scenario, treating the MO-MTOP as an MO-MCOP can find the acceptable solutions for both tasks.

Third, if T_i and T_j are very different, e.g., $PS_i \cap PS_j \approx \emptyset$, these two tasks are not suggested to be integrated together as an MO-MTOP. In this circumstance, the problem with such tasks is actually not a well-defined MO-MTOP, and therefore, it is no need to (and is also not recommended to) treat it as an MO-MCOP.

Based on the above, it is reasonable to treat MO-MTOPs as MO-MCOPs.

The benefit of treating MO-MTOP as MO-MCOP

When treating an MO-MTOP as an MO-MCOP, the algorithm can select an evaluation function as the criterion to evolve the population. To begin with, we can consider an MO-MTOP with all tasks in the same search space (i.e., mapping functions φ are the identity functions $\varphi(x) = x$ and can be omitted) as an MO-MCOP, and then define the following notation:

Given the population at the g th generation and the $(g+z)$ th generation, i.e., P_g and P_{g+z} , respectively, $\Pr(P_{g+z} < P_g | P_g, F_i)$ denotes the probability that no solutions in P_{g+z} are dominated by any solutions in P_g after the P_g evolves z gen-

erations with the multi-objective evaluation function F_i as the selection criterion to become P_{g+z} .

Note that theory analysis and experimental studies have shown that if an algorithm has a larger probability for producing a better population in a given number of generations, the algorithm will have a faster convergence speed and a smaller time complexity [47, 48]. Therefore, a larger $\Pr(P_{g+z} < P_g | P_g, F_i)$ will be more satisfied.

Based on the above, we can discuss the benefit of treating MO-MTOP as MO-MCOP as follows:

Without loss of generality, we can assume that $\Pr(P_{g+z} < P_g | P_g, F_i) = a_g$, $\Pr(P_{g+2z} < P_{g+z} | P_{g+z}, F_i) = a_{g+z}$, $\Pr(P_{g+z} < P_g | P_g, F_j) = b_g$, and $\Pr(P_{g+2z} < P_{g+z} | P_{g+z}, F_j) = b_{g+z}$, where a_g, a_{g+z}, b_g , and b_{g+z} all belong to $[0, 1]$. In general, as different multi-objective functions have different landscapes and the different numbers of local optima, the a_g, a_{g+z}, b_g , and b_{g+z} may be different from each other. Then, if only one of the evaluation functions (e.g., F_i or F_j) is selected as the criterion to evolve population P_g for $2z$ generations, we can have the following equations:

$$\begin{aligned} &\Pr(P_{g+2z} < P_{g+z} < P_g | P_g, F_i) \\ &= \Pr(P_{g+2z} < P_{g+z} | P_{g+z}, F_i) \\ &\quad \cdot \Pr(P_{g+z} < P_g | P_g, F_i) = a_{g+z} \cdot a_g, \end{aligned} \tag{7}$$

$$\begin{aligned} &\Pr(P_{g+2z} < P_{g+z} < P_g | P_g, F_j) \\ &= \Pr(P_{g+2z} < P_{g+z} | P_{g+z}, F_j) \\ &\quad \cdot \Pr(P_{g+z} < P_g | P_g, F_j) = b_{g+z} \cdot b_g. \end{aligned} \tag{8}$$

However, if the problem is treated as an MO-MCOP and the F_i and F_j are selected appropriately as the criteria at different stages (e.g., during the z generations) to evolve P_g , so as to maximize $\Pr(P_{g+2z} < P_{g+z} < P_g | P_g, F_i \text{ or } F_j)$, then we can have

$$\begin{aligned} &\max \Pr(P_{g+2z} < P_{g+z} < P_g | P_g, F_i \text{ or } F_j) \\ &= \max\{\Pr(P_{g+2z} < P_{g+z} | P_{g+z}, F_i \text{ or } F_j) \cdot \Pr(P_{g+z} < P_g | P_g, F_i \text{ or } F_j)\} \\ &= \max\{\Pr(P_{g+2z} < P_{g+z} | P_{g+z}, F_i), \Pr(P_{g+2z} < P_{g+z} | P_{g+z}, F_j)\} \\ &\quad \cdot \max\{\Pr(P_{g+z} < P_g | P_g, F_i), \Pr(P_{g+z} < P_g | P_g, F_j)\} \\ &= \max\{a_{g+z}, b_{g+z}\} \cdot \max\{a_g, b_g\}. \end{aligned} \tag{9}$$

Note that the “ F_i or F_j ” in the formula means that the algorithm can use F_i or F_j during the evolution from P_g to

P_{g+z} and from P_{g+z} to P_{g+2z} . Then, by combining Eqs. (7), (8), and (9), we can have the inequation

$$\begin{aligned} \max \Pr(P_{g+2z} < P_{g+z} < P_g \mid P_g, F_i \text{ or } F_j) \\ &= \max\{a_{g+z}, b_{g+z}\} \cdot \max\{a_g, b_g\} \\ &\geq \max\{a_{g+z} \cdot a_g, b_{g+z} \cdot b_g\} \\ &= \max\{\Pr(P_{g+2z} < P_{g+z} < P_g \mid P_g, F_i), \Pr(P_{g+2z} \\ &< P_{g+z} < P_g \mid P_g, F_j)\}. \end{aligned} \tag{10}$$

That is, when compared with the approach that only uses one function as the criterion for populations for different tasks, treating the MO-MTOP as an MO-MCOP and using multiple fitness functions properly at different stages can make the population have a larger probability to become a better population within the given number of generations, which can result in a higher evolution efficiency and a faster convergence speed. Therefore, treating MO-MTOP as MO-MCOP properly can bring more sufficient knowledge sharing among multiple tasks to benefit the population evolution [49–51].

Based on the above, it is rational and beneficial to treat MO-MTOPs as MO-MCOPs and choose the criterion properly at different evolution stages. Therefore, this paper also proposes the PCSS and APL to select the criterion in different stages properly, which will be described in the following contents.

The proposed MO-MCEA

The overall framework of the MO-MCEA

The overall framework of the MO-MCEA is presented in Fig. 1, which mainly has three parts. The first part is a criterion set of available evaluation functions, the second part is the criterion selection based on PCSS and APL to select one of the evaluation functions as the evaluation criterion to

evolve population at different evolutionary stages, and the third part is the procedure of evolutionary optimization with the selected criterion. In the following contents, the PCSS, APL, and the complete algorithm will be introduced one by one.

Probability-based criterion selection strategy

The PCSS is proposed to select one multi-objective function from multiple multi-objective functions to be the selection criterion for the current generation. The idea of PCSS is straightforward: each multi-objective function will have a criterion selection probability (denoted as csp), e.g., F_i has csp_i , and the multi-objective function with a larger csp will have more opportunities to be selected as the selection criterion. Following this, given K multi-objective functions $F_1, F_2, \dots,$ and F_K , and their corresponding selection probability $csp_1, csp_2, \dots,$ and csp_K , the index of the function selected as the criterion will be determined by the roulette with $csp_1, csp_2, \dots,$ and csp_K . Mathematically, the selection of the criterion can be written as

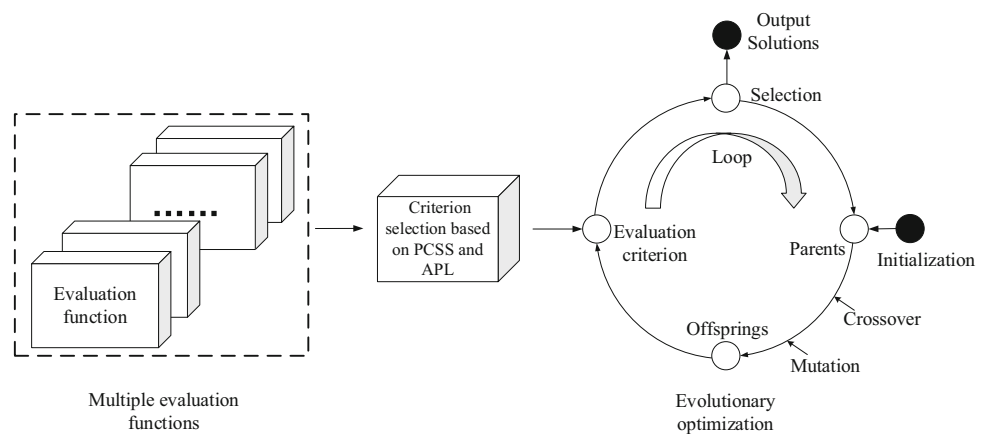
$$cid = \text{roulette}(csp), \tag{11}$$

where cid is the index of the selected criterion (i.e., the selected criterion is denoted as F_{cid}), and the function roulette returns an index based on a roulette with $csp_1, csp_2, \dots,$ and csp_K as the probabilities for the available indexes 1 to K .

Adaptive parameter learning

The initial csp is initialized evenly for different available functions and the APL aims to learn more suitable csp adaptively during the evolution, so that the PCSS can select the criterion more properly. In general, if the population has better improvements after one generation under the current selection criterion, this criterion may be more suitable for the current evolutionary stage, and vice versa. Therefore, the

Fig. 1 The overall framework of MO-MCEA



APL updates the csp according to the population improvement in every generation. To be specific, considering the population at g th generation (i.e., P_g) uses the cid th multi-objective function (i.e., F_{cid}) as the selection criterion, then the APL will update csp_{cid} as

$$csp_{cid} = \begin{cases} csp_{cid} + \Delta, & \text{if } P_{g+1} \text{ is better than } P_g \\ csp_{cid} - \Delta, & \text{otherwise} \end{cases}, \quad (12)$$

where Δ is a fixed value for updating csp , and the “better” can be determined by comparing P_{g+1} and P_g based on various metrics that do not rely on the real Pareto front, such as C metric [52] and hypervolume [32]. Note that the proposed MO-MCEA in this paper uses the C metric to compare P_{g+1} and P_g . Besides the csp_{cid} , the other csp (i.e., csp_j where $j \neq cid$) will also be updated as

$$csp_j = \begin{cases} csp_j - \frac{\Delta}{K-1}, & \text{if } P_{g+1} \text{ is better than } P_g \\ csp_j + \frac{\Delta}{K-1}, & \text{otherwise} \end{cases}, \quad (13)$$

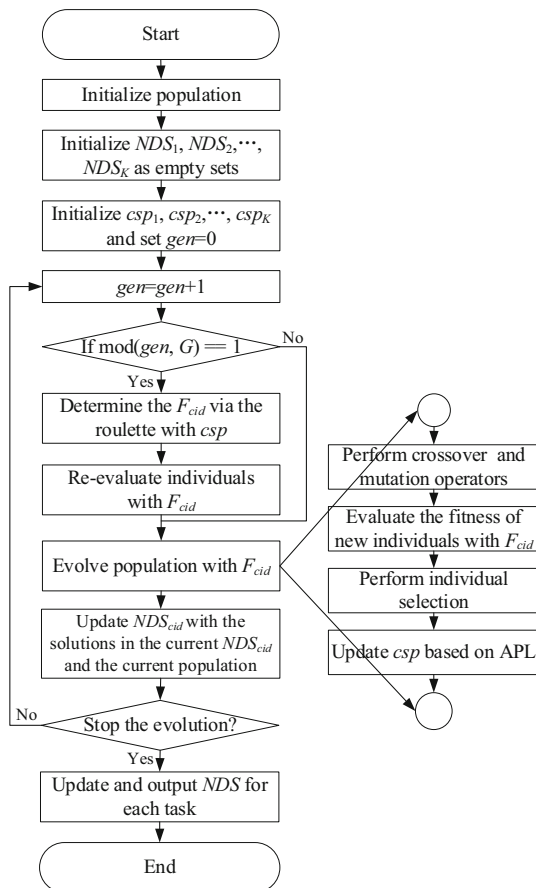


Fig. 2 The flowchart of the complete MO-MCEA

where K is the total number of available multi-objective tasks. Note that if a csp_i is smaller than 0.1, then it will be set as 0.1 and then all csp are normalized, so as to guarantee that the i th function still has a chance to be selected again and the sum of csp equals 1.

The complete MO-MCEA

With the PCSS and APL, the flowchart of the complete MO-MCEA is given in Fig. 2 and the pseudo code of the complete MO-MCEA is shown as Algorithm 1. Note that the search spaces of all tasks will be normalized in the unified search space $[0,1]^D$, where D is the maximum variable dimension among all tasks. That is, solutions for different tasks are mapped to the $[0, 1]^D$, which is a common technique in the literature [4]. As can be seen, after the initialization, Algorithm 1 mainly has three repeated procedures. These three procedures are the criterion determination via PCSS, population evolution, and the parameter learning for PCSS through APL, which can be seen in lines 10–15, lines 16–18, and lines 19–20 of Algorithm 1, respectively. Note that the population evolution procedures in Algorithm 1 can use various existing well-designed operators including efficient crossover, mutation, and selection operators according to the needs and preferences of users. Therefore, the MO-MCEA can be extended with powerful state-of-the-art methods and operators to develop more efficient algorithms. In this paper, the optimization algorithm for each task is the NSGA-II [44]. In addition, it should also be noted that the evaluation criterion will be switched by PCSS every G generations. Moreover, every time the evaluation criterion is switched, the current population will be re-evaluated by the new evaluation criterion before going to the evolution. Therefore, NP fitness evaluations (FEs) are needed after every switch, just as shown in lines 13 and 14 of Algorithm 1. Overall, the algorithm will repeat the PCSS, population evolution, and APL iteratively until the stop criterion is met, e.g., all available FEs are consumed. Note that we use K sets (denoted as $NDS_1, NDS_2, \dots, NDS_K$) to record the current non-dominated solutions for the K tasks, respectively. During the evolutionary process, after the population evolution of one generation with F_{cid} (i.e., the line 16 of Algorithm 1), the corresponding NDS_{cid} will be updated. That is, all the solutions in the current population will merge with the solutions in NS_{cid} , and then, only those non-dominated solutions in the merged solution set can be remained in the NDS_{cid} . After the evolutionary process, all the solutions in the final population will be, respectively, evaluated by each of the K multi-objective functions and be merged with the corresponding NDS to update the NDS (the update process is similar to that in line 18), see lines 23–24 in Algorithm 1. Finally, the algorithm outputs the best-found non-dominated sets for all different tasks.

Algorithm 1: The Complete MO-MCEA

Input: T_1, T_2, \dots, T_K - K optimization tasks;
 F_1, F_2, \dots, F_K -the fitness function of the T_1, T_2, \dots, T_K ;
 NP -the number of individuals in the population;
 G -the number of generations for using each criterion.

Output: $NDS_1, NDS_2, \dots, NDS_K$ -the current non-dominated solution sets for K tasks.

```

1:Begin
2:   Initialize  $NP$  individuals;
3:   Initialize  $NDS_1, NDS_2, \dots, NDS_K$  as empty sets;
4:   For  $i = 1$  to  $K$ 
5:      $csp_i \leftarrow 1/K$ ; // Initialize the criterion selection probability evenly
6:   End For
7:    $FES \leftarrow 0$ ;
8:    $gen \leftarrow 1$ ; // index of generation
9:   While ( $FES + NP \times K <$  maximum number of available FEs) Do
10:    If  $\text{mod}(gen, G) = 1$ 
11:      // Probability-based Criterion Selection Strategy
12:       $cid \leftarrow$  roulette selection of  $[1, 2, \dots, K]$  with  $[csp_1, csp_2, \dots, csp_K]$ ;
13:      Re-evaluate individuals by the function  $F_{cid}$ ;
14:       $FES \leftarrow FES + NP$ ;
15:    End If
16:    Evolve population for one generation with  $F_{cid}$ ;
17:     $FES \leftarrow FES + NP$ ;
18:    Update and record the corresponding  $NDS_{cid}$ ;
19:    // Adaptive Parameter Learning
20:    Update  $[csp_1, csp_2, \dots, csp_K]$  according to Eq.(12) and Eq.(13);
21:     $gen \leftarrow gen + 1$ ;
22:  End While
23:  Evaluate individuals by the  $K$  multi-objective functions  $F_1, F_2, \dots, F_K$ ;
24:  Update and record  $NDS_1, NDS_2, \dots, NDS_K$ , respectively;
25: End

```

Experimental studies

Experiment setup

In the experimental studies, six commonly used MO-MTOPs [53] with different similarities are adopted to investigate the proposed methods and algorithm. The properties of these problems are briefly introduced in Table 1, where the task similarity is evaluated by the Spearman's rank correlation coefficient [53]. In Table 1, the complete intersection (CI) and partial intersection (PI) indicate that the Pareto optimal solutions of the two tasks are similar in all and some dimensions, respectively, while high similarity (HS), medium similarity (MS), and low similarity (LS) mean that the two tasks have high, medium, and low similarity measured by Pearson correlation according to their function landscape. Based on this, the six problems with different properties can be categorized into different categories according to their intersection degree and task similarity. For example, the CI + HS problem has the property of complete intersection and high similarity

between its two internal tasks. Moreover, as shown in Table 1, different multi-objective tasks will also have different Pareto fronts. Therefore, the composition of different intersections, similarity degrees, and Pareto fronts can help provide an in-depth observation of how the proposed algorithm may behave in various situations. In addition, the dimensions of all tasks are set as 10, because tasks with high dimensions can be difficult to solve (e.g., contains many local optima) and different algorithms may have similar results (e.g., similar very poor results) on the high-dimensional tasks, which is not ideal for algorithm comparison and analysis.

To evaluate the effectiveness and efficiency of the proposed algorithm, some state-of-the-art and latest well-performing algorithms with various characteristics are used in comparisons. The adopted algorithms include the MO-MFEA [3], MO-MFEA-II [13], and EMT with autoencoding for MO-MTOPs (denoted as MO-EMTA for simplicity) [41]. To make the comparisons fair, all these algorithms use the same widely used and representative MOEA (i.e., NSGA-II [44]) as the optimizer. By doing so, the differences of the proposed MO-MCEA and the three compared algorithms only

Table 1 The properties of six adopted multi-objective multi-task optimization problems

Problem ID	Tasks ID	D	Search space	Pareto set	Pareto front	Properties	Problem category	Task similarity
MO-MTOP1	Task 1	10	$x_1 \in [0, 1], x_i \in [-100, 100]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to D	$f_2 1 + f_2 2 = 1, f_1 \geq 0, f_2 \geq 0$	Concave, unimodal, separable	Complete intersection (CI) and high similarity (HS)	0.97
	Task 2	10	$x_1 \in [0, 1], x_i \in [-100, 100]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to D	$f_2 = 1 - f_2 1, 0 \leq f_1 \leq 1$	Concave, unimodal, separable		
MO-MTOP2	Task 1	10	$x_1 \in [0, 1], x_i \in [-5, 5]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 1$ for $i = 2$ to D	$f_2 = 1 - f_2 1, 0 \leq f_1 \leq 1$	Concave, multi-modal, nonseparable	Complete intersection (CI) and medium similarity (MS)	0.52
	Task 2	10	$x_1 \in [0, 1], x_i \in [-5, 5]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 1$ for $i = 2$ to D	$f_2 1 + f_2 2 = 1, f_1 \geq 0, f_2 \geq 0$	Concave, unimodal, nonseparable		
MO-MTOP3	Task 1	10	$x_1 \in [0, 1], x_i \in [-2, 2]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to D	$f_2 1 + f_2 2 = 1, f_1 \geq 0, f_2 \geq 0$	Concave, multi-modal, separable	Complete intersection (CI) and low similarity (LS)	0.07
	Task 2	10	$x_1 \in [0, 1], x_i \in [-1, 1]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to D	$f_2 = 1 - \sqrt{f_1}, 0 \leq f_1 \leq 1$	Convex, multi-modal, nonseparable		
MO-MTOP4	Task 1	10	$x_1 \in [0, 1], x_i \in [-100, 100]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to D	$f_2 = 1 - \sqrt{f_1}, 0 \leq f_1 \leq 1$	Convex, unimodal, separable	Partial intersection (PI) and high similarity (HS)	0.99
	Task 2	10	$x_1 \in [0, 1], x_i \in [-100, 100]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to $D/2, x_j = 20$ for $j = D/2 + 1$ to D	$f_2 = 1 - \sqrt{f_1}, 0 \leq f_1 \leq 1$	Convex, multi-modal, separable		
MO-MTOP5	Task 1	10	$x_1 \in [0, 1], x_i \in [-100, 100]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to D	$f_2 1 + f_2 2 = 1, f_1 \geq 0, f_2 \geq 0$	Concave, unimodal, nonseparable	Partial intersection (PI) and medium similarity (MS)	0.55
	Task 2	10	$x_1 \in [0, 1], x_i \in [-100, 100]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to $D/2, x_j = 20$ for $j = D/2 + 1$ to D	$f_2 = 1 - f_2 1, 0 \leq f_1 \leq 1$	Concave, multi-modal, nonseparable		
MO-MTOP6	Task 1	10	$x_1 \in [0, 1], x_i \in [-100, 100]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to D	$f_2 1 + f_2 2 = 1, f_1 \geq 0, f_2 \geq 0$	Concave, multi-modal, nonseparable	Partial intersection(PI) and low similarity (LS)	0.002
	Task 2	10	$x_1 \in [0, 1], x_i \in [-100, 100]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to $D/2, x_j = 20$ for $j = D/2 + 1$ to D	$f_2 1 + f_2 2 = 1, f_1 \geq 0, f_2 \geq 0$	Concave, multi-modal, nonseparable		

lie in their approach or way for handling MO-MTOPs, e.g., handle MO-MTOPs as MC-MTOPs or handle MO-MTOPs via multifactorial-based approach). Moreover, an NSGA-II integrated with the single-task evolutionary optimization paradigm (i.e., solving each task separately and independently) is also adopted as a baseline in the comparisons, where the algorithm is denoted as MO-STEAs in the following contents. All the algorithm settings are kept consistent with their original papers. As for the NSGA-II operators, the settings are set as the same as those used in existing EMO-MTO literature [13]. Moreover, the G and Δ in MO-MCEA are set as 25 and 0.01, respectively. In addition, the population size of individual budgets for each task is set as 50. Therefore, the total population size of both MO-MCEA and MO-STEAs is 50, while the total population size of MO-MFEA, MO-MFEA-II, and MO-EMTA are all 100.

In the experiments, the maximum number of available FEs is 1×10^4 in total (i.e., 5×10^3 for each task) for each independent run of all algorithms. To reduce statistical errors, each algorithm runs 30 times independently on each MO-MTOP and the results are collected for the comparisons, as suggested by the literature [53].

Evaluation metrics

To evaluate the performance of MO-MCEA, four evaluation metrics are adopted in the experimental studies and comparisons. The first two metrics are the mean and standard value of the optimization results of each algorithm for each task over 30 runs. As the tasks are multi-objective optimization tasks, the widely used inverted generational distance (IGD) indicator [43] for multi-objective optimization is adopted to evaluate the performance of an algorithm on each task. The IGD value can be calculated with a set of objective vectors obtained by an algorithm A (denoted as P^A) and a set of the objective vectors uniformly distributed over the true PF (refer to “Multi-objective multi-task optimization”) of a multi-objective task (denoted as P^*), which is mathematically defined as

$$\text{IGD}(P^A, P^*) = \frac{1}{|P^*|} \sqrt{\sum_{\mathbf{x} \in P^*} \left(\min_{\mathbf{y} \in P^A} d(\mathbf{x}, \mathbf{y}) \right)^2}, \quad (14)$$

where $d(\mathbf{x}, \mathbf{y})$ calculates the Euclidean distance between \mathbf{x} and \mathbf{y} in the objective space, and $|P^*|$ is the number of vectors in P^* . In general, the smaller the $\text{IGD}(A, P^*)$ is, the better the algorithm A is, because a smaller IGD indicates that P^A is more close to P^* or contains more data in P^* . In addition, if $|P^*|$ is large enough to represent the PF, the $\text{IGD}(P^A, P^*)$ can measure both the convergence and diversity of the non-dominated solutions obtained by A . In other words, in this paper, the first two metrics are the mean and standard value

of the IGD value obtained by each algorithm for each task over 30 runs, respectively.

Furthermore, the third metric is the Wilcoxon’s rank sum test with a significant level $\alpha = 0.05$ [43] for algorithm comparisons. Specifically, based on the Wilcoxon’s rank sum test, the symbols “+”, “ \approx ”, and “−” are used to represent that the proposed MO-MCEA performs significantly better than, similar to, and significantly worse than the compared algorithms, respectively.

The fourth metric is the mean standard score (MSS) designed for comparing algorithms on MO-MTOPs [53]. This metric is defined as follows. Assuming that there are N algorithms denoted as A_1, A_2, \dots, A_N for an MO-MTOP with K multi-objective minimization tasks T_1, T_2, \dots, T_K , each algorithm runs for R repetitions and the MSS of algorithm A_i can be defined as

$$\text{MSS}_i = \sum_{k=1}^K \sum_{r=1}^R \text{IGD}'(i, k)_r, \quad (15)$$

where $\text{IGD}'(i, k)_r$ is the normalized IGD value obtained by an algorithm A_i for task T_k in r th independent run. The normalization process can be written as

$$\text{IGD}'(i, k)_r = \frac{\text{IGD}(i, k)_r - u_k}{\sigma_k}, \quad (16)$$

where $\text{IGD}(i, k)_r$ is the original IGD value obtained by A_i on task T_k in the r th independent run, and u_k and σ_k represent the mean and standard deviation of IGD value for task T_k over all the R repetitions of all the N algorithms, i.e., the mean and standard value of $N \times R$ IGD results.

Comparisons with state-of-the-art algorithms

To investigate the performance of the proposed MO-MCEA, this part compares it with four algorithms, including the state-of-the-art and well-performing MO-MTO algorithms including MO-MFEA [3], MO-MFEA-II [13], and MO-EMTA [41], and one single-task algorithm MO-STEAs as the baseline. The comparison results in Table 2 show the great efficiency of MO-MCEA. As shown in Table 2, the MO-MCEA can obtain the best MSS on 3 problems, while the MO-MFEA, MO-MFEA-II, MO-EMTA, and MO-STEAs get the best MSS only on 0, 1, 2, and 1 problem(s), respectively. Moreover, according to the Wilcoxon’s rank sum test, the MO-MCEA can significantly outperform MO-MFEA, MO-MFEA-II, MO-EMTA, and MO-STEAs on 7, 7, 7, and 8 tasks, and produce worse results than them only on 4, 4, 4, and 3 tasks, respectively. In addition, Fig. 3 plots the solutions found by different algorithms on the multi-objective tasks of different problems for a better result visualization. As can be observed from Table 2 and Fig. 3, the MO-MCEA performs

Table 2 Comparisons between the proposed MO-MCEA and state-of-the-art algorithms

Problem			MO-MCEA	MO-MFEA	MO-MFEA-II	MO-EMTA	MO-STEA	
MO-MTOP1 (CI + HS)	Task 1 (T_1)	Mean	1.68E−01	6.05E−01(+)	8.50E−01(+)	1.46E+00(+)	2.14E+00(+)	
		Std	9.46E−02	2.08E−01	3.29E−01	9.34E−01	1.13E+00	
	Task 2 (T_2)	Mean	9.35E−01	4.94E+00 (+)	2.09E+00 (+)	2.53E+00(+)	2.43E+00(+)	
		Std	3.28E−01	1.24E+00	5.17E−01	7.92E−01	5.89E−01	
	MSS			− 5.99E+01	3.31E+01	− 1.59E+01	1.18E+01	3.09E+01
MO-MTOP2 (CI + MS)	Task 1 (T_1)	Mean	1.44E−02	3.56E−02(+)	3.14E−01(+)	3.18E−01(+)	2.70E−01(+)	
		Std	4.27E−03	1.39E−02	2.09E−01	1.47E−01	1.32E−01	
	Task 2 (T_2)	Mean	1.20E−01	1.82E−01(+)	1.78E−01(+)	1.84E−01(+)	1.80E−01(+)	
		Std	9.97E−02	4.33E−02	4.09E−02	4.60E−02	4.32E−02	
	MSS			− 5.10E+01	− 1.87E+01	2.40E+01	2.75E+01	1.82E+01
MO-MTOP3 (CI + LS)	Task 1 (T_1)	Mean	3.04E−01	4.53E−01(+)	2.35E+00(+)	4.07E+00(+)	5.25E+00(+)	
		Std	5.35E−01	4.23E−01	2.50E+00	2.39E+00	1.87E+00	
	Task 2 (T_2)	Mean	9.79E−01	9.70E−01(+)	9.17E−01(+)	9.16E−01(+)	9.16E−01(+)	
		Std	5.10E−01	3.02E−01	2.13E−02	2.16E−02	2.18E−02	
	MSS			− 2.04E+01	− 1.96E+01	− 4.12E+00	1.54E+01	2.87E+01
MO-MTOP4 (PI + HS)	Task 1 (T_1)	Mean	3.74E+02	5.52E+00(−)	4.01E+00(−)	2.92E+00(−)	3.65E+00(−)	
		Std	5.55E+02	2.10E+00	1.69E+00	8.83E−01	1.12E+00	
	Task 2 (T_2)	Mean	9.75E+02	3.94E+01(−)	2.85E+01(−)	2.69E+01(−)	2.77E+01(−)	
		Std	6.44E+02	1.20E+01	8.26E+00	1.03E+01	7.34E+00	
	MSS			7.84E+01	− 1.89E+01	− 1.97E+01	− 2.00E+01	− 1.98E+01
MO-MTOP5 (PI + MS)	Task 1 (T_1)	Mean	3.87E+02	3.95E+00(≈)	3.31E+00(≈)	2.83E+00(≈)	3.70E+00(≈)	
		Std	5.82E+02	1.56E+00	1.44E+00	1.09E+00	1.20E+00	
	Task 2 (T_2)	Mean	1.01E+03	3.63E+01(−)	2.65E+01(−)	2.88E+01(−)	2.84E+01(−)	
		Std	6.98E+02	9.43E+00	7.40E+00	9.27E+00	1.01E+01	
	MSS			7.75E+01	− 1.89E+01	− 1.96E+01	− 1.95E+01	− 1.94E+01
MO-MTOP6 (PI + LS)	Task 1 (T_1)	Mean	3.18E−01	3.76E−01(+)	2.69E−01(+)	2.46E−01(+)	2.63E−01(+)	
		Std	3.29E−01	6.91E−02	6.20E−02	5.93E−02	5.47E−02	
	Task 2 (T_2)	Mean	1.75E+01	5.71E+00(−)	1.17E+01(−)	5.08E+00(−)	1.99E+01(+)	
		Std	3.22E+00	1.33E+00	6.11E+00	2.25E+00	2.39E+00	
	MSS			2.80E+01	− 1.23E+01	− 5.69E+00	− 3.86E+01	2.86E+01
Number of +/≈/−			NA	7/1/4	7/1/4	7/1/4	8/1/3	
Number of best MSS			3	0	1	2	0	

The bold values mean the best results

significantly better on the three complete intersection problems with different task similarities, i.e., the MO-MTOP1 to MO-MTOP3. This is consistent with the analysis given in Sect. 3.1 that treating MO-MTOP as MO-MCOP is reasonable and effective when the optimal Pareto set of different tasks share similarities. Overall, the comparison results have verified the efficiency of MO-MCEA.

Ablation studies for component analysis

To perform the component analysis of MO-MCEA, we compare it with its variants that do not use PCSS or APL, which are denoted as MO-MCEA-w/o-PCSS and MO-MCEA-w/o-APL, respectively. To be specific, the MO-MCEA-w/o-PCSS

selects the multiple objective functions as the criterion in sequential order with a round-robin fashion, and the MO-MCEA-w/o-APL will not update the probability parameter for determining the criterion (i.e., each of the multiple objective functions has the same probability to be randomly selected as the criterion in the PCSS).

Table 3 gives the comparison results, which indicate the contributions of both PCSS and APL. As shown in Table 3, MO-MCEA can obtain the best MSS on 3 problems, while the MO-MCEA-w/o-PCSS and MO-MCEA-w/o-APL get the best MSS only on 2 and 1 problem(s), respectively. Furthermore, the Wilcoxon’s rank sum test also shows that the MO-MCEA can perform significantly better than MO-MCEA-w/o-PCSS and MO-MCEA-w/o-APL on 8 and 7

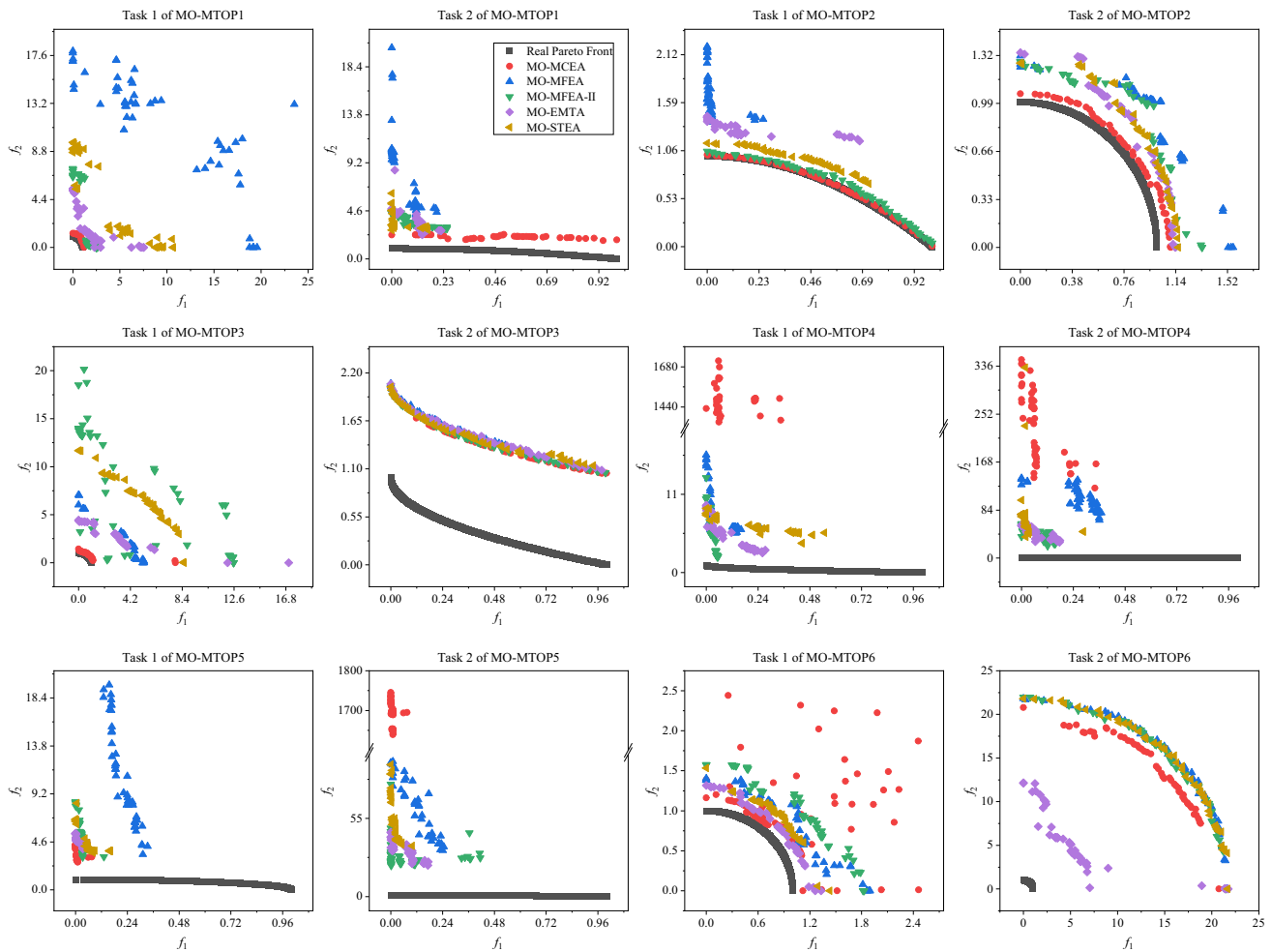


Fig. 3 Plots of solutions found by different algorithms on the multi-objective tasks of different problems

tasks and worse results only on 3 and 3 tasks, respectively, which mean that removing anyone of the PCSS and APL will degrade the performance of MO-MCEA. Therefore, it can be concluded that both PCSS and APL have their contributions to the great problem-solving ability of MO-MCEA.

Parameter studies

To study the influence of the key parameter in MO-MCEA, i.e., G , we compare the original MO-MCEA($G = 25$) with its variants using different G , i.e., $G = 1, G = 5, G = 10, G = 15, G = 20$, and $G = 30$. For simplicity, these variants are denoted as MO-MCEA($G = 1$), MO-MCEA($G = 5$), MO-MCEA($G = 10$), MO-MCEA($G = 15$), MO-MCEA($G = 20$), and MO-MCEA($G = 30$).

Table 4 provides the comparison results, which show that different MO-MTOPs may favor different G . As can be seen, the best MSS values in solving different MO-MTOPs are from different MO-MCEA variants and none of the variants can obtain the best MSS on all problems. Besides, the orig-

inal MO-MCEA($G = 25$) is shown to be more promising. Based on the Wilcoxon’s rank sum test, MO-MCEA($G = 25$) obtains significantly better results than MO-MCEA($G = 1$) and MO-MCEA($G = 5$) on 9 and 5 tasks, and worse results only on 3 and 1 task(s), respectively. Also, the MO-MCEA($G = 25$) can obtain similar results on most MO-MTOPs when compared with MO-MCEA($G = 10$), MO-MCEA($G = 15$), MO-MCEA($G = 20$), and MO-MCEA($G = 30$). In addition, although the MO-MCEA($G = 25$) obtains the best MSS on 1 problem, while the MO-MCEA($G = 1$) can get the best MSS on 2 problems, the MO-MCEA($G = 25$) significantly outperforms MO-MCEA($G = 1$) on highly intersection problems, e.g., MO-MTOP1 to MO-MTOP3. Therefore, $G = 25$ is in general a more promising setting for MO-MCEA and is recommended in this paper.

Experiments on MO-MTOP with more tasks

This part further investigates the problem-solving ability of the proposed MO-MCEA on MO-MTOPs with more than

Table 3 Comparisons among the MO-MCEA variants with or without PCSS and APL

Problem			MO-MCEA	MO-MCEA-w/o-PCSS	MO-MCEA-w/o-APL	
MO-MTOP1 (CI + HS)	Task 1 (T_1)	Mean	1.68E−01	2.14E+00(+)	1.41E−01(≈)	
		Std	9.46E−02	1.13E+00	5.34E−02	
	Task 2 (T_2)	Mean	9.35E−01	2.43E+00(+)	8.95E−01(≈)	
		Std	3.28E−01	5.89E−01	1.96E−01	
	MSS			− 3.47E+01	7.16E+01	− 3.69E+01
MO-MTOP2 (CI + MS)	Task 1 (T_1)	Mean	1.44E−02	2.70E−01(+)	3.77E−02(+)	
		Std	4.27E−03	1.32E−01	2.95E−02	
	Task 2 (T_2)	Mean	1.20E−01	1.80E−01(+)	4.23E−01(+)	
		Std	9.97E−02	4.32E−02	2.37E−01	
	MSS			− 3.82E+01	2.59E+01	1.23E+01
MO-MTOP3 (CI + LS)	Task 1 (T_1)	Mean	3.04E−01	5.25E+00(+)	3.77E+00(+)	
		Std	5.35E−01	1.87E+00	1.43E+00	
	Task 2 (T_2)	Mean	9.79E−01	9.16E−01(+)	3.74E+00(+)	
		Std	5.10E−01	2.18E−02	6.38E−01	
	MSS			− 5.30E+01	5.12E+00	4.79E+01
MO-MTOP4 (PI + HS)	Task 1 (T_1)	Mean	3.74E+02	3.65E+00(−)	1.14E+00(−)	
		Std	5.55E+02	1.12E+00	1.14E−01	
	Task 2 (T_2)	Mean	9.75E+02	2.77E+01(−)	1.62E+03(+)	
		Std	6.44E+02	7.34E+00	1.13E+01	
	MSS			2.44E+01	− 4.38E+01	1.95E+01
MO-MTOP5 (PI + MS)	Task 1 (T_1)	Mean	3.87E+02	3.70E+00(≈)	8.46E−01(−)	
		Std	5.82E+02	1.20E+00	2.20E−01	
	Task 2 (T_2)	Mean	1.01E+03	2.84E+01(−)	1.62E+03(+)	
		Std	6.98E+02	1.01E+01	1.08E+01	
	MSS			2.51E+01	− 4.34E+01	1.83E+01
MO-MTOP6 (PI + LS)	Task 1 (T_1)	Mean	3.18E−01	2.63E−01(+)	1.32E−01(−)	
		Std	3.29E−01	5.47E−02	3.74E−02	
	Task 2 (T_2)	Mean	1.75E+01	1.99E+01(+)	1.98E+01(+)	
		Std	3.22E+00	2.39E+00	6.47E−01	
	MSS			− 7.04E+00	1.34E+01	− 6.32E+00
Number of + / ≈ / −			NA	8/1/3	7/2/3	
Number of best MSS			3	2	1	

The bold values mean the best results

two tasks. For this, four additional problems with three tasks (i.e., MO-MTOP7 to MO-MTOP10) are developed based on the MO-MTOP1 to MO-MTOP6, where the problem characteristics can be seen in Table 5. The experimental settings are the same as those in previous experiments, with the FEs for each task as 5×10^3 (1.5×10^4 in total). In addition, the total population size of both MO-MCEA and MO-STEAs is 50 (because they handle one evaluation function every time), while the total population size of MO-MFEA, MO-MFEA-II, and MO-EMTA are all 150.

Experimental results of the five algorithms over 30 runs are compared in Table 6. Table 6 shows that the MO-MCEA can obtain the best MSS value on two MO-MTOPs with high similarity (i.e., MO-MTOP7 and MO-MTOP8) and MO-

EMTA gets the best MSS value on two MO-MTOPs with low similarity (i.e., MO-MTOP9 and MO-MTOP10), while other algorithms cannot gain the best MSS value on any of the four tested problems. This may be due to that MO-MCEA can treat MO-MTOP with high similarity as MO-MCOP properly and solve it efficiently. Moreover, the Wilcoxon's rank sum test reflects that the MO-MCEA can significantly outperform MO-MFEA, MO-MFEA-II, MO-EMTA, and MO-STEAs on 7, 8, 6, and 8 of the 12 tasks, and only has worse performance on 3, 4, 5, and 4 tasks, respectively. Based on the above, the experimental results have verified the effectiveness of the MO-MCEA on MO-MTOPs with three tasks.

Table 4 Comparisons among the MO-MCEA variants with different parameter G

Problem	MO-MCEA (G = 25)		MO-MCEA (G = 1)		MO-MCEA (G = 5)		MO-MCEA (G = 10)		MO-MCEA (G = 15)		MO-MCEA (G = 20)		MO-MCEA (G = 30)	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
MO-MTOP1 (CI + HS)	Task 1 (T ₁)	Mean	1.68E-01	1.84E+00(+)	3.30E-01(+)	2.19E-01(≈)	1.74E-01(≈)	1.85E-01(≈)	2.21E-01(≈)	1.85E-01(≈)	1.74E-01(≈)	1.85E-01(≈)	2.21E-01(≈)	1.85E-01(≈)
	Task 1 (T ₁)	Std	9.46E-02	1.07E+00	1.95E-01	1.23E-01	1.65E-01	1.49E-01	1.86E-01	1.65E-01	1.65E-01	1.49E-01	1.86E-01	1.86E-01
	Task 2 (T ₂)	Mean	9.35E-01	3.12E+00(+)	1.23E+00(+)	9.48E-01(≈)	8.45E-01(≈)	8.19E-01(≈)	8.33E-01(≈)	8.45E-01(≈)	8.45E-01(≈)	8.19E-01(≈)	8.33E-01(≈)	8.33E-01(≈)
	Task 2 (T ₂)	Std	3.28E-01	9.03E-01	3.67E-01	1.74E-01	3.03E-01	3.15E-01	2.60E-01	3.03E-01	3.03E-01	3.15E-01	2.60E-01	2.60E-01
MO-MTOP2 (CI + MS)	MSS		- 2.23E+01	1.22E+02	- 5.52E+00	- 1.98E+01	- 2.51E+01	- 2.55E+01	- 2.35E+01	- 2.51E+01	- 2.51E+01	- 2.55E+01	- 2.35E+01	- 2.35E+01
	Task 1 (T ₁)	Mean	1.44E-02	1.46E-01(+)	2.12E-02(+)	1.81E-02(≈)	1.69E-02(≈)	1.97E-02(≈)	1.93E-02(≈)	1.69E-02(≈)	1.69E-02(≈)	1.97E-02(≈)	1.93E-02(≈)	1.93E-02(≈)
	Task 1 (T ₁)	Std	4.27E-03	1.13E-01	1.11E-02	9.99E-03	7.70E-03	1.28E-02	1.16E-02	7.70E-03	7.70E-03	1.28E-02	1.16E-02	1.16E-02
	Task 2 (T ₂)	Mean	1.20E-01	2.91E-01(+)	1.32E-01(≈)	1.34E-01(≈)	1.28E-01(≈)	1.58E-01(≈)	1.57E-01(≈)	1.28E-01(≈)	1.28E-01(≈)	1.58E-01(≈)	1.57E-01(≈)	1.57E-01(≈)
MO-MTOP3 (CI + LS)	MSS		- 2.06E+01	8.53E+01	- 1.45E+01	- 1.53E+01	- 1.73E+01	- 8.58E+00	- 8.94E+00	- 1.73E+01	- 1.73E+01	- 8.58E+00	- 8.94E+00	- 8.94E+00
	Task 1 (T ₁)	Mean	3.04E-01	1.68E+00(+)	2.78E-01(≈)	3.23E-01(≈)	1.72E-01(≈)	2.12E-01(≈)	3.74E-01(≈)	1.72E-01(≈)	1.72E-01(≈)	2.12E-01(≈)	3.74E-01(≈)	3.74E-01(≈)
	Task 1 (T ₁)	Std	5.35E-01	2.76E+00	3.57E-01	4.44E-01	7.20E-02	1.87E-01	5.35E-01	7.20E-02	7.20E-02	1.87E-01	5.35E-01	5.35E-01
	Task 2 (T ₂)	Mean	9.79E-01	1.14E+00(+)	9.05E-01(≈)	1.01E+00(≈)	8.77E-01(≈)	9.30E-01(≈)	9.98E-01(≈)	8.77E-01(≈)	8.77E-01(≈)	9.30E-01(≈)	9.98E-01(≈)	9.98E-01(≈)
MO-MTOP4 (PI + HS)	MSS		- 4.08E+00	4.24E+01	- 1.05E+01	- 1.48E+00	- 1.53E+01	- 1.02E+01	- 8.50E-01	- 1.53E+01	- 1.48E+00	- 1.02E+01	- 8.50E-01	- 8.50E-01
	Task 1 (T ₁)	Mean	3.74E+02	3.85E+02(+)	3.18E+02(≈)	3.26E+02(≈)	3.66E+02(≈)	4.25E+02(≈)	3.97E+02(≈)	3.66E+02(≈)	3.66E+02(≈)	4.25E+02(≈)	3.97E+02(≈)	3.97E+02(≈)
	Task 1 (T ₁)	Std	5.55E+02	3.36E+02	3.80E+02	4.39E+02	4.86E+02	5.39E+02	5.63E+02	4.86E+02	4.86E+02	5.39E+02	5.63E+02	5.63E+02
	Task 2 (T ₂)	Mean	9.75E+02	4.72E+02(-)	7.35E+02(≈)	8.74E+02(≈)	8.70E+02(≈)	7.79E+02(≈)	9.15E+02(≈)	8.70E+02(≈)	8.70E+02(≈)	7.79E+02(≈)	9.15E+02(≈)	9.15E+02(≈)
MO-MTOP5 (PI + MS)	MSS		6.44E+02	3.27E+02	5.57E+02	5.98E+02	6.64E+02	6.24E+02	6.62E+02	6.64E+02	6.64E+02	6.24E+02	6.62E+02	6.62E+02
	Task 1 (T ₁)	Mean	8.71E+00	1.53E+01	- 6.57E+00	7.20 E-01	3.05E+00	2.25E+00	7.17E+00	3.05E+00	3.05E+00	2.25E+00	7.17E+00	7.17E+00
	Task 1 (T ₁)	Std	3.87E+02	3.55E+02(+)	5.03E+02(+)	3.69E+02(≈)	3.59E+02(≈)	5.83E+02(+)	3.69E+02(≈)	5.83E+02(+)	3.59E+02(≈)	5.83E+02(+)	3.69E+02(≈)	3.69E+02(≈)
	Task 2 (T ₂)	Mean	5.82E+02	2.88E+02	4.96E+02	4.75E+02	5.04E+02	6.27E+02	5.28E+02	5.04E+02	5.04E+02	6.27E+02	5.28E+02	5.28E+02
MO-MTOP6 (PI + LS)	MSS		1.01E+03	4.81E+02(-)	4.88E+02(-)	7.97E+02(≈)	9.08E+02(≈)	7.21E+02(≈)	8.98E+02(≈)	9.08E+02(≈)	9.08E+02(≈)	7.21E+02(≈)	8.98E+02(≈)	8.98E+02(≈)
	Task 1 (T ₁)	Mean	6.98E+02	3.38E+02	4.48E+02	5.57E+02	6.32E+02	6.46E+02	6.42E+02	6.32E+02	6.32E+02	6.46E+02	6.42E+02	6.42E+02
	Task 1 (T ₁)	Std	1.08E+01	1.73E+01	- 8.33E+00	- 9.50E-01	3.95 E+00	7.77E+00	4.03E+00	3.95 E+00	3.95 E+00	7.77E+00	4.03E+00	4.03E+00
	Task 2 (T ₂)	Mean	3.18E-01	5.10E-01(+)	3.33E-01(+)	3.43E-01(≈)	4.25E-01(+)	3.51E-01(≈)	3.10E-01(≈)	4.25E-01(+)	4.25E-01(+)	3.51E-01(≈)	3.10E-01(≈)	3.10E-01(≈)
Number of + / ≈ / -	Task 1 (T ₁)	Mean	3.29E-01	1.89E-01	1.61E-01	2.46E-01	3.02E-01	2.80E-01	3.05E-01	3.02E-01	2.80E-01	3.05E-01	3.05E-01	3.05E-01
	Task 2 (T ₂)	Mean	1.75E+01	1.63E+01(-)	1.70E+01(≈)	1.74E+01(≈)	1.69E+01(≈)	1.73E+01(≈)	1.80E+01(≈)	1.69E+01(≈)	1.69E+01(≈)	1.73E+01(≈)	1.80E+01(≈)	1.80E+01(≈)
	Task 2 (T ₂)	Std	3.22E+00	2.71E+00	3.12E+00	2.85E+00	3.54E+00	3.44E+00	2.97E+00	3.54E+00	3.54E+00	3.44E+00	2.97E+00	2.97E+00
	MSS		- 3.30E+00	7.16E+00	- 6.25E+00	- 6.70E-01	3.45E+00	- 9.60E-01	5.70E-01	3.45E+00	3.45E+00	- 9.60E-01	5.70E-01	5.70E-01
Number of best MSS		NA	9/0/3	5/6/1	0/12/0	1/11/0	1/11/0	1/11/0	1/11/0	1/11/0	1/11/0	1/11/0	0/12/0	0/12/0
Number of + / ≈ / -		1	2	1	0	1	1	1	1	1	1	1	0	

The bold values mean the best results

Table 5 The four multi-objective multi-task optimization problems with three tasks

Problem ID	Tasks ID	Sources	D	Search space	Pareto set	Pareto front	Properties	Similarity degree
MO-MTOP7	Task 1	Task 1 of MO-MTOP1	10	$x_1 \in [0, 1], x_i \in [-100, 100]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to D	$f_2 1 + f_2 2 = 1, f_1 \geq 0, f_2 \geq 0$	Concave, unimodal, separable	High
	Task 2	Task 2 of MO-MTOP1	10	$x_1 \in [0, 1], x_i \in [-100, 100]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to D	$f_2 = 1 - f_2 1, 0 \leq f_1 \leq 1$	Concave, unimodal, separable	
	Task 3	Task 1 of MO-MTOP2	10	$x_1 \in [0, 1], x_i \in [-5, 5]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to D	$f_2 = 1 - f_2 1, 0 \leq f_1 \leq 1$	Concave, multi-modal, nonseparable	
MO-MTOP8	Task 1	Task 2 of MO-MTOP2	10	$x_1 \in [0, 1], x_i \in [-5, 5]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to D	$f_2 1 + f_2 2 = 1, f_1 \geq 0, f_2 \geq 0$	Concave, unimodal, nonseparable	High
	Task 2	Task 1 of MO-MTOP3	10	$x_1 \in [0, 1], x_i \in [-2, 2]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to D	$f_2 1 + f_2 2 = 1, f_1 \geq 0, f_2 \geq 0$	Concave, multi-modal, separable	
	Task 3	Task 2 of MO-MTOP3	10	$x_1 \in [0, 1], x_i \in [-1, 1]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to D	$f_2 = 1 - \sqrt{f_1}, 0 \leq f_1 \leq 1$	Convex, multi-modal, nonseparable	
MO-MTOP9	Task 1	Task 1 of MO-MTOP4	10	$x_1 \in [0, 1], x_i \in [-100, 100]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to D	$f_2 = 1 - \sqrt{f_1}, 0 \leq f_1 \leq 1$	Convex, unimodal, separable	Low
	Task 2	Task 2 of MO-MTOP4	10	$x_1 \in [0, 1], x_i \in [-100, 100]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to D	$f_2 = 1 - \sqrt{f_1}, 0 \leq f_1 \leq 1$	Convex, multi-modal, separable	
MO-MTOP10	Task 3	Task 1 of MO-MTOP5	10	$x_1 \in [0, 1], x_i \in [-100, 100]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to D	$f_2 1 + f_2 2 = 1, f_1 \geq 0, f_2 \geq 0$	Concave, unimodal, nonseparable	
	Task 1	Task 2 of MO-MTOP5	10	$x_1 \in [0, 1], x_i \in [-100, 100]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to D	$f_2 = 1 - f_2 1, 0 \leq f_1 \leq 1$	Concave, multi-modal, nonseparable	Low
	Task 2	Task 1 of MO-MTOP6	10	$x_1 \in [0, 1], x_i \in [-100, 100]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to D	$f_2 1 + f_2 2 = 1, f_1 \geq 0, f_2 \geq 0$	Concave, multi-modal, nonseparable	
Task 3	Task 2 of MO-MTOP6	10	$x_1 \in [0, 1], x_i \in [-100, 100]$ for $i = 2$ to D	$x_1 \in [0, 1], x_i = 0$ for $i = 2$ to D	$f_2 1 + f_2 2 = 1, f_1 \geq 0, f_2 \geq 0$	Concave, multi-modal, nonseparable		

Table 6 Comparisons between the proposed MO-MCEA and state-of-the-art algorithms on MO-MTOPs with three tasks

Problem			MO-MCEA	MO-MFEA	MO-MFEA-II	MO-EMTA	MO-STEA
MO-MTOP7 (high similarity)	Task 1 (T_1)	Mean	9.50E−02	1.38E+01(+)	9.00E−01(+)	1.88E+00(+)	2.14E+00(+)
		Std	4.42E−02	6.63E+00	3.93E−01	9.88E−01	1.13E+00
	Task 2 (T_2)	Mean	5.96E−01	7.72E+00(+)	2.15E+00(+)	2.33E+00(+)	2.43E+00(+)
		Std	1.20E−01	1.73E+00	5.14E−01	5.81E−01	5.89E−01
	Task 3 (T_3)	Mean	4.57E−02	4.22E−01(+)	1.98E−01(+)	7.88E−02(+)	2.70E−01(+)
		Std	4.66E−02	1.53E−01	1.15E−01	6.31E−02	1.32E−01
	MSS		− 7.04E+01	1.44E+02	− 2.29E+01	− 3.57E+01	− 1.31E+00
MO-MTOP8 (high similarity)	Task 1 (T_1)	Mean	6.88E−02	5.28E−01(+)	1.42E−01(+)	1.75E−01(+)	1.80E−01(+)
		Std	1.71E−02	1.18E−01	3.95E−02	3.13E−02	4.32E−02
	Task 2 (T_2)	Mean	3.07E−02	6.22E+00(+)	1.35E−01(+)	5.90E+00(+)	5.25E+00(+)
		Std	1.38E−02	1.86E+00	1.26E−01	1.83E+00	1.87E+00
	Task 3 (T_3)	Mean	8.56E−01	1.07E+00(+)	9.66E−01(+)	9.08E−01(+)	9.16E−01(+)
		Std	2.08E−03	6.35E−02	7.86E−02	1.36E−02	2.18E−02
	MSS		− 8.93E+01	1.18E+02	− 4.05E+01	1.72E+00	− 9.99E−01
MO-MTOP9 (low similarity)	Task 1 (T_1)	Mean	1.83E+02	1.63E+01(−)	3.81E+00(−)	3.50E+00(−)	3.65E+00(−)
		Std	3.39E+02	5.74E+00	1.43E+00	1.16E+00	1.12E+00
	Task 2 (T_2)	Mean	9.17E−01	7.40E−01(≈)	5.44E−01(−)	5.71E−01(−)	2.77E+01(+)
		Std	4.25E−01	7.91E−02	6.51E−02	6.02E−02	7.34E+00
	Task 3 (T_3)	Mean	1.62E+01	1.56E+01 (≈)	1.97E+01(+)	5.77E+00(−)	3.70E+00(−)
		Std	4.80E+00	5.28E+00	2.70E+00	3.50E+00	1.20E+00
	MSS		1.98E+01	− 1.29E+01	2.66E+00	− 6.01E+01	2.69E+00
MO-MTOP10 (low similarity)	Task 1 (T_1)	Mean	1.41E+02	6.67E+01(−)	2.68E+01(−)	2.54E+01(−)	2.14E+00(−)
		Std	3.67E+02	1.61E+01	8.77E+00	6.79E+00	1.13E+00
	Task 2 (T_2)	Mean	1.23E+00	4.51E−01(−)	2.94E−01(−)	2.47E−01(−)	2.43E+00(+)
		Std	3.71E−01	9.07E−02	6.52E−02	5.08E−02	5.89E−01
	Task 3 (T_3)	Mean	6.29E+00	1.60E+01(+)	1.94E+01(+)	6.03E+00(≈)	2.70E−01(−)
		Std	4.92E+00	5.20E+00	3.37E+00	4.35E+00	1.32E−01
	MSS		2.88E+00	2.53E+00	4.33E+00	− 5.16E+01	5.49E+00
Number of + / ≈ / −		NA	7/2/3	8/0/4	6/1/5	8/0/4	
Number of best MSS		2	0	0	2	0	

The bold values mean the best results

Conclusion

In this paper, we have attempted to treat MO-MTOP as MO-MCOP and solve it efficiently. For this, we have provided an analysis of the rationality and benefit of treating MO-MTOP as MO-MCOP. Moreover, we have further proposed the PCSS to select different multi-objective fitness functions as criteria in different evolutionary stages to evolve individuals. In addition, the APL method has been further proposed to learn the parameter in PCSS adaptively. Based on PCSS and APL, the complete algorithm framework called MO-MCEA has been developed for solving MO-MTOPs. To investigate the proposed methods and algorithm, experiments have been conducted on widely used MO-MTOP benchmarks. Also, four state-of-the-art and well-performing algorithms have been adopted in comparisons to challenge the proposed

MO-MCEA. The experimental results have shown the great effectiveness and efficiency of the proposed MO-MCEA, showing that treating MO-MTOP as MO-MCOP can be a potential way for solving MO-MTOP more efficiently.

For future work, the MO-MCEA, including the PCSS and APL, will be further improved and extended to solve more difficult MO-MTOPs (e.g., MO-MTOPs with different similarities and intersections) and more complex real-world MO-MTOPs, such as those are also with other difficult characteristics in data-driven optimization problems [54, 55], expensive optimization problems [56–58], multi-modal optimization problems [59], large-scale optimization problems [60, 61], and many-objective optimization problems [62, 63].

Acknowledgements This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2102102, in part by the Key-Area Research and Development

of Guangdong Province under Grant 2020B010166002, in part by the National Natural Science Foundations of China (NSFC) under Grant 62176094 and Grant 61873097, and in part by the Guangdong Natural Science Foundation Research Team under Grant 2018B030312003.

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

- Tan KC, Feng L, Jiang M (2021) Evolutionary transfer optimization—a new frontier in evolutionary computation research. *IEEE Comput Intell Mag* 16(1):22–33. <https://doi.org/10.1109/MCI.2020.3039066>
- Ong YS, Gupta A (2019) AIR5: five pillars of artificial intelligence research. *IEEE Trans Emerg Top Comput Intell* 3(5):411–415. <https://doi.org/10.1109/TETCI.2019.2928344>
- Gupta A, Ong YS, Feng L, Tan KC (2017) Multiobjective multifactorial optimization in evolutionary multitasking. *IEEE Trans Cybern* 47(7):1652–1665. <https://doi.org/10.1109/TCYB.2016.2554622>
- Gupta A, Ong YS, Feng L (2016) Multifactorial evolution: toward evolutionary multitasking. *IEEE Trans Evol Comput* 20(3):343–357. <https://doi.org/10.1109/TEVC.2015.2458037>
- Gupta A, Ong YS, Feng L (2017) Insights on transfer optimization: because experience is the best teacher. *IEEE Trans Emerg Top Comput Intell* 2(1):51–64. <https://doi.org/10.1109/TETCI.2017.2769104>
- Ong YS, Gupta A (2016) Evolutionary multitasking: a computer science view of cognitive multitasking. *Cogn Comput* 8(2):125–142. <https://doi.org/10.1007/s12559-016-9395-7>
- Li G, Lin Q, Gao W (2020) Multifactorial optimization via explicit multipopulation evolutionary framework. *Inf Sci* 512:1555–1570. <https://doi.org/10.1016/j.ins.2019.10.066>
- Feng L, Zhou L, Gupta A, Zhong J, Qin K (2019) Solving generalized vehicle routing problem with occasional drivers via evolutionary multitasking. *IEEE Trans Cybern* 51(6):3171–3184. <https://doi.org/10.1109/TCYB.2019.2955599>
- Feng L et al (2020) Towards faster vehicle routing by transferring knowledge from customer representation. *IEEE Trans Intell Transp Syst.* <https://doi.org/10.1109/TITS.2020.3018903>
- Wang H, Feng L, Jin Y, Doherty J (2020) Surrogate-assisted evolutionary multitasking for expensive minimax optimization in multiple scenarios. *IEEE Comput Intell Mag* 16(1):34–48. <https://doi.org/10.1109/MCI.2020.3039067>
- Li JY, Zhan ZH, Zhang J (2022) Evolutionary computation for expensive optimization: a survey. *Mach Intell Res* 19(1):3–23. <https://doi.org/10.1007/s11633-022-1317-4>
- Cheng MY, Gupta A, Ong YS, Ni ZW (2017) Coevolutionary multitasking for concurrent global optimization: With case studies in complex engineering design. *Eng Appl Artif Intell* 64:13–24. <https://doi.org/10.1016/j.engappai.2017.05.008>
- Bali KK, Gupta A, Ong YS, Tan PS (2021) Cognizant multitasking in multiobjective multifactorial evolution: MO-MFEA-II. *IEEE Trans Cybern* 51(4):1784–1796. <https://doi.org/10.1109/TCYB.2020.2981733>
- Zhan ZH, Shi L, Tan KC, Zhang J (2021) A survey on evolutionary computation for complex continuous optimization. *Artif Intell Rev.* <https://doi.org/10.1007/s10462-021-10042-y>
- Zhan ZH et al (2021) Matrix-based evolutionary computation. *IEEE Trans Emerg Top Comput Intell.* <https://doi.org/10.1109/TETCI.2020.3047410>
- Li JY, Zhan ZH, Tan KC, Zhang J (2021) A meta-knowledge transfer-based differential evolution for multitask optimization. *IEEE Trans Evol Comput.* <https://doi.org/10.1109/TEVC.2021.3131236>
- Li JY, Zhan ZH, Xu J, Kwong S, Zhang J (2021) Surrogate-assisted hybrid-model estimation of distribution algorithm for mixed-variable hyperparameters optimization in convolutional neural networks. *IEEE Trans Neural Netw Learn Syst.* <https://doi.org/10.1109/TNNLS.2021.3106399>
- Yi J, Bai J, He H, Zhou W, Yao L (2020) A multifactorial evolutionary algorithm for multitasking under interval uncertainties. *IEEE Trans Evol Comput* 24(5):908–922. <https://doi.org/10.1109/TEVC.2020.2975381>
- Li JY, Zhan ZH, Wang H, Zhang J (2021) Data-driven evolutionary algorithm with perturbation-based ensemble surrogates. *IEEE Trans Cybern* 51(8):3925–3937. <https://doi.org/10.1109/TCYB.2020.3008280>
- Li JY, Zhan ZH, Wang C, Jin H, Zhang J (2020) Boosting data-driven evolutionary algorithm with localized data generation. *IEEE Trans Evol Comput* 24(5):923–937. <https://doi.org/10.1109/TEVC.2020.2979740>
- Lin Q et al (2018) Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems. *IEEE Trans Evol Comput* 22(1):32–46. <https://doi.org/10.1109/TEVC.2016.2631279>
- Zhang X, Du KJ, Zhan ZH, Kwong S, Gu TL, Zhang J (2020) Cooperative co-evolutionary bare-bones particle swarm optimization with function independent decomposition for large-scale supply chain network design with uncertainties. *IEEE Trans Cybern* 50(10):4454–4468. <https://doi.org/10.1109/TCYB.2019.2937565>
- Li JY, Zhan ZH, Liu RD, Wang C, Kwong S, Zhang J (2021) Generation-level parallelism for evolutionary computation: a pipeline-based parallel particle swarm optimization. *IEEE Trans Cybern* 51(10):4848–4859
- Sun J, Liu X, Bäck T, Xu Z (2021) Learning adaptive differential evolution algorithm from optimization experiences by policy gradient. *IEEE Trans Evol Comput.* <https://doi.org/10.1109/TEVC.2021.3060811>
- Zhan ZH, Wang ZJ, Jin H, Zhang J (2020) Adaptive distributed differential evolution. *IEEE Trans Cybern* 50(11):4633–4647. <https://doi.org/10.1109/TCYB.2019.2944873>
- Liu XF, Zhan ZH, Zhang J (2021) Resource-aware distributed differential evolution for training expensive neural-network-based controller in power electronic circuit. *IEEE Trans Neural Netw Learn Syst.* <https://doi.org/10.1109/TNNLS.2021.3075205>
- Yang M, Zhou A, Li C, Yao X (2021) An efficient recursive differential grouping for large-scale continuous problems. *IEEE Trans Evol Comput* 25(1):159–171. <https://doi.org/10.1109/TEVC.2020.3009390>
- Wang F, Li Y, Zhou A, Tang K (2019) An estimation of distribution algorithm for mixed-variable newsvendor problems. *IEEE Trans Evol Comput* 24(3):479–493. <https://doi.org/10.1109/TEVC.2019.2932624>
- Chen ZG, Lin Y, Gong YJ et al (2021) Maximizing lifetime of range-adjustable wireless sensor networks: a neighborhood-

- based estimation of distribution algorithm. *IEEE Trans Cybern* 51:5433–5444. <https://doi.org/10.1109/TCYB.2020.2977858>
30. Sun Y, Yen GG, Yi Z (2018) Improved regularity model-based eda for many-objective optimization. *IEEE Trans Evol Comput* 22(5):662–678. <https://doi.org/10.1109/TEVC.2018.2794319>
 31. Zhang X, Zhan ZH, Fang W, Qian P, Zhang J (2021) Multi population ant colony system with knowledge based local searches for multiobjective supply chain configuration. *IEEE Trans Evol Comput*. <https://doi.org/10.1109/TEVC.2021.3097339>
 32. Zhou S-Z, Zhan Z-H, Chen Z-G et al (2020) A multi-objective ant colony system algorithm for airline crew rostering problem with fairness and satisfaction. *IEEE Trans Intell Transp Syst* 22:6784–6798. <https://doi.org/10.1109/tits.2020.2994779>
 33. Liang D, Zhan ZH, Zhang Y, Zhang J (2020) An efficient ant colony system approach for new energy vehicle dispatch problem. *IEEE Trans Intell Transp Syst* 21(11):4784–4797. <https://doi.org/10.1109/TITS.2019.2946711>
 34. Zhou L, Feng L, Gupta A, Ong YS (2021) Learnable evolutionary search across heterogeneous problems via kernelized autoencoding. *IEEE Trans Evol Comput* 25(3):567–581. <https://doi.org/10.1109/TEVC.2021.3056514>
 35. Feng L, Zhou W, Liu W, Ong YS, Tan KC (2020) Solving dynamic multiobjective problem via autoencoding evolutionary search. *IEEE Trans Cybern*. <https://doi.org/10.1109/TCYB.2020.3017017>
 36. Zheng Y, Zhu Z, Qi Y, Wang L, Ma X (2020) Multi-objective multifactorial evolutionary algorithm enhanced with the weighting helper-task. In: 2nd Int. Conf. Ind. Artif. Intell. IAI. <https://doi.org/10.1109/IAI50351.2020.9262200>.
 37. Yang C, Ding J, Tan KC, Jin Y (2017) Two-stage assortative mating for multi-objective multifactorial evolutionary optimization. In: 2017 IEEE 56th annu. conf. decis. control. <https://doi.org/10.1109/CDC.2017.8263646>
 38. Binh HTT, Tuan NQ, Long DCT (2019) A multi-objective multi-factorial evolutionary algorithm with reference-point-based approach. In: Proc. IEEE Congr. Evol. Comput. <https://doi.org/10.1109/CEC.2019.8790034>
 39. Lin J, Liu HL, Xue B, Zhang M, Gu F (2019) Multiobjective multitasking optimization based on incremental learning. *IEEE Trans Evol Comput* 24(5):824–838. <https://doi.org/10.1109/TEVC.2019.2962747>
 40. Liang Z, Zhang J, Feng L, Zhu Z (2019) A hybrid of genetic transform and hyper-rectangle search strategies for evolutionary multi-tasking. *Expert Syst Appl*. <https://doi.org/10.1016/j.eswa.2019.07.015>
 41. Feng L, Zhou L, Zhong JH et al (2019) Evolutionary multitasking via explicit autoencoding. *IEEE Trans Cybern* 49(9):3457–3470. <https://doi.org/10.1109/TCYB.2018.2845361>
 42. Li JY, Du KJ, Zhan ZH, Wang H, Zhang J (2021) Multi-criteria differential evolution: treating multitask optimization as multi-criteria optimization. In: Proc. genet. evol. comput. conf. <https://doi.org/10.1145/3449726.3459456>.
 43. Zhan ZH, Li J, Cao J, Zhang J, Chung HS, Shi Y (2013) Multiple populations for multiple objectives: a coevolutionary technique for solving multiobjective optimization problems. *IEEE Trans Cybern* 43(2):445–463. <https://doi.org/10.1109/TSMCB.2012.2209115>
 44. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197. <https://doi.org/10.1109/4235.996017>
 45. Wang H, Jin Y, Doherty J (2018) A generic test suite for evolutionary multifidelity optimization. *IEEE Trans Evol Comput* 22(6):836–850. <https://doi.org/10.1109/TEVC.2017.2758360>
 46. Wu SH, Zhan ZH, Zhang J (2021) SAFE: scale-adaptive fitness evaluation method for expensive optimization problems. *IEEE Trans Evol Comput* 25(3):478–491. <https://doi.org/10.1109/TEVC.2021.3051608>
 47. Zhou ZH, Yang Y, Chao Q (2019) *Evolutionary learning: advances in theories and algorithms*. Springer, Singapore
 48. Huang H, Su J, Zhang Y, Hao Z (2020) An experimental method to estimate running time of evolutionary algorithms for continuous optimization. *IEEE Trans Evol Comput* 24(2):275–289. <https://doi.org/10.1109/TEVC.2019.2921547>
 49. Lin Q, Lin W, Zhu Z, Gong M, Li J, Coello CAC (2020) Multimodal multiobjective evolutionary optimization with dual clustering in decision and objective spaces. *IEEE Trans Evol Comput* 25(1):130–144. <https://doi.org/10.1109/TEVC.2020.3008822>
 50. Liu S, Lin Q, Tan KC, Gong M, Coello CAC (2020) A fuzzy decomposition-based multi/many-objective evolutionary algorithm. *IEEE Trans Cybern*. <https://doi.org/10.1109/TCYB.2020.3008697>
 51. Liu S, Lin Q, Wong KC et al (2020) A self-guided reference vector strategy for many-objective optimization. *IEEE Trans Cybern*. <https://doi.org/10.1109/TCYB.2020.2971638>
 52. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans Evol Comput* 3(4):257–271
 53. Yuan Y, Ong YS, Feng L et al (2017) Evolutionary multitasking for multi-objective continuous optimization: benchmark problems, performance metrics and baseline results. In: Technical Report. DIALOG. <https://arxiv.org/abs/1706.02766>
 54. Jin Y, Wang H, Sun C (2021) *Data-driven evolutionary optimization*. Springer, Cham
 55. Tian J, Tan Y, Zeng J, Sun C, Jin Y (2019) Multiobjective infill criterion driven gaussian process-assisted particle swarm optimization of high-dimensional expensive problems. *IEEE Trans Evol Comput* 23(3):459–472. <https://doi.org/10.1109/TEVC.2018.2869247>
 56. Lin Q, Wu X, Ma L, Li J, Gong M, Coello CAC (2021) An ensemble surrogate-based framework for expensive multiobjective evolutionary optimization. *IEEE Trans Evol Comput*. <https://doi.org/10.1109/TEVC.2021.3103936>
 57. Song Z, Wang H, He C, Jin Y (2021) A Kriging-assisted two-archive evolutionary algorithm for expensive many-objective optimization. *IEEE Trans Evol Comput*. <https://doi.org/10.1109/TEVC.2021.3073648>
 58. Wang H, Feng L, Jin Y, Doherty J (2021) Surrogate-assisted evolutionary multitasking for expensive minimax optimization in multiple scenarios. *IEEE Comput Intell Mag* 16(1):34–48. <https://doi.org/10.1109/MCI.2020.3039067>
 59. Lin Q, Lin W, Zhu Z, Gong M, Li J, Coello CAC (2021) Multimodal multiobjective evolutionary optimization with dual clustering in decision and objective spaces. *IEEE Trans Evol Comput* 25(1):130–144. <https://doi.org/10.1109/TEVC.2020.3008822>
 60. Liu S, Lin Q, Tian Y, Tan KC (2021) A variable importance-based differential evolution for large-scale multiobjective optimization. *IEEE Trans Cybern*. <https://doi.org/10.1109/TCYB.2021.3098186>
 61. Wang ZJ, Zhan ZH, Kwong S, Jin H, Zhang J (2021) Adaptive granularity learning distributed particle swarm optimization for large-scale optimization. *IEEE Trans Cybern* 51(3):1175–1188. <https://doi.org/10.1109/TCYB.2020.2977956>
 62. Liu SC, Zhan ZH, Tan KC, Zhang J (2021) A multiobjective framework for many-objective optimization. *IEEE Trans Cybern*. <https://doi.org/10.1109/tcyb.2021.3082200>
 63. Liu XF, Zhan ZH, Gao Y, Zhang J, Kwong S, Zhang J (2019) Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization. *IEEE Trans Evol Comput* 23(4):587–602. <https://doi.org/10.1109/TEVC.2018.2875430>