Applied Soft Computing 136 (2023) 110070

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Aligning heterogeneous optimization problems with optimal correspondence assisted affine transformation for evolutionary multi-tasking

An Chen [*], Zhigang Ren, Muyi Wang, Shenyu Su, Jiaqi Yun, Yichuang Wang

School of Automation Science and Engineering, Xi'an Jiaotong University, Xi'an, China

ARTICLE INFO

Article history:
Received 5 January 2022
Received in revised form 16 October 2022
Accepted 25 January 2023
Available online 3 February 2023

Keywords:
Evolutionary multi-tasking
Intertask alignment
Affine transformation
Many-tasking optimization
Estimation of distribution algorithm

ABSTRACT

Evolutionary multi-tasking optimization (EMTO) aims to boost the overall efficiency of optimizing multiple tasks by triggering knowledge transfer among them. Unfortunately, it may suffer from negative transfer on heterogeneous composite tasks that have low similarity. Some studies try to learn an intertask alignment transformation based on the paired samples from the involved tasks, but risk a failed alignment with improper pairwise methods. To solve this issue, this study proposes an optimal correspondence assisted affine transformation (OCAT) algorithm. OCAT explicitly constructs a mathematical model for the intertask alignment problem and theoretically deduces its optimal solution in an iterative method. As a result, the sample correspondences that enable the learned transformation to achieve the maximum intertask similarity can be located. Besides, a novel approach to deriving the affine transformation formula is also developed for OCAT. The resulting affine alignment transformation will not impair the knowledge contained in the tasks during the alignment process. By integrating OCAT with the estimation of distribution algorithm, this study finally develops a many-tasking optimization algorithm named MaT-EDA, where the solutions from other tasks are explicitly transferred as the samples for estimating the current distribution model. Extensive simulation studies have indicated that OCAT can significantly enhance the performance of EMTO, and MaT-EDA also achieves impressive many-tasking optimization performance.

1. Introduction

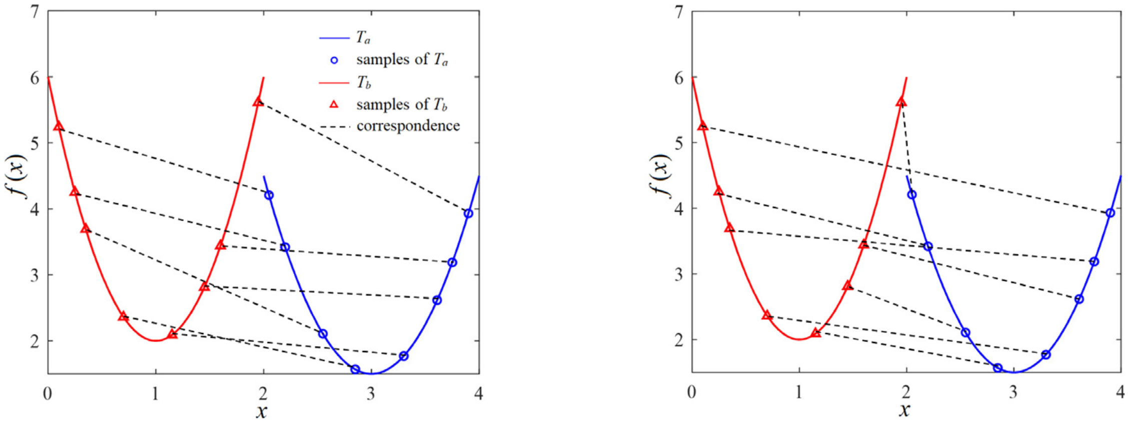
Evolutionary algorithms (EAs) are a kind of bio-inspired metaheuristic algorithms that imitate organism reproduction and natural selection. Due to the impressive global searching ability and derivative-free property, EAs have been successfully applied to various complex optimization problems over the past decades [1–3]. The federated learning management for image processing [4], the neural network training in reinforcement learning [5], and the proportional fairness problem [6] are several typical application examples in the real world.

Despite the enjoyed success, it is worth noting that conventional EAs generally optimize a single task at a time and conduct the search process from scratch. On the other hand, problems seldom exist in isolation in practice [7–10]. Given this fact, researchers recently developed a brand-new evolutionary paradigm, i.e., evolutionary multi-tasking optimization (EMTO) [11–14], which tackles multiple distinct optimization tasks simultaneously and conducts knowledge transfer among them to enhance the overall optimization performance. Existing EMTO algorithms can be mainly classified into implicit ones and explicit ones. The former, including the influential multifactorial evolutionary algorithm (MFEA) and its variants, employ one population to tackle all tasks in a unified search space [11,15–17]. Each individual is associated with a specific task via the skill factor, and the genetic materials are implicitly transferred through the crossover between the individuals with different skill factors. In contrast, explicit EMTO algorithms assign an exclusive population to each task and trigger knowledge transfer among tasks by migrating promising individuals [12,18–20]. By now, lots of studies have successfully applied EMTO algorithms to real-world multi-tasking optimization problems, including the hyperspectral endmember extraction [21] and the clustered minimum routing cost problem [22].

EMTO provides a promising solver for multi-tasking optimization problems, but its efficacy may degrade when the involved tasks are heterogeneous [23]. The reason lies in that there is little off-the-shelf similarity among these tasks and *negative transfer* is likely to occur. To alleviate this issue, some researchers tried to adapt the intensity of knowledge transfer according to the

[*] Correspondence to: School of Automation Science and Engineering, Xi'an Jiaotong University, No. 28 Xianning West Road, Xi'an Shaanxi, 710049, China.
E-mail address: chenan123@stu.xjtu.edu.cn (A. Chen).

https://doi.org/10.1016/j.asoc.2023.110070
1568-4946/© 2023 Elsevier B.V. All rights reserved.



(a) ideal correspondences

(b) correspondences based on fitness values

Fig. 1. An illustration of chaotic matching.

intertask similarity [16,18,19]. For example, Bali et al. proposed an improved MFEA, i.e., MFEA-II, where a low crossover probability will be specified for two tasks with explicitly discrepant landscapes [16]. Despite the success of these algorithms, the explicit similarity is just a sufficient but not necessary condition for triggering positive transfer [24,25]. It has been widely demonstrated that the transferability between two heterogeneous tasks can be also uncovered and exploited with a proper intertask alignment transformation [8–10]. Lately, this methodology has gained increasing research attention in the EMTO realm.

Existing intertask alignment algorithms for EMTO can be mainly classified into pairwise-learning-based ones and distribution-learning-based ones [14]. The former kind of method first pairs up the representative solution sets from the two tasks to be aligned and then learns an alignment transformation based on the paired samples [12,17,26,27]. For example, the linear domain adaption (LDA) algorithm pairs up the training samples based on their fitness rankings and then builds a linear alignment transformation [17]. With the same pairwise method, Feng et al. employed a single-layer autoencoder with bias term (SLA) to learn the intertask mapping [12]. In contrast, Zhou et al. [26] and Lim et al. [27] replaced SLA with a kernelized autoencoder and a two-layer neural network, respectively, to capture the nonlinear relationship between tasks. As a comparison, the distribution-learning-based alignment methods first estimate the task distributions and then derive a transformation to map them [28–32]. Xue et al. proposed an evolutionary-path-based affine transformation (EPAT) [29], which represents the tasks using two evolutionary-path-based Gaussian probabilistic models, respectively, and then employs affine transformation to bridge the model gap. Chen et al. modeled the task decision spaces as a joint manifold and learned their relationship by projecting the manifold into a common latent space [30]. In contrast, the subspace alignment (SA) algorithm first establishes a subspace for each task using the principal component analysis and then learns alignment transformation by minimizing the discrepancy between the two subspaces [31]. On the bedrock of SA, Gao et al. further aligned the current two task population distributions and also applied a decision variable transfer mechanism on the aligned space to enhance the positive knowledge transfer [32].

Despite the success achieved by the pairwise-learning-based alignment methods, it remains an open issue to configure proper correspondences between learning samples. Existing algorithms generally pair up the learning samples based on their fitness rankings by considering the rank correlation. This pairwise fashion, however, may cause a severe issue called chaotic matching and dramatically degrades the intertask alignment [29]. Fig. 1 graphically illustrates the chaotic matching issue on two one-dimensional tasks T_a and T_b to be aligned. As depicted in Fig. 1(a), their training samples are expected to be corresponded by taking the rank correlation and the topological consistency into account. However, the samples from different fitness landscape flanks can be matched in the light of their fitness rankings as shown in Fig. 1(b). It has been pointed out in [29] that these paired samples will result in a failed intertask alignment transformation. Unfortunately, due to the lack of topological information for a specific task, configuring proper correspondences between the training samples is nontrivial.

In fact, the pairwise-learning-based intertask alignment problem is quite similar to the point set registration one [33–35], which also requires to search for the correct correspondences between two point sets and then conduct a space geometry transformation to register them. Inspired by the influential registration algorithm iterative closest point (ICP) [33], this proposes a novel pairwise-learning-based intertask alignment algorithm named optimal correspondence assisted affine transformation (OCAT). OCAT describes the relationship between any two involved tasks using affine transformation. It first constructs a mathematical model for the intertask alignment problem, and then theoretically deduces the model solution by iteratively rebuilding the correspondences between the transformed samples and deriving the new alignment affine transformation. Accordingly, the correspondences that enable the affine transformation to achieve maximum intertask alignment can be located finally. OCAT notices that for the given samples and their correspondences, there are generally infinite affine alignment transformations, but some of them may dramatically impair the knowledge contained in the task during the alignment process. Purposefully, it imposes a constraint on the transformation parameters according to the task domain space sizes and then derives a closed-form solution of affine transformation, ensuring a smooth knowledge transfer.

This study finally develops an explicit many-tasking estimation distribution algorithm (MaT-EDA) by integrating OCAT with EDA. As model-based EAs, EDAs show impressive convergence speed over other EAs by using statistical learning methods to estimate the distribution of promising solutions [36–38]. However, conventional EDAs usually require a large population to supply enough samples for distribution modeling, which sharply limits their performance. On the other hand, a many-tasking optimization problem generally involves more than ten tasks. The population demand of each task can be alleviated by transferring the solutions for distribution modeling among the composite tasks. Following this idea, MaT-EDA explicitly migrates some

Table 1
A list of the abbreviations in this paper.

Abbreviation	Full name
BoKT	Bi-objective Knowledge Transfer
EA	Evolutionary Algorithm
EMaTO-MKT	Evolutionary Many-Task Optimization Algorithm based on Multi-Source Knowledge Transfer
EMTO	Evolutionary Multi-tasking Optimization
EPAT	Evolutionary-Path-based Affine Transformation
FE	Fitness Evaluation
ICP	Iterative Closest Point
IGD	Inverted Generational Distance
KAE	Kernelized AutoEncoder
LDA	Linear Domain Adaption
MaT-EDA	Many-Tasking Estimation Distribution Algorithm
MaTDE	Many-Tasking Differential Evolution
MFEA	Multifactorial Evolutionary Algorithm
MOMFEA	Multiple-Objective MFEA
NSGA-II	Non-Dominated Sorting Genetic Algorithm Ii
OCAT	Optimal Correspondence assisted Affine Transformation
SA	Subspace alignment
SLA	Single-Layer Autoencoder
SOMFEA	Single-Objective MFEA
SSR	Sum of Squared Residuals

promising solutions from the other tasks to the current one through OCAT and then estimates the distribution model accordingly. As a result, it only requires a small-scale population on each task, facilitating more evolution generations to locate better solutions.

To evaluate the effectiveness of OCAT and MaT-EDA, comprehensive simulation tests have been conducted on several multi-tasking and many-tasking optimization benchmark suites. The results indicate that OCAT can significantly enhance EMTO and also performs superior over some state-of-the-art intertask alignment algorithms. Moreover, MaT-EDA achieves very competitive many-tasking optimization performance. In summary, the main contributions of this study are as follows:

(1) A novel method to locate the optimal correspondences between training samples is developed. It can help the pairwise-learning-based methods achieve proper intertask alignment.
(2) A new derivation method for affine transformation is proposed. It generates an affine alignment transformation with small influence on the knowledge contained in the tasks during the alignment process.
(3) An explicit many-tasking optimization framework is designed. It enables EDA to effectively solve many-tasking optimization problems effectively.
(4) Empirical studies were conducted to investigate the performance of OCAT and MaT-EDA. A thorough analysis of the empirical results is also presented.

The remainder of this paper proceeds as follows: Section 2 reviews the related works. Section 3 describes the proposed algorithm in detail. Section 4 reports experimental settings and results. Finally, Section 5 concludes this paper and discusses some future research directions. Table 1 lists all the abbreviations and their full names in this paper.

2. Related works

In this section, we first give a general introduction of evolutionary multi-tasking optimization. Next, comprehensive reviews of the existing intertask alignment and many-tasking optimization algorithms are conducted in Sections 2.2 and 2.3, respectively.

2.1. Evolutionary multi-tasking optimization

The general framework of EMTO is depicted by Fig. 2. Through the knowledge transfer among the composite optimization tasks, EMTO tends to produce more promising individuals for each of them and then expedite the overall optimization efficiency of the multi-tasking optimization problem [13,14]. Taking the single-objective minimization as an example, a multi-tasking optimization model can be formulized as follows:

$$\begin{cases} \mathbf{x}_1^* = \arg\min_{\mathbf{x}_1} f_1(\mathbf{x}_1), \quad \mathbf{x}_1 \in [\mathbf{lb}_1, \mathbf{ub}_1]^{d_1} \\ \mathbf{x}_2^* = \arg\min_{\mathbf{x}_2} f_2(\mathbf{x}_2), \quad \mathbf{x}_2 \in [\mathbf{lb}_2, \mathbf{ub}_2]^{d_2} \\ \dots \\ \mathbf{x}_m^* = \arg\min_{\mathbf{x}_m} f_m(\mathbf{x}_m), \quad \mathbf{x}_m \in [\mathbf{lb}_m, \mathbf{ub}_m]^{d_m} \end{cases}, \tag{1}$$

where m is the number of tasks, $f_i(\mathbf{x}_i)$ $(i = 1, 2, \dots, m)$ is the objective function of the ith task T_i, and $[\mathbf{lb}_i, \mathbf{ub}_i]^{d_i}$ denotes the corresponding d_i-dimensional decision space with $\mathbf{lb}_i$ and $\mathbf{ub}_i$ being the lower bound and upper bound, respectively. The problem described by Eq. (1) is called a single-objective multi-tasking optimization problem. It is notable that each task can be also multiple-objective and the corresponding multi-tasking problem becomes a multiple-objective one. Besides, the problem is technically called a many-tasking optimization problem when $m > 10$. Generally, the two tasks T_a and T_b with $[\mathbf{lb}_a, \mathbf{ub}_a]^{d_a} \simeq [\mathbf{lb}_b, \mathbf{ub}_b]^{d_b}$ and $f_a^r \simeq f_b^r$ are called homogeneous tasks, where $\simeq$ denotes a similar relationship and f_a^r is the abstract rank-based fitness space. Otherwise, they are heterogeneous. The homogeneous tasks can be tackled smoothly and simultaneously since the promising solution exchange between them can naturally lead to their solution quality improvement. On the contrary, the rigid knowledge transfer between two heterogeneous tasks may be detrimental to problem solving.

One of the pioneering EMTO algorithms is the influential MFEA [11] developed from the genetic algorithm (GA), whose general procedure is presented in Algorithm 1. MFEA first initializes a population in a unified representation space for all tasks (line 1). According to the performance on each task, each population individual is assigned with a *skill factor*, i.e., the index of the task on which the individual performs best. During the evolutionary process, the knowledge transfer is triggered via two approaches, i.e., assortative mating (line 10) and vertical cultural transmission (line 11). The former allows a crossover between two individuals with different skill factors under a probability of *rmp*, while the latter encourages the generated offspring to imitate the phenotype of parents. In this way, the genetic materials are implicitly transferred between different tasks. Inspired by the success of MFEA, the proposers further integrated the classic multiple-objective non-dominated sorting genetic algorithm II (NSGA-II) [39] into the multifactorial framework and developed a multiple-objective multi-tasking MFEA [15]. Due to the ingeniousness of MFEA, many follow-up studies have been conducted. For example, MFEA-II [16] first estimates a transfer probability matrix RMP online by maximizing the log-likelihood of multiple mixture distribution representations. The higher relatedness the two tasks possess, the larger the corresponding element in RMP will be, thereby boosting the positive transfer. In contrast, Zheng et al. [40] recorded the capability of an individual in tackling different component tasks by extending its skill factor into an ability vector. Based on the elements in the ability vector, each individual will be probabilistically selected by the corresponding task to form its private task group. As a result, the highly related tasks will share a high overlap in their task groups, facilitating intense knowledge exchange.

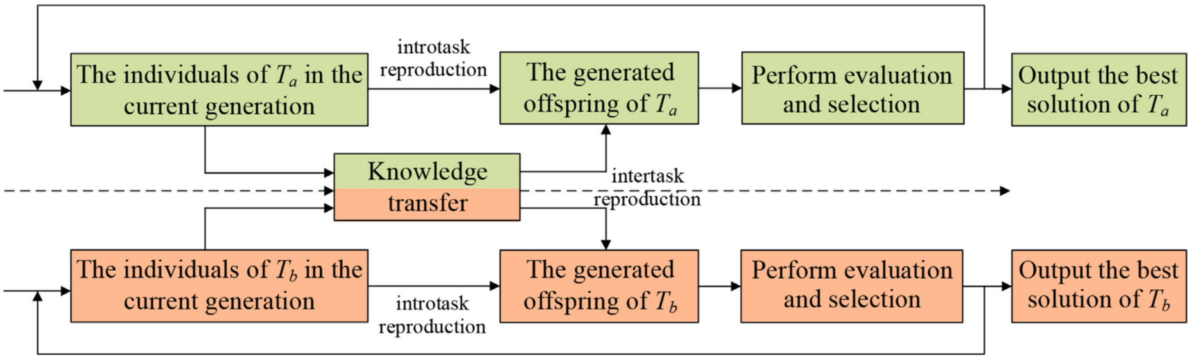

The individuals of T_a in the current generation
introtask reproduction
The generated offspring of T_a
Perform evaluation and selection
Output the best solution of T_a
Knowledge transfer
intertask reproduction
The individuals of T_b in the current generation
introtask reproduction
The generated offspring of T_b
Perform evaluation and selection
Output the best solution of T_b

Fig. 2. The general framework of EMTO.

```
Algorithm 1: General Procedure of MFEA
Randomly initialize a population P in the unified space;
Evaluate each individual in P on all tasks and calculate its skill factor $\tau$;
while there are fitness evaluations remained
    Empty offspring O: $\mathbf{O} \leftarrow \varnothing$;
    for each two random parents $\mathbf{p}_1$ and $\mathbf{p}_2$ in P
        if $\tau_1 == \tau_2$
            Perform intratask crossover: $(\mathbf{o}_1, \mathbf{o}_2) \leftarrow \text{crossover}(\mathbf{p}_1, \mathbf{p}_2)$;
            Assign $\mathbf{o}_1$ and $\mathbf{o}_2$ with skill factor $\tau_1$;
        else if $rand < rmp$
            Perform intertask crossover: $(\mathbf{o}_1, \mathbf{o}_2) \leftarrow \text{crossover}(\mathbf{p}_1, \mathbf{p}_2)$;
            Each offspring is randomly assigned skill factor $\tau_1$ or $\tau_2$;
        else
            Perform mutation: $\mathbf{o}_1 \leftarrow \text{mutate}(\mathbf{p}_1)$ and assign $\mathbf{o}_1$ with skill factor $\tau_1$;
            Perform mutation: $\mathbf{o}_2 \leftarrow \text{mutate}(\mathbf{p}_2)$ and assign $\mathbf{o}_2$ with skill factor $\tau_2$;
        Evaluate $\mathbf{o}_1$ and $\mathbf{o}_2$ on their associated tasks, respectively;
        Update O: $\mathbf{O} \leftarrow \mathbf{o}_1, \mathbf{o}_2$;
    Update the population P with O;
return the best individuals for each task in P.
```

In addition to the implicit single-population EMTO framework, some researchers also developed a multi-population one, in which the promising individuals are explicitly transferred from the assisted task population to the target one. For example, Feng et al. proposed an explicit EMTO algorithm [12], where elite individuals are exchanged across task populations with a fixed generation interval in the evolutionary process. There are also some other modalities of knowledge transfer. Da et al. shared the structure information among the component tasks [23]. They first estimated the distribution of each assisted population using a probabilistic model and then employed the models to guide the search on the target task. Lin et al. proposed an effective knowledge transfer approach for multi-objective multi-tasking optimization [41]. If a transferred individual is nondominated in the target task, its neighbors will be selected as the transferred solutions in the next generation since they are more likely to achieve positive transfer.

2.2. Intertask alignment algorithms for EMTO

The EMTO algorithms discussed above are mainly designed for the multi-tasking problem involving homogeneous optimization tasks. In fact, the latent synergies between two heterogeneous tasks can be also uncovered and exploited as long as a proper intertask alignment transformation can be achieved [9,10]. So far, two types of methods have been developed to learn the transformation, namely, the pairwise-learning-based ones and the distribution-learning-based ones.

LDA is the first attempt at the intertask alignment technique for EMTO [17]. It first pairs up some representative individuals in the course of evolution from the two tasks to be aligned based on their fitness rankings, and then derives a linear intertask transformation accordingly. With the same pairwise method, Feng et al. [12] randomly and independently sampled two solution sets from the tasks and regarded them as the input and output, respectively, to train an SLA, which will then be employed as the alignment transformation. In fact, SLA can be considered an affine transformation integrated with contractive and translation transformations and thus is linear. Afterwards, Zhou et al. [26] proposed a kernelized autoencoder (KAE) algorithm to align heterogeneous tasks. KAE first proposes to train the SLA between the ranked contemporary task populations in a reproducing kernel Hilbert space to capture the intertask nonlinearity. Then instead of solely using the kernelized SLA or the linear SLA, it adaptively determines which one to select based on the similarity of the two solution sets. By this means, the superiorities of both kinds of SLAs can be fully exploited. In contrast, Lim et al. [27] employed a two-layer neural network to learn the nonlinear intertask alignment based on the paired populations. Despite the success achieved by these pairwise-learning-based alignment algorithms, the commonly adopted fitness-ranking-based methods will result in the issue of chaotic matching.

In addition, there are some other methods aligning heterogeneous tasks based on the distributions estimated for them. A considerable number of studies directly align the central locations of the corresponding individual sets and apply well to the optimization tasks with different optima [28,42,43]. However, they tend to lose efficiency on the ones with distinct structural characteristics. In contrast, EPAT first represents the heterogeneous tasks with two evolutionary-path-based Gaussian models. Then according to a rank loss function that aims to minimize the intertask mismatch, EPAT derives a closed-form solution of affine transformation to bridge the model gap [29]. Chen et al. [30] drew inspiration from the manifold alignment technique and projected the decision spaces of the composite tasks into a common latent space by solving a generalized eigenvalue decomposition problem. The intertask similarity can thus be enhanced with the projection matrix. As a comparison, SA [31] applies the principal component analysis to extract the subspace of each task based on its current population and then derives the intertask alignment transformation by aligning the bases of the corresponding subspaces. After that, Gao et al. [32] proposed a DAVT algorithm that involves the distribution alignment strategy and the decision variable transfer mechanism. The DA strategy aligns the covariance matrices of the current two task populations as well as the subspace bases, while the VT mechanism is to further enhance the positive knowledge transfer by removing the small bias between the centers of the aligned populations.

From the above descriptions, we can observe that the pairwise-learning-based alignment is straightforward by intuitively learning the intertask transformation based on the paired samples. However, it remains an open issue to configure proper correspondences between the sample sets. As a comparison, the distribution-learning-based alignment avoids this dilemma by mapping the task distributions, but estimating a reliable distribution for the task is nontrivial. Given these observations, this study proposes the OCAT algorithm. It can effectively locate the correspondences between training samples that facilitate an appropriate intertask alignment transformation.

2.3. Many-tasking optimization algorithms

With the deepening of the research on EMTO, there are some emerging studies developed for many-tasking optimization problems. The main challenge in many-tasking optimization is that only a minority of composite tasks are similar [18]. The preliminary studies identify an appropriate assisted task for the current target one according to the historical performance. For example, Liaw and Ting [44] selected the task that gains the most frequent positive transfer on the target one as the next assisted task. Thanh et al. [45] employed the multi-armed bandit technique and chose the assisted task based on the reward feedback of the action. Although these methods have achieved some successes in solving many-tasking optimization problems, they ignore the intertask similarity and thus may cause negative transfer.

Chen et al. alleviated this issue by proposing an archive-based many-tasking differential evolution (MaTDE) algorithm [18]. It first measures the similarity between the task populations according to the Kullback–Leibler divergence and then synthetically considers the intertask similarity and the accumulated transfer rewards when assigning the assisted task. Liang et al. developed an evolutionary many-task optimization algorithm based on a multi-source knowledge transfer mechanism (EMaTO-MKT) [19], which first adaptively calculates the probability of knowledge transfer on the target task according to its current evolution trend. When the knowledge transfer is triggered, EMaTO-MKT selects multiple assisted tasks using the maximum mean discrepancy measure and generates the offspring by sampling from the distribution model estimated based on the individuals from the target task and the assisted ones. More recently, Jiang et al. [46] tried to measure the intertask similarity from the perspectives of the fitness shape and the global optimal domain. Then based on the type of intertask similarity, they further configured the most suitable knowledge transfer strategy for the involved tasks. By this means, the developed bi-objective knowledge transfer (BoKT) framework achieves impressive many-tasking optimization performance.

Despite the differences among the above algorithms, they possess the same motivation of decreasing the probability of negative transfer in many-tasking optimization. On the other hand, the intertask alignment methods can smoothly alleviate the negative transfer issue. It is thus promising to exploit the intertask alignment method to enhance the many-tasking optimization. With this consideration in mind, this study develops the many-tasking optimization algorithm MaT-EDA with the help of the proposed OCAT method.

3. Proposed algorithm

In this section, we successively elaborate on the proposed OCAT and MaT-EDA methods. The general framework of OCAT is first detailed in Section 3.1, where the location of the optimal correspondences between training samples is provided. Next, the new derivation method of affine transformation is presented in Section 3.2. Then the overall implementation of OCAT and its time complexity analysis are given in Section 3.3. Finally, the whole process of MaT-EDA is described in Section 3.4.

3.1. General framework of OCAT

For the pairwise-learning-based intertask alignment, building the correspondences between the training samples and deriving the specific transformation formula are the two key issues, and the former plays a fundamental role. Existing algorithms directly pair up the training samples based on their fitness values, but this pairwise fashion may prevent the resulting transformation from getting proper alignment for the involved two task landscapes. Inspired by the idea of ICP in point set registration [33], this study proposes an optimal correspondence assisted affine transformation (OCAT) algorithm, which can locate the correspondences between the task samples that enable the transformation to achieve maximized intertask structural similarity.

Suppose that the two tasks T_a and T_b need to be aligned, and $\mathbf{P}_a = \{\mathbf{p}_1^a, \ldots, \mathbf{p}_{NP}^a\}$ and $\mathbf{P}_b = \{\mathbf{p}_1^b, \ldots, \mathbf{p}_{NP}^b\}$ are their current populations, respectively. If $d_a < d_b$ (or $d_a > d_b$), we will pad $\mathbf{p}_i^a$ (or $\mathbf{p}_i^b$), $i = 1, \ldots, NP$, with $d_b - d_a$ (or $d_a - d_b$) random values between the corresponding lower bounds and upper bounds of T_b (or T_a). OCAT first separately extracts some promising individuals from $\mathbf{P}_a$ and $\mathbf{P}_b$ and employs them to learn the intertask transformation. The motivation for this operation lies in that the promising solutions contain the main characteristics of the corresponding fitness landscape and thus facilitate a more reliable transformation. Let $\mathbf{E}_a = \{\mathbf{p}_{(1)}^a, \ldots, \mathbf{p}_{(NS)}^a\}$ and $\mathbf{E}_b = \{\mathbf{p}_{(1)}^b, \ldots, \mathbf{p}_{(NS)}^b\}$ denote the two extracted individual sets, respectively, then the problem of intertask alignment can be formulated as:

$$\begin{aligned}&\max_{\boldsymbol{C}_{ab},\boldsymbol{M}_{ab}} J(\boldsymbol{C}_{ab}(\mathbf{E}_a), \boldsymbol{M}_{ab}(\mathbf{E}_a))\\ &s.t. \boldsymbol{C}_{ab}(\mathbf{p}_{(i)}^a) \neq \boldsymbol{C}_{ab}(\mathbf{p}_{(j)}^a),\ i, j \in \{1, \ldots, NS\} \text{ and } i \neq j\end{aligned}, \tag{2}$$

where $\boldsymbol{C}_{ab} : \mathbf{E}_a \rightarrow \mathbf{E}_b$ denotes the individual correspondence between $\mathbf{E}_a$ and $\mathbf{E}_b$, $\boldsymbol{M}_{ab}$ is the transformation to be derived, and J is the criterion for measuring the similarity between $\boldsymbol{C}_{ab}(\mathbf{E}_a)$ and $\boldsymbol{M}_{ab}(\mathbf{E}_a)$. In this study, we set J to the sum of squared residuals (SSR) and employ affine transformation as the basic transformation strategy. As a result, Eq. (2) becomes

$$\begin{aligned}&\min_{\mathbf{A}_{ab},\mathbf{t}_{ab},\boldsymbol{c}_{ab}(i)\in\{1,\ldots,NS\}} \sum_{i=1}^{NS} \|\mathbf{p}_{(\boldsymbol{c}_{ab}(i))}^b - (\mathbf{A}_{ab} \cdot \mathbf{p}_{(i)}^a + \mathbf{t}_{ab})\|^2\\ &s.t. \boldsymbol{c}_{ab}(i) \neq \boldsymbol{c}_{ab}(j),\ i, j \in \{1, \ldots, NS\} \text{ and } i \neq j\end{aligned}, \tag{3}$$

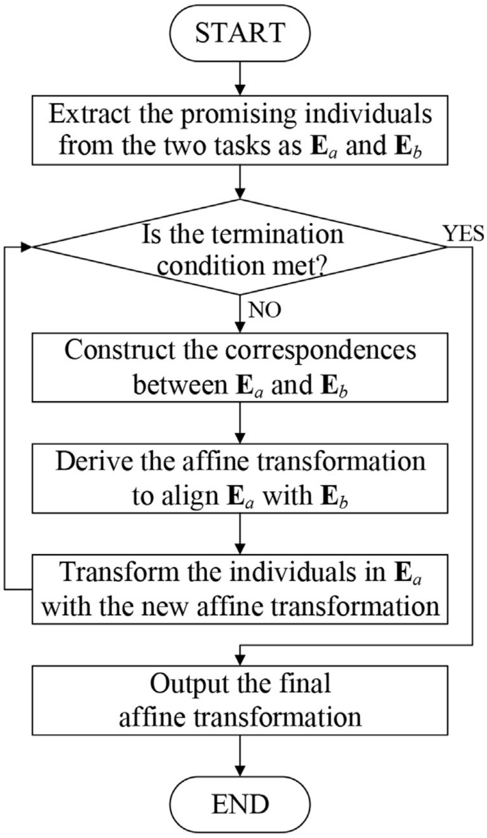

START
Extract the promising individuals from the two tasks as $\mathbf{E}_a$ and $\mathbf{E}_b$
Is the termination condition met?
YES
NO
Construct the correspondences between $\mathbf{E}_a$ and $\mathbf{E}_b$
Derive the affine transformation to align $\mathbf{E}_a$ with $\mathbf{E}_b$
Transform the individuals in $\mathbf{E}_a$ with the new affine transformation
Output the final affine transformation
END

Fig. 3. The overall flowchar of OCAT.

where "·" denotes the operator for matrix–matrix multiplication, $\mathbf{p}^b_{(\mathbf{c}_{ab}(i))}$ is the corresponding individual of $\mathbf{p}^a_{(i)}$ in $\mathbf{E}_b$, $\mathbf{A}_{ab}$ is a $D_m \times D_m$ ($D_m = \max(d_a, d_b)$) matrix, and $\mathbf{t}_{ab}$ is a $D_m \times 1$ translation vector. For such a minimization problem, it is hard to directly calculate the theoretical optima $\mathbf{A}^*_{ab}$, $\mathbf{t}^*_{ab}$, and $\mathbf{c}^*_{ab}(i)$, $i = 1, \ldots, NS$. To this end, OCAT solves Eq. (3) in an iterative method. Fig. 3 depicts the overall procedure of OCAT. In each iteration, it first updated the correspondences between the transformed $\mathbf{E}_a$ and $\mathbf{E}_b$ and then derives the new affine transformation to align them. More specifically, OCAT conducts the following two main steps in the kth iteration ($k \geq 1$):

Step 1. Set up the individual correspondence to minimize the SSR between $\mathbf{E}_a$ and $\mathbf{E}_b$, which means to find the individual in $\mathbf{E}_b$ that is closest to each transformed $\mathbf{p}^a_{(i)}$ in $\mathbf{E}_a$:

$$\mathbf{c}^k_{ab}(i) = \operatorname*{argmin}_{\mathbf{c}_{ab}(i)\in\{1,\ldots,NS\},\mathbf{c}_{ab}(i)\neq\mathbf{c}_{ab}(j)} \|\mathbf{p}^b_{(\mathbf{c}_{ab}(i))} - (\mathbf{A}^{k-1}_{ab} \cdot \mathbf{p}^a_{(i)} + \mathbf{t}^{k-1}_{ab})\|^2,$$
$$i, j \in \{1, \ldots, NS\} \text{ and } i \neq j, \quad (4)$$

where $\mathbf{A}^{k-1}_{ab}$ and $\mathbf{t}^{k-1}_{ab}$ are the transformation parameters derived in the $(k-1)$th iteration.

Step 2. Based on the updated correspondences, derive the new parameters $\mathbf{A}^k_{ab}$ and $\mathbf{t}^k_{ab}$:

$$(\mathbf{A}^k_{ab}, \mathbf{t}^k_{ab}) = \operatorname*{argmin}_{\mathbf{A}_{ab},\mathbf{t}_{ab}} \sum_{i=1}^{NS} \|\mathbf{p}^b_{(\mathbf{c}^k_{ab}(i))} - (\mathbf{A}_{ab} \cdot \mathbf{p}^a_{(i)} + \mathbf{t}_{ab})\|^2. \quad (5)$$

With the increase of k, $\mathbf{c}^k_{ab}(i)$, $\mathbf{A}^k_{ab}$, and $\mathbf{t}^k_{ab}$ will consistently approach $\mathbf{c}^*_{ab}(i)$, $\mathbf{A}^*_{ab}$, and $\mathbf{t}^*_{ab}$, respectively. The final $\mathbf{A}^k_{ab}$ and $\mathbf{t}^k_{ab}$ thus promote minimizing the discrepancy between T_a and T_b. Theorem 1 illustrates the convergence of OCAT.

Theorem 1. *For any two individual sets* $\mathbf{E}_a = \{\mathbf{p}^a_{(1)}, \ldots, \mathbf{p}^a_{(NS)}\}$ *and* $\mathbf{E}_b = \{\mathbf{p}^b_{(1)}, \ldots, \mathbf{p}^b_{(NS)}\}$*, OCAT must converge on the optimal solution of Eq.* (3)*.*

Proof. Please see the proof given in Appendix A.1.

From the above descriptions, we can know that how to solve Eqs. (4) and (5) is the key to OCAT. Eq. (4) can be easily solved. As for Eq. (5), it generally involves multiple optima, some of which may result in unsatisfying intertask transformations. Next, we will discuss this issue in detail and further provide an effective method to derive $\mathbf{A}^k_{ab}$ and $\mathbf{t}^k_{ab}$ that can ensure the quality of the obtained affine transformation.

3.2. Derivation of affine transformation

Due to the free constraint on $\mathbf{A}_{ab}$, the minimization problem described by Eq. (5) is ill-posed. That is to say Eq. (5) has more than one optimum. On the other side, not all the transformations formed by these solutions are qualified. For example, ($\mathbf{A}_{ab} = \mathbf{0}$, $\mathbf{t}_{ab} = \sum_{i=1}^{NS} \mathbf{p}^b_{(\mathbf{c}^k_{ab}(i))}/NS$) makes the objective function in Eq. (5) zero, but the resulting affine transformation will project all the solutions from T_a into the center of $\mathbf{E}_b$, which dramatically impairs the knowledge transferred from T_a. Therefore, some constraints should be imposed on $\mathbf{A}_{ab}$ to ensure an effective affine transformation.

By conducting the singular value decomposition on $\mathbf{A}_{ab}$, we can obtain $\mathbf{A}_{ab} = \mathbf{R}_{ab} \cdot \mathbf{S}_{ab} \cdot \mathbf{O}_{ab}$, where $\mathbf{R}_{ab}$ and $\mathbf{O}_{ab}$ are orthogonal matrices and $\mathbf{S}_{ab}$ is a diagonal one, i.e., $\mathbf{S}_{ab} = diag(s_1, \ldots, s_{dm})$. Among them, $\mathbf{R}_{ab}$ and $\mathbf{O}_{ab}$ conduct the rotation transformation, while $\mathbf{S}_{ab}$ performs the contractive transformation. It can be seen that $\det(\mathbf{A}_{ab}) = \det(\mathbf{R}_{ab}) \cdot \det(\mathbf{S}_{ab}) \cdot \det(\mathbf{O}_{ab}) = \det(\mathbf{S}_{ab})$. Therefore, the quality of the resulting affine transformation can be guaranteed by first properly configuring $\mathbf{S}_{ab}$. Considering the contractive effect of $\mathbf{S}_{ab}$, this study directly specifies it according to the domain space sizes of T_a and T_b, i.e.,

$$s_i = \frac{\mathbf{ub}_b(i) - \mathbf{lb}_b(i)}{\mathbf{ub}_a(i) - \mathbf{lb}_a(i)}, \; i = 1, \ldots, D_m. \quad (6)$$

The lower bounds and upper bounds of the padded dimensions in T_a (or T_b) are just the corresponding ones of T_b (or T_a). Besides, only one rotation matrix $\mathbf{R}_{ab}$ is maintained to facilitate the derivation. As such, Eq. (5) becomes:

$$(\mathbf{R}^k_{ab}, \mathbf{t}^k_{ab}) = \operatorname*{argmin}_{\mathbf{R}^T_{ab}\mathbf{R}_{ab}=\mathbf{I},\det(\mathbf{R}_{ab})=1,\mathbf{t}_{ab}} \sum_{i=1}^{NS} \|\mathbf{p}^b_{(\mathbf{c}^k_{ab}(i))} - (\mathbf{R}_{ab} \cdot \mathbf{q}^a_{(i)} + \mathbf{t}_{ab})\|^2, \quad (7)$$

where $\mathbf{q}^a_{(i)} = \mathbf{S}_{ab} \cdot \mathbf{p}^a_{(i)}$. Theorem 2 provides the final solutions of Eq. (7).

Theorem 2. *Suppose that* $\hat{\mathbf{p}}^b_{(\mathbf{c}^k_{ab}(i))} = \mathbf{p}^b_{(\mathbf{c}^k_{ab}(i))} - \sum_{i=1}^{NS} \mathbf{p}^b_{(\mathbf{c}^k_{ab}(i))}/NS$*,* $\hat{\mathbf{q}}^a_{(i)} = \mathbf{q}^a_{(i)} - \sum_{i=1}^{NS} \mathbf{q}^a_{(i)}/NS$*, and the singular value decomposition result on* $\sum_{i=1}^{NS} \hat{\mathbf{p}}^a_{(i)} \cdot \hat{\mathbf{p}}^b_{(\mathbf{c}^k_{ab}(i))}{}^T$ *is* $\mathbf{U} \cdot \mathbf{\Lambda} \cdot \mathbf{V}$*, then the optima of Eq.* (7) *are*

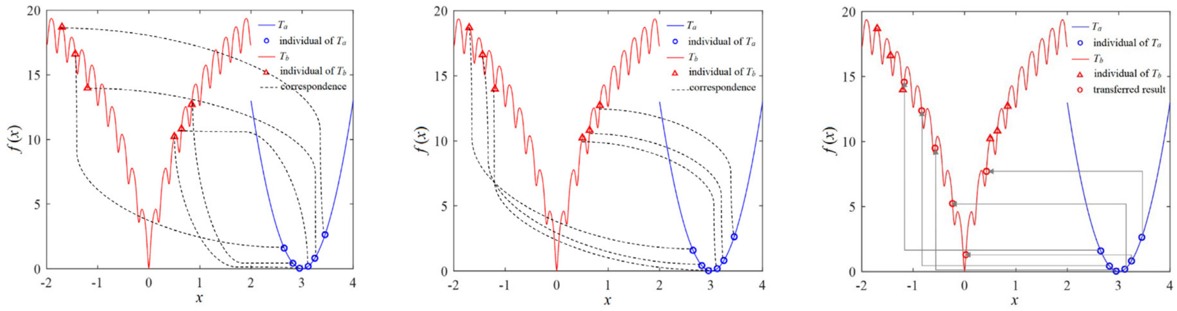
$$\mathbf{R}^k_{ab} = \mathbf{X} = \mathbf{V} \cdot \mathbf{U}^T \quad (8)$$

and

$$\mathbf{t}^k_{ab} = \frac{1}{NS} \sum_{i=1}^{NS} \mathbf{p}^b_{(\mathbf{c}^k_{ab}(i))} - \frac{1}{NS} \sum_{i=1}^{NS} \mathbf{V} \cdot \mathbf{U}^T \cdot \mathbf{q}^a_{(i)}. \quad (9)$$

Proof. Please see the proof given in Appendix A.2.

```
Algorithm 2: $(\mathbf{S}_{ab}, \mathbf{R}_{ab}, \mathbf{t}_{ab}) \leftarrow \mathrm{OCAT}(\mathbf{P}_a, \mathbf{P}_b)$
1.  Sort the individuals in $\mathbf{P}_a$ and $\mathbf{P}_b$ in ascending order based on their fitness values;
2.  Save the first $NS \leftarrow \lceil \rho \cdot |\mathbf{P}_a| \rceil$ individuals of $\mathbf{P}_a$ and $\mathbf{P}_b$ as $\mathbf{E}_a$ and $\mathbf{E}_b$, respectively;
3.  Perform initializations: $k \leftarrow 0$, $\mathbf{R}_{ab}^0 \leftarrow \mathbf{I}$, $\mathbf{t}_{ab}^0 \leftarrow \mathbf{0}$, $lastSSR \leftarrow \inf$, $curSSR \leftarrow 0$;
4.  Calculate the contractive matrix S according to Eq. (6);
5   while $curSSR < lastSSR$
6.      Update lastSSR: $lastSSR \leftarrow currSSR$;
7.      Set $k \leftarrow k+1$ and $\mathbf{E}_b^{'} \leftarrow \mathbf{E}_b$; // $\mathbf{E}_b^{'}$ is used to construct the individual correspondence
8.      for each individual $\mathbf{p}_{(i)}^a$ in $\mathbf{E}_a$
9.          Perform transformation: $\bar{\mathbf{p}}_{(i)}^a \leftarrow \mathbf{R}_{ab}^{k-1} \cdot \mathbf{S} \cdot \mathbf{p}_{(i)}^a + \mathbf{t}_{ab}^{k-1}$;
10.         Find the individual closest to $\bar{\mathbf{p}}_{(i)}^a$ in $\mathbf{E}_b^{'}$ and set its index to $\mathbf{c}_{ab}^k(i)$;
11.         Remove $\mathbf{p}_{(\mathbf{c}_{ab}^k(i))}^b$ from $\mathbf{E}_b^{'}$;
12.     Calculate the rotation matrix $\mathbf{R}_{ab}^k$ according to Eq. (8);
13.     Calculate the translation vector $\mathbf{t}_{ab}^k$ according to Eq. (9);
14.     Calculate the new SSR value: $currSSR \leftarrow \sum_{i=1}^{NS} \| \mathbf{p}_{(\mathbf{c}_{ab}^k(i))}^b - (\mathbf{R}_{ab}^k \cdot \mathbf{S} \cdot \mathbf{p}_{(i)}^a + \mathbf{t}_{ab}^k) \|^2$;
15. Set: $\mathbf{R}_{ab} \leftarrow \mathbf{R}_{ab}^k$ and $\mathbf{t}_{ab} \leftarrow \mathbf{t}_{ab}^k$;
16. return $\mathbf{S}_{ab}$, $\mathbf{R}_{ab}$, and $\mathbf{t}_{ab}$.
```



(a) The correspondence results based on fitness value (b) The correspondence results obtained by OCAT (c)The transformed results with OCAT

Fig. 4. A practical application of OCAT.

3.3. Overall implementation of OCAT

Algorithm 2 provides the whole pseudocode of OCAT. Lines 1–2 describe the generation of the promising individual sets $\mathbf{E}_a$ and $\mathbf{E}_b$. The parameter ρ is the selection ratio varying in (0, 1], which determines the quantity and quality of $\mathbf{E}_a$ and $\mathbf{E}_b$. We will experimentally investigate its influence in Section 4.3. Then after some initializations in line 3 and the calculation of the contractive matrix $\mathbf{S}_{ab}$ in line 4, lines 5–14 detail the iteration process of OCAT. During each iteration, the correspondences between the individuals in $\mathbf{E}_a$ and $\mathbf{E}_b$ are first built in lines 8–11. Afterwards, the parameters $\mathbf{R}_{ab}^k$ and $\mathbf{t}_{ab}^k$ are calculated in lines 12 and 13, respectively. The whole iteration process continues until the intertask SSR value remains unchanged. Finally, $\mathbf{S}_{ab}$ and the last $\mathbf{R}_{ab}^k$ and $\mathbf{t}_{ab}^k$ are output to generate the final intertask alignment transformation.

Fig. 4 presents a practical application of OCAT on two heterogeneous optimization tasks, i.e., the one-dimensional shifted Sphere function (T_a) and the shifted Rastrigin's one (T_b). Different from the fitness-value-based correspondence method that causes chaotic matching (shown in Fig. 4(a)), OCAT successfully locates the ideal correspondences between the individuals from the two composite tasks (shown in Fig. 4(b)). On that condition, the derived affine transformation enhances knowledgeable transfer from T_a to T_b, which helps explore the potential region of T_b effectively (shown in Fig. 4(c)).

The complexity of OCAT mainly comes from the construction of the correspondences between the sample sets and the derivation of affine transformation. Suppose that the whole number of iterations in OCAT is K, then the correspondence construction process has a time complexity of $O(K \cdot NS^2 \cdot D_m)$ in total. As for the derivation process, its complexity is mainly from the singular value decomposition operator, and thus the total time complexity is $O(K \cdot NS \cdot D_m^2)$. Therefore, the cumulative time complexity of OCAT is $O(K \cdot NS^2 \cdot D_m) + O(K \cdot NS \cdot D_m^2)$.

As an intertask alignment algorithm, OCAT can be seamlessly embedded into existing EMTO algorithms. Taking the classic MFEA shown in Algorithm 1 as an example, OCAT can be integrated with the intertask crossover process (Lines 10 and 11). Specifically, with the learned alignment transformation between T_{τ_1} and T_{τ_2}, the individual from T_{τ_1} is first transformed into the domain of T_{τ_2} before its crossover with the individual therein. The generated offspring will be also transformed back into T_{τ_1} if it is assigned the skill factor τ_1. As a result, the genetic materials between heterogeneous tasks are smoothly exchanged. Besides, OCAT can be also exploited to design fresh EMTO algorithms. This study attempts to combine it with EDA to solve many-tasking optimization problems.

3.4. EDA for many-tasking optimization

The main idea of EDA is to learn the distribution of an optimization problem using a Gaussian probability model and to generate offspring based on the model [36]. Concretely, it first initializes the population **P** with some randomly generated individuals. After the evaluation operation, the algorithm picks up a certain number of high-quality solutions from **P** to generate a training set **TS** for the probability model learning. For an n-dimensional Gaussian random vector $\mathbf{x}$, its probability model is as follows:

$$\boldsymbol{G}(\mathbf{x})=\frac{(2\pi)^{-n/2}}{\det(\mathbf{Cov})^{1/2}}\cdot\exp(-\frac{(\mathbf{x}-\mathbf{u})^T\cdot(\mathbf{Cov})^{-1}\cdot(\mathbf{x}-\mathbf{u})}{2}), \quad (10)$$

where $\mathbf{u}$ and **Cov** are the mean vector and the covariance matrix, respectively. EDA estimates $\mathbf{u}$ and **Cov** using the solutions in **TS** and generates offspring by sampling from $\boldsymbol{G}(\mathbf{x})$. This process iterates until all the fitness evaluations (FEs) are exhausted. With the captured problem structure characteristic, $\boldsymbol{G}(\mathbf{x})$ endows EDA with impressive convergence speed. Nevertheless, the basic EDA may suffer from premature convergence due to the rapid shrink of the search scope and the degrading search direction. To alleviate this issue, many EDA variants have been proposed. As a representative, AMaLGaM achieves significantly better convergence performance by developing an adaptive variance scaling strategy and an anticipated mean shift operator [37].

Despite the effectiveness of EDA and its variants, they generally require a large population. The purpose is to provide sufficient training samples to accurately estimate the covariance matrix, which involves $0.5(n^2+n)$ free parameters. Given a certain number of FEs, such a requirement necessarily reduces evolution generations, and thus limits the optimization performance. Fortunately, this dilemma can be potentially alleviated in explicit many-tasking optimization. Under the framework of many-tasking optimization, the individuals required by EDA for solving a target task can be transferred from assisted tasks. Accordingly, the population size for a single task can be significantly reduced. With this idea in mind, this study combines the developed OCAT algorithm with EDA and develops a many-tasking-oriented algorithm MaT-EDA.

The overall framework of MaT-EDA is shown in Fig. 5. The composite tasks are tacked in parallel with each assigned with an exclusive EDA, and they exchange information when generating the training set for probability model estimation. For each task T_i, MaT-EDA first employs OCAT to learn its alignment transformation $\boldsymbol{M}_{ji}$ with the task T_j, $j = 1, \ldots, m$ and $j \neq i$. Then the high-quality solutions from T_j are transformed with $\boldsymbol{M}_{ji}$ and added into the training set $\mathbf{TS}_i$ of T_i. By this means, the quality and quantity of samples in $\mathbf{TS}_i$ can be easily met, and we only need to configure a small population for T_i. With this mechanism, the more tasks the problem involves, the smaller the population size can be. Our preliminary experiments indicated that when the number of composite tasks m varies in $10 < m \leq 20$, $20 < m \leq 50$, or $50 \leq m$, MaT-EDA can achieve satisfying performance by setting the population size for each task to $NP/10$, $NP/20$, or $NP/40$, respectively, where NP is the population size of the corresponding single-task-oriented EDA.

Algorithm 3 presents the pseudocode of MaT-EDA. It can be observed that except for the generation of the training set for each task in lines 5–8, the main evolutionary process in MaT-EDA is identical to that in the conventional EDA. As such, this many-tasking optimization framework possesses good generalizability. It is worth noting that in addition to the basic EDA, some EDA variants like AMaLGaM can be also integrated into this framework. The reason is that the main difference among different EDAs lies in the estimation of the probability distribution model, which is independent of this many-tasking optimization framework. We will further illustrate this characteristic in Section 4.2. Finally, it should be admitted that due to the multiple runs of OCAT, MaT-EDA is of relatively high time complexity. Fortunately, the algorithm is also of high parallelism, which can be exploited to promote computation efficiency to a large extent.

Table 2
The property of each function set in the CEC2017 multi-tasking optimization benchmark suite.

Single-objective suite	Multiple-objective suite	Intersection degree	Ordinal correlation
S1	S1	complete	high
S2	S2	complete	middle
S3	S3	complete	low
S4	S4	partial	high
S5	S5	partial	middle
S6	S6	partial	low
S7	S7	no	high
S8	S8	no	middle
S9	S9	no	low

4. Simulation studies

The purpose of this section is three-fold[1]: (1) to investigate the effectiveness of OCAT; (2) to evaluate the performance of MaT-EDA; (3) to study the influence of the involved parameters.

4.1. Investigation of the effectiveness of OCAT

This section aims to investigate the effectiveness of the proposed OCAT algorithm by comparing it with five state-of-the-art intertask alignment algorithms, i.e., LDA [17], SLA [12], KAE [26], EPAT [29], and DAVT [32]. The main idea of each competitor has been described in Section 2.2. To this end, we separately embedded each intertask alignment algorithm into the same MFEA framework [11] and then compared the achieved multi-tasking optimization performance. For a comprehensive investigation, not only the single-objective MFEA (SOMFEA) framework [11] but also the multiple-objective one (MOMFEA) one [15] were employed, and the binding mode between the intertask alignment algorithm and the multifactorial framework is illustrated in Section 3.3. We represented the SOMFEA integrated with OCAT as SOMFEA$_{\text{OCAT}}$. Similar representations were given to other resulting multi-tasking optimization algorithms. To provide comparison baselines, the performance of the basic SOMFEA, the basic MOMFEA, and the two corresponding conventional EAs (GA and NSGA-II [39]) was also assessed. The widely-used CEC2017 multi-tasking optimization benchmark suite [47,48] was employed to evaluate each algorithm. This test suite contains nine single-objective function sets and nine multiple-objective ones, each of which involves two optimization tasks. Among these function sets, the optimal solution intersection degree of the composite tasks and the intertask ordinal correlation vary a lot. Table 2 briefly presents the properties of these function sets. More details about this benchmark suite can be found in [47,48].

As suggested by [47,48], the maximum numbers of FEs were set to 1×10^5 and 2×10^5 for the single-objective multi-tasking algorithm and the multiple-objective one, respectively. On each benchmark function set, we independently ran each algorithm 30 times and assessed its performance in terms of the median, mean, and standard deviation of the final obtained optimization results. For statistical analysis, we first used the Kruskal–Wallis nonparametric one-way ANOVA test [49] with a confidence interval of 0.95 to determine whether there is at least one method showing distinct optimization performance and then conducted a series of two-tailed Wilcoxon rank-sum tests at a significance level of 0.05 [49] in a pairwise fashion to compare the optimization results. During the simulation, the parameters involved in the two SOMFEA and MOMFEA frameworks were strictly set following

[1] The source codes of OCAT and MaT-EDA are available from URL https://github.com/chenanXJTU.

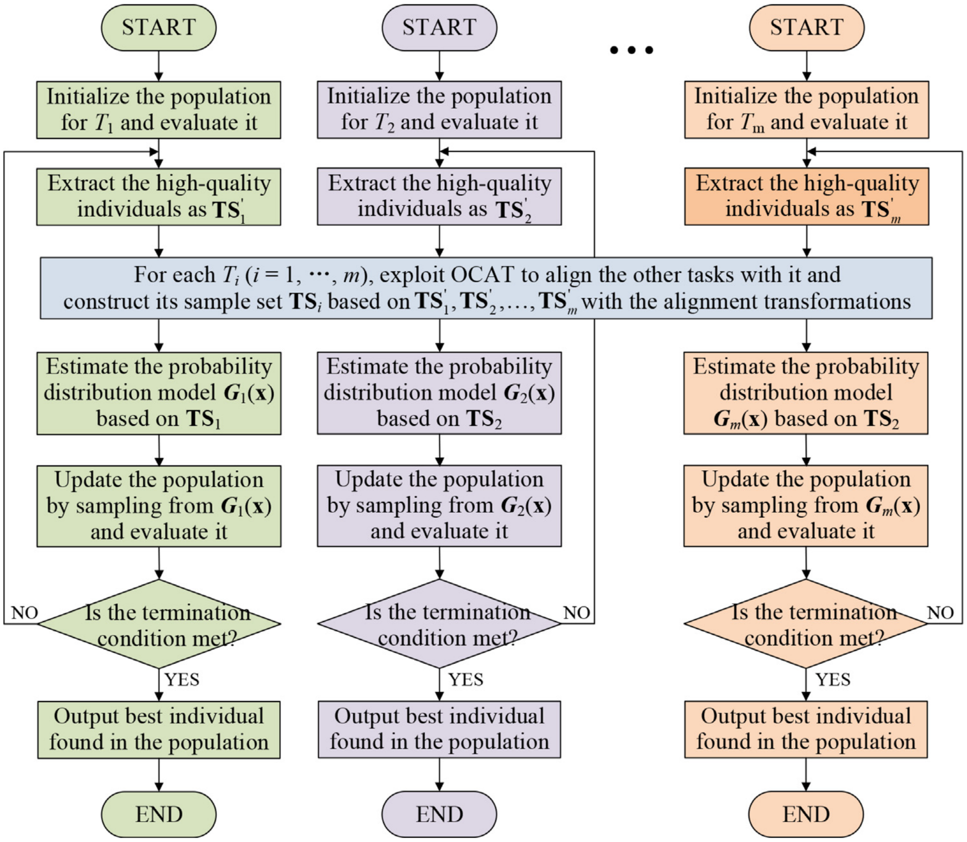

START
START
START
• • •
Initialize the population for T_1 and evaluate it
Initialize the population for T_2 and evaluate it
Initialize the population for T_m and evaluate it
Extract the high-quality individuals as $\mathbf{TS}_1'$
Extract the high-quality individuals as $\mathbf{TS}_2'$
Extract the high-quality individuals as $\mathbf{TS}_m'$
For each T_i (i = 1, ⋯, m), exploit OCAT to align the other tasks with it and construct its sample set $\mathbf{TS}_i$ based on $\mathbf{TS}_1', \mathbf{TS}_2', \ldots, \mathbf{TS}_m'$ with the alignment transformations
Estimate the probability distribution model $\boldsymbol{G}_1(\mathbf{x})$ based on $\mathbf{TS}_1$
Estimate the probability distribution model $\boldsymbol{G}_2(\mathbf{x})$ based on $\mathbf{TS}_2$
Estimate the probability distribution model $\boldsymbol{G}_m(\mathbf{x})$ based on $\mathbf{TS}_2$
Update the population by sampling from $\boldsymbol{G}_1(\mathbf{x})$ and evaluate it
Update the population by sampling from $\boldsymbol{G}_2(\mathbf{x})$ and evaluate it
Update the population by sampling from $\boldsymbol{G}_m(\mathbf{x})$ and evaluate it
NO
Is the termination condition met?
Is the termination condition met?
NO
Is the termination condition met?
NO
YES
YES
YES
Output best individual found in the population
Output best individual found in the population
Output best individual found in the population
END
END
END

Fig. 5. The overall framework of MaT-EDA.

```
Algorithm 3: MaT-EDA
Randomly initialize the population $\mathbf{P}_i$ for each task $T_i$, $i=1,\ldots,m$;
Evaluate the individuals in $\mathbf{P}_1,\ldots,\mathbf{P}_m$ on the corresponding tasks;
while there are fitness evaluations remained
    for each task $T_i$
        Initialize $\mathbf{TS}_i$ with the high-quality individuals in $\mathbf{P}_i$;
        for each task $T_j \neq T_i$
            Learn the intertask alignment transformation between $T_j$ and $T_i$:
                $(\mathbf{S}_{ji},\mathbf{R}_{ji},\mathbf{t}_{ji}) \leftarrow \mathrm{OCAT}(\mathbf{P}_j,\mathbf{P}_i)$;
            Transform each high-quality individual $\mathbf{p}_j$ of $\mathbf{P}_j$ and add them into $\mathbf{TS}_i$:
                $\mathbf{TS}_i \leftarrow \mathbf{R}_{ji}\cdot\mathbf{S}_{ji}\cdot\mathbf{p}_j+\mathbf{t}_{ji}$;
        Estimate the probability distribution model $\boldsymbol{G}_i(\mathbf{x})$ based on $\mathbf{TS}_i$;
        Generate the offspring $\mathbf{O}_i$ by sampling from $\boldsymbol{G}_i(\mathbf{x})$ and evaluate them;
    Update $\mathbf{P}_1,\ldots,\mathbf{P}_m$ with $\mathbf{O}_1,\ldots,\mathbf{O}_m$, respectively;
return the best individuals in $\mathbf{P}_1,\ldots,\mathbf{P}_m$.
```

the original papers, so were the peculiar parameters in each comparative intertask alignment algorithm. By this means, a fair comparison can be ensured. As for the parameter of OCAT, i.e., the selection ratio ρ, we set it to 0.15 based on the empirical study, which will be provided in Section 4.3.

(1) *Comparison on the single-objective multi-tasking benchmark problems*: Table 3 reports the final optimization results of SOMFEA$_{\mathrm{OCAT}}$ and its seven competitors on the CEC2017 single-objective multi-tasking benchmark suite. The results highlighted in bold are the best, and the bottom line summarizes the comparison result. From Table 3, we can observe that OCAT achieves impressive performance. It enables SOMFEA$_{\mathrm{OCAT}}$ to obtain the best solution quality on 12 out of all 18 composite functions, which ranks first among the eight evaluated multitasking optimization algorithms.

The two baseline algorithms GA and SOMFEA perform best on two and five functions, respectively. These inferior results to SOMFEA$_{\mathrm{OCAT}}$ indicate that OCAT can effectively improve evolutionary multitasking. A closer inspection finds that the tasks where SOMFEA outperforms the other algorithms mainly come from S1–S3. The reason is that the composite tasks have the same optimal solution and thus are homogeneous. As a result, the intertask alignment takes little effect on them.

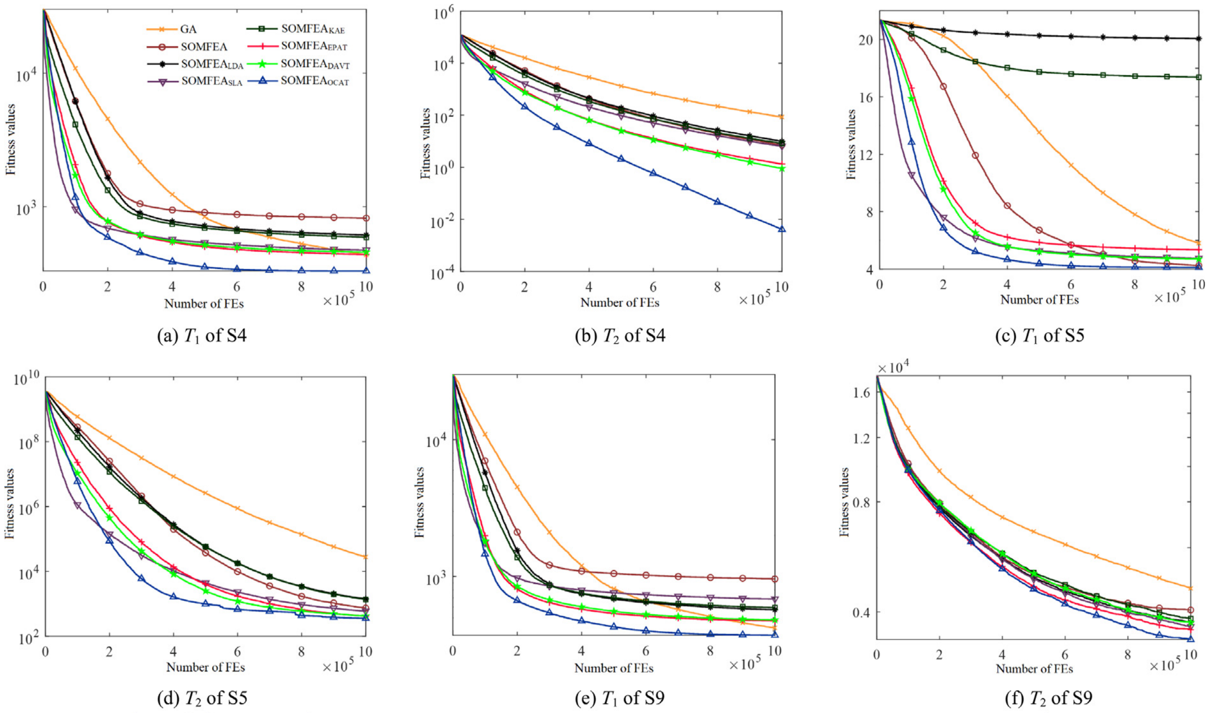
SOMFEA$_{\mathrm{LDA}}$, SOMFEA$_{\mathrm{SLA}}$, and SOMFEA$_{\mathrm{KAE}}$ are all equipped with the pairwise-learning-based intertask alignment algorithm. They accomplish the best optimization results on two, two, and four functions, respectively. Since the main differences among these

Table 3
The median, mean, and standard deviation of the best fitness values obtained by SOMFEA$_{\mathrm{OCAT}}$ and its seven competitors over 30 independent runs on the CEC2017 single-objective multi-tasking benchmark problems.

Function		Stats.	GA	SOMFEA	SOMFEA$_{\mathrm{LDA}}$	SOMFEA$_{\mathrm{SLA}}$	SOMFEA$_{\mathrm{KAE}}$	SOMFEA$_{\mathrm{EPAT}}$	SOMFEA$_{\mathrm{DAVT}}$	SOMFEA$_{\mathrm{OCAT}}$
S1	T_1	Median	9.24e−01	3.56e−01	4.29e−01	3.12e−01	1.01e−01	1.82e−01	1.66e−01	**4.17e−02**
		Mean	9.00e−01	3.60e−01	4.29e−01	3.15e−01	1.02e−01	1.88e−01	1.57e−01	**4.15e−02**
		Std	7.68e−02	5.29e−02	6.40e−02	5.94e−02	2.67e−02	3.92e−02	3.49e−02	**1.51e−02**
	T_2	Median	4.10e+02	**1.83e+02**	5.91e+02	4.34e+02	5.67e+02	4.35e+02	4.37e+02	3.30e+02
		Mean	4.24e+02	**1.88e+02**	5.84e+02	4.30e+02	6.01e+02	4.49e+02	4.31e+02	3.56e+02
		Std	4.62e+01	**4.28e+01**	1.12e+02	6.12e+01	1.61e+02	9.49e+01	9.81e+01	8.25e+01
S2	T_1	Median	5.07e+00	4.56e+00	2.01e+01	4.69e+00	1.98e+01	4.07e+00	**3.70e+00**	4.15e+00
		Mean	5.29e+00	4.71e+00	2.02e+01	4.80e+00	1.53e+01	5.15e+00	**4.20e+00**	4.06e+00
		Std	1.09e+00	8.67e−01	1.93e−01	6.98e−01	7.17e+00	4.00e+00	**3.04e+00**	6.66e−01
	T_2	Median	4.28e+02	**2.25e+02**	5.68e+02	4.17e+02	5.80e+02	4.24e+02	4.22e+02	3.46e+02
		Mean	4.33e+02	**2.21e+02**	5.94e+02	4.32e+02	5.69e+02	4.29e+02	4.25e+02	3.61e+02
		Std	5.91e+01	**5.63e+01**	1.32e+02	7.48e+01	1.13e+02	8.51e+01	8.77e+01	9.06e+01
S3	T_1	Median	2.12e+01	2.01e+01	2.02e+01	2.01e+01	2.01e+01	2.01e+01	**3.72e+00**	2.00e+01
		Mean	2.12e+01	2.02e+01	2.02e+01	2.02e+01	1.90e+01	2.01e+01	**5.02e+00**	2.01e+01
		Std	9.49e−02	7.52e−02	1.41e−01	6.43e−02	4.19e+00	5.93e−02	**4.24e+00**	6.15e−02
	T_2	Median	4.16e+03	**3.64e+03**	**3.61e+03**	**3.64e+03**	**3.06e+03**	**3.67e+03**	**3.59e+03**	**3.48e+03**
		Mean	4.21e+03	**3.73e+03**	**3.69e+03**	**3.53e+03**	**3.34e+03**	**3.65e+03**	**3.81e+03**	**3.50e+03**
		Std	5.53e+02	**5.03e+02**	**4.82e+02**	**4.18e+02**	**9.25e+02**	**4.38e+02**	**9.90e+02**	**5.48e+02**
S4	T_1	Median	4.36e+02	5.90e+02	6.02e+02	4.96e+02	5.46e+02	4.36e+02	4.44e+02	**3.25e+02**
		Mean	4.38e+02	6.17e+02	6.15e+02	4.74e+02	5.57e+02	4.40e+02	4.58e+02	**3.31e+02**
		Std	5.41e+01	1.38e+02	1.14e+02	1.04e+02	9.39e+01	7.05e+01	9.01e+01	**7.75e+01**
	T_2	Median	8.42e+01	8.55e+00	9.89e+00	6.37e+00	3.19e−01	1.36e+00	8.54e−01	**2.40e−03**
		Mean	8.52e+01	8.40e+00	9.85e+00	6.47e+00	3.32e−01	1.34e+00	9.03e−01	**4.07e−03**
		Std	1.72e+01	1.51e+00	1.87e+00	1.34e+00	8.79e−02	3.28e−01	2.96e−01	**3.77e−03**
S5	T_1	Median	5.20e+00	3.63e+00	2.00e+01	4.69e+00	1.98e+01	4.21e+00	4.13e+00	**2.99e+00**
		Mean	5.81e+00	3.62e+00	2.01e+01	4.76e+00	1.72e+01	5.35e+00	4.71e+00	**4.12e+00**
		Std	2.59e+00	5.34e−01	9.91e−02	8.98e−01	5.75e+00	4.03e+00	3.00e+00	**4.24e+00**
	T_2	Median	2.51e+04	6.11e+02	1.16e+03	5.45e+02	**2.57e+02**	3.93e+02	3.33e+02	**1.95e+02**
		Mean	2.78e+04	6.50e+02	1.41e+03	5.88e+02	**4.13e+02**	4.14e+02	4.16e+02	**3.57e+02**
		Std	1.26e+04	2.44e+02	8.28e+02	2.48e+02	**8.26e+02**	1.09e+02	2.80e+02	**6.98e+02**
S6	T_1	Median	4.89e+00	2.00e+01	2.00e+01	5.07e+00	1.96e+01	**4.25e+00**	4.62e+00	**3.76e+00**
		Mean	5.16e+00	2.00e+01	2.00e+01	5.06e+00	1.52e+01	**4.27e+00**	4.75e+00	**5.38e+00**
		Std	8.72e−01	1.11e−01	9.37e−02	8.72e−01	7.01e+00	**8.26e−01**	7.74e−01	**4.98e+00**
	T_2	Median	**1.12e+01**	2.14e+01	2.19e+01	8.99e+00	2.08e+01	1.99e+01	1.72e+01	1.77e+01
		Mean	**1.17e+01**	2.09e+01	2.20e+01	9.01e+00	2.04e+01	2.00e+01	1.73e+01	1.77e+01
		Std	**2.13e+00**	3.60e+00	2.78e+00	1.90e+00	3.20e+00	3.79e+00	2.78e+00	4.07e+00
S7	T_1	Median	2.25e+04	8.12e+02	1.31e+03	5.18e+02	**2.74e+02**	5.21e+02	3.22e+02	**2.45e+02**
		Mean	2.67e+04	8.50e+02	1.84e+03	6.04e+02	**4.87e+02**	7.51e+02	4.97e+02	**3.67e+02**
		Std	1.29e+04	2.69e+02	1.66e+03	3.42e+02	**8.08e+02**	6.62e+02	4.46e+02	**4.36e+02**
	T_2	Median	4.16e+02	**2.69e+02**	5.90e+02	4.43e+02	5.61e+02	4.36e+02	4.66e+02	3.46e+02
		Mean	4.27e+02	**2.86e+02**	6.05e+02	4.51e+02	5.58e+02	4.68e+02	4.79e+02	3.68e+02
		Std	6.04e+01	**8.33e+01**	1.15e+02	1.03e+02	1.25e+02	1.19e+02	1.11e+02	1.10e+02
S8	T_1	Median	9.35e−01	3.99e−01	4.32e−01	3.48e−01	9.62e−02	2.01e−01	1.63e−01	**7.41e−02**
		Mean	9.22e−01	4.08e−01	4.16e−01	3.46e−01	9.64e−02	2.01e−01	1.61e−01	**7.41e−02**
		Std	6.11e−02	6.45e−02	6.98e−02	5.72e−02	2.08e−02	3.97e−02	2.78e−02	**1.63e−02**
	T_2	Median	**3.81e+01**	2.73e+01	5.17e+01	2.02e+01	4.96e+01	4.64e+01	4.57e+01	**3.77e+01**
		Mean	**3.78e+01**	2.71e+01	5.14e+01	2.12e+01	4.84e+01	4.56e+01	4.52e+01	**3.75e+01**
		Std	**4.21e+00**	3.18e+00	4.67e+00	4.23e+00	5.82e+00	5.54e+00	5.61e+00	**5.12e+00**
S9	T_1	Median	4.12e+02	6.41e+02	5.67e+02	6.83e+02	5.46e+02	4.86e+02	4.71e+02	**3.56e+02**
		Mean	4.22e+02	6.26e+02	5.71e+02	6.86e+02	5.61e+02	4.75e+02	4.83e+02	**3.73e+02**
		Std	6.55e+01	1.08e+02	1.29e+02	1.19e+02	1.17e+02	9.80e+01	1.10e+02	**6.92e+01**
	T_2	Median	4.50e+03	**3.60e+03**	**3.76e+03**	**3.84e+03**	**3.06e+03**	**3.57e+03**	**3.52e+03**	**3.48e+03**
		Mean	4.64e+03	**3.55e+03**	**3.74e+03**	**3.65e+03**	**3.14e+03**	**3.58e+03**	**3.74e+03**	**3.35e+03**
		Std	6.68e+02	**5.03e+02**	**3.97e+02**	**5.41e+02**	**5.11e+02**	**4.50e+02**	**9.81e+02**	**4.37e+02**
No. Success			2	5	2	2	4	3	4	12

algorithms mainly lie in the adopted intertask alignment algorithms, the worse results of SOMFEA$_{\mathrm{LDA}}$, SOMFEA$_{\mathrm{SLA}}$, and SOMFEA$_{\mathrm{KAE}}$ than SOMFEA$_{\mathrm{OCAT}}$ can fully reveal the superiority of OCAT over the other three pairwise-learning-based alignment methods. Specifically, LDA, SLA, and KAE may encounter the issue of chaotic matching with the fitness-ranking-based pairwise fashion. OCAT can smoothly alleviate this issue by constructing a mathematical model for the intertask alignment and deducing its optimal.

The comparison between SOMFEA$_{\mathrm{OCAT}}$ and the remaining two algorithms further demonstrate the effectiveness of OCAT. SOMFEA$_{\mathrm{EPAT}}$ and SOMFEA$_{\mathrm{DAVT}}$ exhibit the best optimization performance on 3 and 4 functions, respectively, but are dominated by SOMFEA$_{\mathrm{OCAT}}$ on most of the others. Especially on tasks such as T_2 of S4 and T_1 of S8, SOMFEA$_{\mathrm{OCAT}}$ obtains better solution quality than the two competitors by order of magnitude in terms of fitness mean. These results show that OCAT can also outperform



(a) T_1 of S4 (b) T_2 of S4 (c) T_1 of S5

(d) T_2 of S5 (e) T_1 of S9 (f) T_2 of S9

Fig. 6. The evolution curves of GA, SOMFEA, SOMFEA$_{\mathrm{LDA}}$, SOMFEA$_{\mathrm{SLA}}$, SOMFEA$_{\mathrm{KAE}}$, SOMFEA$_{\mathrm{EPAT}}$, SOMFEA$_{\mathrm{DAVT}}$, and SOMFEA$_{\mathrm{OCAT}}$ on some representative problem sets in the CEC2017 single-objective multi-tasking benchmark problems.

the two distribution-learning-based intertask algorithms EPAT and DAVT in terms of the amelioration of negative transfer.

To further illustrate the effectiveness of OCAT, we take S4, S5, and S9 as examples and present the evolution curves of the participant algorithms on their composite tasks in Figs. 6(a)–(f). From these figures, it can be seen that SOMFEA quickly gets prematurely convergent due to the effect of negative transfer. Benefiting from the intertask alignment algorithms, the other five SOMFEA variants well alleviate this limitation and SOMFEA$_{\mathrm{OCAT}}$ is the dominant one. It greatly expedites the convergence and facilitates locating higher-quality solutions.

(2) *Comparison on the multiple-objective multi-tasking benchmark problems*: Table 4 presents the optimization results of MOMFEA$_{\mathrm{OCAT}}$ and its seven competitors on the CEC2017 multiple-objective multi-tasking benchmark suite. The widely-used inverted generational distance (IGD) is employed to measure the performance of each algorithm. The smaller the IGD value is, the better performance the algorithm has. For the calculation of IGD, readers can refer to [50]. From Table 4, it can be observed that MOMFEA$_{\mathrm{OCAT}}$ also achieves satisfying performance. It obtains the best results on 6 out of all 18 tasks, while NSGA-II, MOMFEA, MOMFEA$_{\mathrm{LDA}}$, MOMFEA$_{\mathrm{SLA}}$, MOMFEA$_{\mathrm{KAE}}$, MOMFEA$_{\mathrm{EPAT}}$, and MOMFEA$_{\mathrm{DAVT}}$ achieve the best optimization performance on 4, 4, 0, 2, 3, 5, and 1 tasks, respectively. Moreover, the tasks on which MOMFEA$_{\mathrm{OCAT}}$ significantly dominates the competitors are mainly from the problem sets with partial intersection or no intersection. For example, on T_1 of S4 and T_2 of S7, the IGD values achieved by MOMFEA$_{\mathrm{OCAT}}$ are smaller than those of the competitors by orders of magnitude. Fig. 7 presents the IGD curves of the eight algorithms on the composite tasks of S4, S7, and S8. We can notice that MOMFEA$_{\mathrm{OCAT}}$ shows faster convergence speed as compared with its competitors, especially in the early evolutionary stage. This further verifies that OCAT can effectively weaken the impact of negative transfer.

4.2. Investigation of the performance of MaT-EDA

To investigate the performance of MaT-EDA, we first compared its optimization results on many-tasking problems with those of the conventional EDA. For a comprehensive investigation of this many-tasking optimization framework, we further integrated the EDA variant AMaLGaM [37] into it and assessed the performance difference between the resulting algorithm and the original one. Besides, the comparison with two state-of-the-art many-tasking algorithms, i.e., MaTDE [18] and EMaTO-MKT [19], was also conducted. Their main ideas are described in Section 2.1. As MaT-EDA is mainly developed for single-objective optimization problem, this simulation was conducted on the WCCI2020 single-objective many-tasking optimization benchmark problems [51]. This benchmark suite contains ten many-tasking problems sets, and each one involves 50 composite tasks. Table 5 provides the function combination of each problem set. For the detailed composition of this benchmark suite, readers can refer to [51].

According to the guidelines in [51], each tested algorithm was given a maximum number of 5 × 10^6 FEs in a single run, and its performance was assessed based on the results obtained over 30 independent runs. Besides, the Kruskal–Wallis nonparametric one-way ANOVA test with a confidence interval of 0.95 and the two-tailed Wilcoxon rank-sum tests at a significance level of 0.05 were also employed for the performance comparison. It is worth mentioning that we thought there is no difference between two solutions whose fitness values are both smaller than 10^{-10}. During the simulation, the population size in EDA or AMaLGaM was set to 2000 as [37] suggested. Since each problem set involves 50 tasks, we set the population size for each task in MaT-EDA or MaT-AMaLGaM to 100. The other peculiar parameters involved in EDA, AMaLGaM, MaTDE, and EMaTO-MKT were configured strictly following their original papers for a fair comparison.

(1) *Comparison with the single-task-oriented EDAs*: Fig. 8(a) and (b) present the comparison results between MaT-EDA and EDA

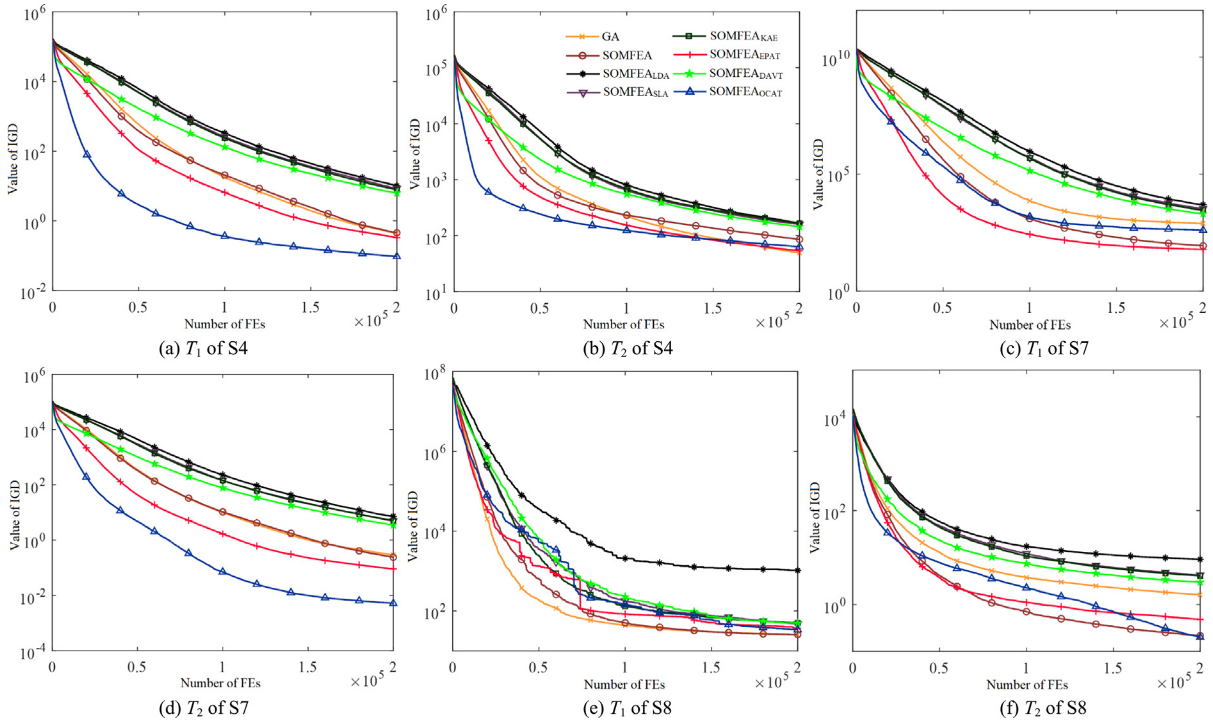
Table 4
The median, mean, and standard deviation of the best IGD values obtained by SOMFEA$_{\mathrm{OCAT}}$ and its seven competitors over 30 independent runs on the CEC2017 multiple-objective multi-tasking benchmark problems.

Function		Stats.	NSGA-II	MOMFEA	MOMFEA$_{\mathrm{LDA}}$	MOMFEA$_{\mathrm{SLA}}$	MOMFEA$_{\mathrm{KAE}}$	MOMFEA$_{\mathrm{EPAT}}$	MOMFEA$_{\mathrm{DAVT}}$	MOMFEA$_{\mathrm{OCAT}}$
S1	T_1	Median	9.04e−01	2.05e−01	1.30e+01	1.10e+01	1.02e+01	**5.55e−02**	7.36e+00	1.97e−01
		Mean	8.91e−01	2.09e−01	1.28e+01	1.05e+01	1.07e+01	**5.55e−02**	7.96e+00	2.20e−01
		Std	2.30e−01	3.99e−02	2.94e+00	2.15e+00	2.38e+00	**1.17e−02**	1.95e+00	8.97e−02
	T_2	Median	5.01e−01	4.22e−01	2.18e+00	1.90e+00	1.90e+00	2.00e−01	1.66e+00	**4.84e−02**
		Mean	4.91e−01	4.16e−01	2.15e+00	1.93e+00	1.91e+00	1.96e−01	1.68e+00	**4.88e−02**
		Std	5.73e−02	4.04e−02	3.67e−01	2.52e−01	2.27e−01	2.33e−02	2.54e−01	**1.04e−02**
S2	T_1	Median	4.15e+00	**7.28e−03**	4.17e+00	5.75e+00	5.32e+00	3.24e−01	5.18e+00	3.82e−01
		Mean	3.83e+00	**4.80e−01**	3.51e+00	4.61e+00	4.48e+00	2.09e+00	4.32e+00	4.49e−01
		Std	2.22e+00	**1.45e+00**	2.33e+00	2.11e+00	2.16e+00	2.53e+00	2.05e+00	2.47e−01
	T_2	Median	3.01e−01	4.37e−02	5.56e−01	5.59e−01	5.66e−01	3.01e−01	2.68e−01	**1.07e−02**
		Mean	4.55e−01	1.48e−01	7.54e−01	7.82e−01	6.84e−01	3.54e−01	3.79e−01	**7.17e−02**
		Std	4.41e−01	2.36e−01	7.24e−01	7.60e−01	6.41e−01	3.43e−01	3.05e−01	**1.79e−01**
S3	T_1	Median	2.78e+01	2.63e−01	1.35e+02	7.02e+01	7.90e+01	**4.92e−02**	2.63e+01	3.45e+01
		Mean	2.77e+01	2.87e−01	1.44e+02	6.17e+01	7.43e+01	**4.83e−02**	3.18e+01	3.42e+01
		Std	3.78e+00	7.59e−02	2.20e+01	2.15e+01	2.40e+01	**1.30e−02**	2.17e+01	6.31e+00
	T_2	Median	1.58e−02	5.90e−03	1.08e+00	3.82e−02	3.81e−02	**3.02e−03**	3.41e−02	1.20e−02
		Mean	1.59e−02	6.11e−03	9.69e−01	3.88e−02	3.83e−02	**3.00e−03**	3.41e−02	1.20e−02
		Std	1.77e−03	8.16e−04	7.34e−01	4.55e−03	3.95e−03	**3.75e−04**	2.59e−03	1.39e−03
S4	T_1	Median	4.15e−01	4.47e−01	1.01e+01	8.54e+00	7.54e+00	3.26e−01	6.20e+00	**8.77e−02**
		Mean	4.38e−01	4.55e−01	1.03e+01	8.44e+00	7.79e+00	3.33e−01	6.26e+00	**9.44e−02**
		Std	1.21e−01	1.27e−01	2.32e+00	2.11e+00	1.56e+00	7.49e−02	1.33e+00	**2.70e−02**
	T_2	Median	**4.83e+01**	8.24e+01	1.58e+02	1.49e+02	1.63e+02	5.45e+01	1.41e+02	6.11e+01
		Mean	**4.87e+01**	8.56e+01	1.66e+02	1.59e+02	1.66e+02	5.39e+01	1.43e+02	6.31e+01
		Std	**1.12e+01**	1.42e+01	3.85e+01	3.26e+01	4.20e+01	1.05e+01	3.23e+01	1.60e+01
S5	T_1	Median	3.82e−01	3.42e−01	2.21e+00	**1.51e−01**	**1.39e−01**	7.85e−01	5.28e−01	5.02e−01
		Mean	3.90e−01	3.49e−01	3.18e+00	**2.00e−01**	**1.56e−01**	7.82e−01	6.39e−01	5.18e−01
		Std	7.59e−02	9.54e−02	2.05e+00	**2.30e−01**	**7.35e−02**	1.16e−01	4.16e−01	1.57e−01
	T_2	Median	1.27e+03	8.11e+02	9.74e+02	**5.74e+01**	**5.79e+01**	8.35e+02	**5.91e+01**	8.32e+02
		Mean	1.27e+03	8.14e+02	1.45e+03	**1.16e+02**	**1.25e+02**	8.72e+02	**2.79e+02**	8.93e+02
		Std	3.87e+02	1.26e+02	9.15e+02	**1.25e+02**	**1.43e+02**	2.84e+02	**3.63e+02**	2.64e+02
S6	T_1	Median	**3.01e−02**	9.72e−02	3.23e−01	2.42e−01	2.22e−01	7.62e−02	2.09e−01	1.21e−01
		Mean	**3.29e−02**	9.90e−02	3.21e−01	2.48e−01	2.30e−01	7.94e−02	2.22e−01	1.16e−01
		Std	**1.00e−02**	1.87e−02	6.61e−02	5.02e−02	3.61e−02	1.46e−02	5.26e−02	3.84e−02
	T_2	Median	2.08e+01	**4.17e+00**	2.00e+01	2.06e+01	2.06e+01	2.00e+01	2.00e+01	2.12e+01
		Mean	2.08e+01	**4.19e+00**	2.00e+01	2.06e+01	1.96e+01	2.00e+01	1.48e+01	1.93e+01
		Std	1.37e−01	**3.60e−01**	1.19e−03	1.22e−01	3.92e+00	1.49e−03	6.67e+00	5.74e+00
S7	T_1	Median	4.23e+02	8.69e+01	4.09e+03	2.86e+03	2.41e+03	**5.95e+01**	1.83e+03	2.17e+02
		Mean	7.76e+02	8.62e+01	4.87e+03	3.30e+03	2.83e+03	**6.05e+01**	2.00e+03	4.00e+02
		Std	1.36e+03	5.87e+00	2.41e+03	1.56e+03	1.60e+03	**3.74e+00**	6.85e+02	6.34e+02
	T_2	Median	2.89e−01	2.44e−01	7.38e+00	4.79e+00	4.86e+00	8.61e−02	3.32e+00	**5.12e−03**
		Mean	2.91e−01	2.38e−01	7.22e+00	4.90e+00	5.17e+00	9.03e−02	3.50e+00	**5.12e−03**
		Std	5.30e−02	4.62e−02	1.50e+00	1.15e+00	1.23e+00	2.68e−02	8.51e−01	**1.09e−03**
S8	T_1	Median	**1.72e+01**	**1.78e+01**	5.73e+01	3.18e+01	4.09e+01	**1.79e+01**	2.90e+01	**2.00e+01**
		Mean	**2.55e+01**	**2.56e+01**	1.04e+03	4.70e+01	5.03e+01	**3.65e+01**	4.91e+01	**3.38e+01**
		Std	**2.22e+01**	**2.22e+01**	4.60e+03	3.20e+01	3.07e+01	**3.40e+01**	3.35e+01	**2.68e+01**
	T_2	Median	1.19e+00	**7.05e−02**	3.95e+00	3.58e+00	2.16e+00	2.25e−01	1.80e+00	**1.08e−01**
		Mean	1.61e+00	**2.16e−01**	9.16e+00	4.22e+00	4.07e+00	4.81e−01	2.99e+00	**2.00e−01**
		Std	1.50e+00	**4.17e−01**	1.41e+01	3.99e+00	6.24e+00	7.37e−01	2.70e+00	**1.92e−01**
S9	T_1	Median	**2.23e−01**	1.06e+00	1.36e+00	1.13e+00	1.12e+00	1.07e+00	1.10e+00	1.06e+00
		Mean	**2.33e−01**	9.72e−01	3.00e+00	1.13e+00	1.12e+00	1.06e+00	1.10e+00	1.05e+00
		Std	**7.20e−02**	2.07e−01	2.36e+00	1.99e−02	2.20e−02	4.13e−02	1.83e−02	2.63e−02
	T_2	Median	2.00e+01	2.01e+01	2.00e+01	2.00e+01	**2.00e+01**	2.00e+01	2.00e+01	2.02e+01
		Mean	2.00e+01	2.01e+01	2.00e+01	2.00e+01	**2.00e+01**	2.00e+01	1.95e+01	2.02e+01
		Std	5.69e−02	1.06e−01	1.86e−01	2.97e−02	**8.18e−03**	3.06e−04	3.00e+00	1.27e−01
No. Success			4	4	0	2	3	5	1	6

and those between MaT-AMaLGaM and AMaLGaM, respectively.[2] From Fig. 8, we can observe that each resulting many-tasking algorithm achieves significantly better optimization performance than the corresponding conventional one. The superiority of MaT-EDA and MaT-AMaLGaM mainly benefits from the fact that with the assistance of OCAT, they can employ the solutions from other tasks as the samples for the current probability model estimation. As a result, the population requirement of each task is reduced so that more generations can be evolved to locate better optimization results.

For MaT-EDA, its performance is superior to that of EDA on nine problem sets. A closer inspection finds that the algorithm achieves significantly better optimization results than EDA by winning on 40+ out of all the 50 composite tasks on each of them. Besides, it should be pointed out that MaT-EDA is defeated by EDA on S2. The inferior of MaT-EDA on this set can

[2] The detailed many-tasking optimization results are available from URL https://github.com/chenanXJTU.



(a) T_1 of S4

(b) T_2 of S4

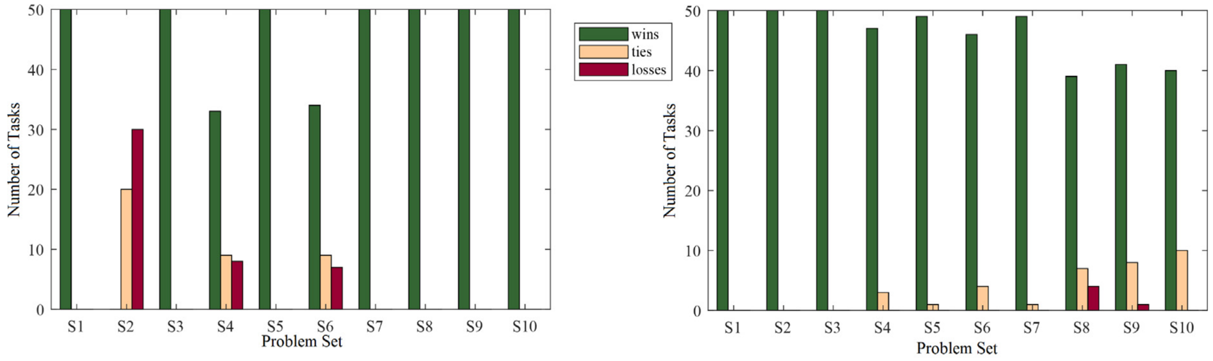
(c) T_1 of S7

(d) T_2 of S7

(e) T_1 of S8

(f) T_2 of S8

Fig. 7. The IGD curves of NSGA-II, MOMFEA, MOMFEA$_{\text{LDA}}$, MOMFEA$_{\text{SLA}}$, MOMFEA$_{\text{KAE}}$, MOMFEA$_{\text{EPAT}}$, MOMFEA$_{\text{OCAT}}$, and MOMFEA$_{\text{OCAT}}$ on some representative problem sets in the CEC2017 multiple-objective multi-tasking benchmark problems.



(a) MaT-EDA *vs* EDA

(b) MaT-AMaLGaM *vs* AMaLGaM

Fig. 8. The numbers of wins, ties, and losses of each resulting many-tasking optimization algorithm against its original one on each WCCI2020 single-objective 50-tasking optimization benchmark problem set.

Table 5
The function combination of each problem set in WCCI2020 single-objective many-tasking optimization benchmark problems.

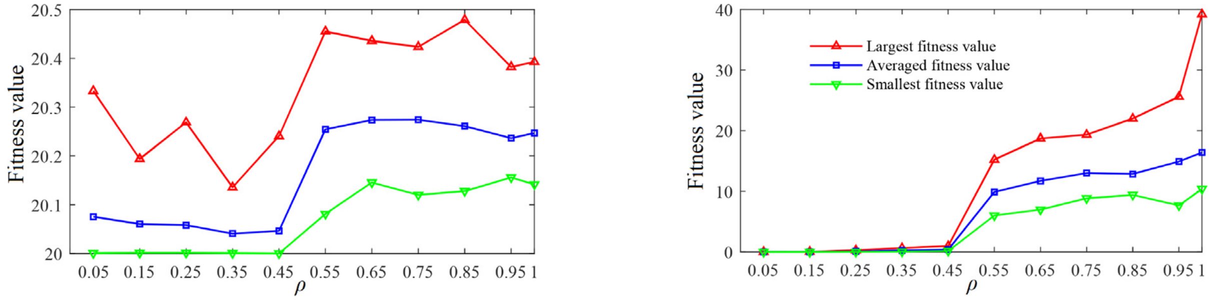
	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Sphere	✔			✔						
Rosenbrock		✔		✔		✔		✔	✔	
Rastrigin			✔		✔		✔	✔	✔	✔
Ackley				✔			✔	✔	✔	✔
Griewank					✔	✔		✔	✔	✔
Weierstrass					✔		✔	✔	✔	✔
Schwefel						✔			✔	✔

Table 6
The number of best results for all tasks in each problem set obtained by MaTDE, EMaTO-MKT, and MaT-EDA on the WCCI2020 single-objective many-tasking optimization benchmark suite.

Problem set	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
MaTDE	0	25	0	7	0	3	0	0	0	0
EMaTO-MKT	**50**	**50**	0	**50**	17	**50**	17	29	**34**	**30**
MaT-EDA	**50**	19	**50**	40	**33**	7	**50**	**34**	28	**30**

be mainly attributed to that the employed basic EDA can easily get prematurely converged on the composite tasks and the corresponding populations lose their diversity quickly. When taking the solutions from these tasks as the training samples, the estimated probability model will be distorted so that the quality of the located solutions dramatically deteriorates. As a comparison, MaT-AMaLGaM alleviates the premature convergence and achieves impressive many-tasking optimization performance. It wins AMaLGaM on 40+ out of all the 50 composite tasks on each problem set.

(2) *Comparison with the state-of-the-art many-tasking optimization algorithms*: Table 6 summarizes the comparison results of MaT-EDA with MaTDE and EMaTO-MKT. Particularly, for each algorithm on each benchmark problem set, the number of tasks on which the algorithm achieves the best performance is reported. The corresponding comparison results of MaT-AMaLGaM



(a) T_1 of S3

(b) T_2 of S4

Fig. 9. The performance variation of SOMFEA$_{\text{OCAT}}$ w.r.t. different ρ values.

Table 7
The number of best results for all tasks in each problem set obtained by MaTDE, EMaTO-MKT, and MaT-AMaLGaM on the WCCI2020 single-objective many-tasking optimization benchmark suite.

Problem set	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
MaTDE	0	0	0	0	0	0	0	0	0	0
EMaTO-MKT	50	0	0	**33**	0	17	17	10	17	20
MaT-AMaLGaM	**50**	**50**	**50**	17	**50**	**33**	**50**	**40**	**33**	**30**

are shown in Table 7 document. From Table 6, we can observe that MaT-EDA significantly outperforms MaTDE and achieves competitive performance compared with EMaTO-MKT. The three algorithms accomplish the best optimization results on 6, 0, and 6 problem sets, respectively. The performance deterioration of MaT-EDA on the other four sets can be attributed to the premature convergence of EDA. By alleviating this issue, MaT-AMaLGaM achieves the best optimization results on 9 problem sets as reported in Table 7. Moreover, on most of these sets, such as S2, S3, and S5, MaT-AMaLGaM shows great dominance against its competitors that it accomplishes the best optimization results on over 40 tasks.

4.3. Influence of parameters

OCAT only involves one free parameter, i.e., the selection ratio ρ. It determines the quality and quantity of the selected solution sets used to learn the intertask transformation. A small value of ρ can ensure the solution quality and promote the selected solutions to describe corresponding task landscape. Nevertheless, it meanwhile reduces the set quantity, which may be adverse to the intertask alignment transformation learning. Therefore, it is expected to specify a proper value for ρ.

To reveal the sensitivity of OCAT to ρ, we investigated the performance variation of SOMFEA$_{\text{OCAT}}$ w.r.t. the parameter on a variety of multi-tasking benchmark problems. The values considered for ρ include {0.05, 0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95, 1}. Taking T_1 of S3 and T_2 of S4 in the CEC2017 single-objective multi-tasking benchmark suite as the examples, Fig. 9 presents the tested results. It can be observed that SOMFEA$_{\text{OCAT}}$ performs well when ρ varies in the range of [0.05, 0.45] and shows performance deterioration when ρ exceeds 0.5. This result means that the algorithm is rather robust to ρ. Only when some middle-quality or low-quality solutions are selected, will the efficacy of the learned intertask transformation deteriorate. As a result, we recommend selecting a value for ρ from the interval [0.05, 0.45] and set it to 0.15 in this study.

5. Conclusions

In this paper, a novel intertask alignment algorithm named optimal correspondence assisted affine transformation (OCAT) is proposed to enhance evolutionary multi-tasking on heterogeneous optimization tasks. Instead of pairing up the training samples from the involved tasks rigidly, OCAT constructs their correspondences and calculates the transformation parameters in turn, which facilitates the optimal correspondences that enable the transformation to achieve maximized intertask structural similarity. Moreover, OCAT develops a novel method to derive the parameters in affine transformation and helps ensure the quality of the obtained transformation. Finally, this study combines OCAT with EDA and develops a many-tasking optimization algorithm MaT-EDA. With the help of OCAT, MaT-EDA explicitly transfers the high-quality solutions from the assisted tasks into the target one as the samples for estimating the current probability model, thereby reducing the population demand. Extensive experiments have been conducted to investigate the effectiveness of OCAT and MaT-EDA. The empirical results show that OCAT outperforms some state-of-the-art intertask alignment algorithms, and MaT-EDA can also achieve impressive many-tasking optimization performance.

In our future work, we will replace affine transformation with some more sophisticated non-linear transformation strategies to improve the final intertask alignment performance. Besides, applying the developed algorithms to some real-world multi-tasking or many-tasking optimization problems is also worth pursuing.

CRediT authorship contribution statement

An Chen: Conceptualization, Methodology, Software, Validation, Investigation, Writing – original draft, Writing – review & editing. **Zhigang Ren:** Conceptualization, Methodology, Visualization, Investigation, Writing – review & editing. **Muyi Wang:** Software, Validation, Visualization, Investigation, Writing – review & editing. **Shenyu Su:** Software, Investigation, Writing – review & editing. **Jiaqi Yun:** Writing – review & editing. **Yichuang Wang:** Visualization, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant number 61873199), the Natural Science Basic Research Plan in Shaanxi Province of China (grant number 2020JM-059), and the Fundamental Research Funds for the Central Universities of China [grant numbers xzy022020057].

Appendix

A.1. The proof process of Theorem 1

Theorem 1. *For any two individual sets* $\mathbf{E}_a = \{\mathbf{p}^a_{(1)}, \ldots, \mathbf{p}^a_{(NS)}\}$ *and* $\mathbf{E}_b = \{\mathbf{p}^b_{(1)}, \ldots, \mathbf{p}^b_{(NS)}\}$, *OCAT must converge on the optimal solution of Eq.* (3).

Proof. After the update of the correspondences between the individuals in $\mathbf{E}_a$ and $\mathbf{E}_b$ in the kth iteration, the minimum SSR is:

$$e_k = \sum_{i=1}^{NS} \|\mathbf{p}^b_{(\mathbf{c}^k_{ab}(i))} - (\mathbf{A}^{k-1}_{ab} \cdot \mathbf{p}^a_{(i)} + \mathbf{t}^{k-1}_{ab})\|^2.$$

To further decrease SSR, the new transformation parameters $\mathbf{A}^k_{ab}$ and $\mathbf{t}^k_{ab}$ are derived according to Eq. (5), and the minimum SSR thus becomes:

$$\varepsilon_k = \sum_{i=1}^{NS} \|\mathbf{p}^b_{(\mathbf{c}^k_{ab}(i))} - (\mathbf{A}^{k}_{ab} \cdot \mathbf{p}^a_{(i)} + \mathbf{t}^{k}_{ab})\|^2.$$

It is obvious that ε_k must be not greater than e_k. In the next iteration, the new individual correspondences $\mathbf{c}^{k+1}_{ab}(i)$, $i = 1, \ldots, NS$ will be constructed. At that time, the minimum SSR is:

$$e_{k+1} = \frac{1}{NS} \sum_{i=1}^{NS} \|\mathbf{p}^b_{(\mathbf{c}^{k+1}_{ab}(i))} - (\mathbf{A}^{k}_{ab} \cdot \mathbf{p}^a_{(i)} + \mathbf{t}^{k}_{ab})\|^2 \leq \varepsilon_k.$$

As a result, the following inequality can be obtained by iterating over the above process:

$$0 \leq \cdots \leq \varepsilon_{k+1} \leq e_{k+1} \leq \varepsilon_k \leq e_k \leq \cdots \leq e_1..$$

According to the monotone bounded theorem, OCAT is monotonically converged. □

A.2. The proof process of Theorem 2

Theorem 2. *Suppose that* $\hat{\mathbf{p}}^b_{(\mathbf{c}^k_{ab}(i))} = \mathbf{p}^b_{(\mathbf{c}^k_{ab}(i))} - \frac{1}{NS}\sum_{i=1}^{NS} \mathbf{p}^b_{(\mathbf{c}^k_{ab}(i))}$, $\hat{\mathbf{q}}^a_{(i)} = \mathbf{q}^a_{(i)} - \frac{1}{NS}\sum_{i=1}^{NS} \mathbf{q}^a_{(i)}$, *and the singular value decomposition result of* $\sum_{i=1}^{NS} \hat{\mathbf{p}}^a_{(i)} \cdot \hat{\mathbf{p}}^{b\;T}_{(\mathbf{c}^k_{ab}(i))}$ *is* $\mathbf{U} \cdot \mathbf{\Lambda} \cdot \mathbf{V}$, *then the optima of Eq.* (7) *are*

$$\mathbf{R}^k_{ab} = \mathbf{X} = \mathbf{V} \cdot \mathbf{U}^T$$

and

$$\mathbf{t}^k_{ab} = \frac{1}{NS}\sum_{i=1}^{NS} \mathbf{p}^b_{(\mathbf{c}^k_{ab}(i))} - \frac{1}{NS}\sum_{i=1}^{NS} \mathbf{V} \cdot \mathbf{U}^T \cdot \mathbf{q}^a_{(i)}.$$

Before the proof of Theorem 2, the following two lemmas are first provided:

Lemma 1. *For any two individual sets* $\{\mathbf{p}_i\}_{i=1,\ldots,NS}$ *and* $\{\mathbf{q}_i\}_{i=1,\ldots,NS}$, *the function* $f(\mathbf{t}) = \sum_{i=1}^{NS} \|\mathbf{p}_i - \mathbf{q}_i - \mathbf{t}\|^2$ *is minimized when* $\mathbf{t} = \frac{1}{NS}\sum_{i=1}^{NS} \mathbf{p}_i - \frac{1}{NS}\sum_{i=1}^{NS} \mathbf{q}_i$.

Proof. When $f(\mathbf{t})$ is minimized, its first derivative must equal zero, i.e., $\frac{df(\mathbf{t})}{d\mathbf{t}} = 2 \times \sum_{i=1}^{N}(\mathbf{p}_i - \mathbf{q}_i - \mathbf{t}) = 0$. Therefore, we can get $\mathbf{t} = \frac{1}{NS}\sum_{i=1}^{NS} \mathbf{p}_i - \frac{1}{NS}\sum_{i=1}^{NS} \mathbf{q}_i$ on this condition. □

Lemma 2. *Supposing that the matrix* $\mathbf{L} \cdot \mathbf{L}^T$ *is positive definite, then the inequality* $trace(\mathbf{L} \cdot \mathbf{L}^T) \geq trace(\mathbf{B} \cdot \mathbf{L} \cdot \mathbf{L}^T)$ *must hold for a random orthogonal matrix* ***B***.

Proof. Let $\mathbf{l}_i$ denote the ith column of $\mathbf{L}$, then we have

$$trace(\mathbf{B} \cdot \mathbf{L} \cdot \mathbf{L}^T) = trace(\mathbf{L}^T \cdot \mathbf{B} \cdot \mathbf{L}) = \sum_i \mathbf{l}_i^T \cdot (\mathbf{B} \cdot \mathbf{l}_i).$$

According to the Schwarz inequality $\mathbf{l}_i^T \cdot (\mathbf{B} \cdot \mathbf{l}_i) \leq \sqrt{(\mathbf{l}_i^T \cdot \mathbf{l}_i) \cdot (\mathbf{l}_i^T \cdot \mathbf{B}^T \cdot \mathbf{B} \cdot \mathbf{l}_i)} = \mathbf{l}_i^T \cdot \mathbf{l}_i$, it can be further deduced that

$$trace(\mathbf{B} \cdot \mathbf{L} \cdot \mathbf{L}^T) \leq \sum_i \mathbf{l}_i^T \cdot \mathbf{l}_i = trace(\mathbf{L} \cdot \mathbf{L}^T). \quad \square$$

According to Lemma 1, it can be known that $\mathbf{t}^k_{ab} = \frac{1}{NS}\sum_{i=1}^{NS} \mathbf{p}^b_{(\mathbf{c}^k_{ab}(i))} - \frac{1}{NS}\sum_{i=1}^{NS} \mathbf{R}^k_{ab}\mathbf{q}^a_{(i)}$, and thus Eq. (7) can be simplified as:

$$\mathbf{R}^k_{ab} = \underset{\mathbf{R}^T_{ab}\mathbf{R}_{ab}=\mathbf{I},\det(\mathbf{R}_{ab})=1}{\operatorname{argmin}} \left(\sum_{i=1}^{NS} \|\hat{\mathbf{p}}^b_{(\mathbf{c}^k_{ab}(i))} - \mathbf{R}_{ab} \cdot \hat{\mathbf{q}}^a_{(i)}\|^2\right), \tag{S2}$$

where $\hat{\mathbf{p}}^b_{(\mathbf{c}^k_{ab}(i))} = \mathbf{p}^b_{(\mathbf{c}^k_{ab}(i))} - \frac{1}{NS}\sum_{i=1}^{NS} \mathbf{p}^b_{(\mathbf{c}^k_{ab}(i))}$ and $\hat{\mathbf{q}}^a_{(i)} = \mathbf{q}^a_{(i)} - \frac{1}{NS}\sum_{i=1}^{NS} \mathbf{q}^a_{(i)}$. By expanding the corresponding objective function, we have

$$\mathbf{R}^k_{ab} = \underset{\mathbf{R}^T_{ab}\mathbf{R}_{ab}=\mathbf{I},\det(\mathbf{R}_{ab})=1}{\operatorname{argmin}} \sum_{i=1}^{NS}(\|\hat{\mathbf{p}}^b_{(\mathbf{c}^k_{ab}(i))}\|^2 + \|\hat{\mathbf{q}}^a_i\|^2 - 2\hat{\mathbf{p}}^{b\;T}_{(\mathbf{c}^k_{ab}(i))} \cdot \mathbf{R}_{ab} \cdot \hat{\mathbf{q}}^a_i).$$

Therefore, Eq. (S2) is equivalent to:

$$\begin{aligned} \mathbf{R}^k_{ab} &= \underset{\mathbf{R}^T_{ab}\cdot\mathbf{R}_{ab}=\mathbf{I},\det(\mathbf{R}_{ab})=1}{\operatorname{argmax}} \left(\sum_{i=1}^{NS} \hat{\mathbf{p}}^{b\;T}_{(\mathbf{c}^k_{ab}(i))} \cdot \mathbf{R}_{ab} \cdot \hat{\mathbf{q}}^a_i\right) \\ &= \underset{\mathbf{R}^T_{ab}\cdot\mathbf{R}_{ab}=\mathbf{I},\det(\mathbf{R}_{ab})=1}{\operatorname{argmax}} trace(\mathbf{R}_{ab} \cdot \mathbf{H}), \end{aligned}$$

where $\mathbf{H}$ denotes $\sum_{i=1}^{NS} \hat{\mathbf{p}}^a_{(i)} \cdot \hat{\mathbf{p}}^{b\;T}_{(\mathbf{c}^k_{ab}(i))}$. To solve this problem, we first conduct SVD on $\mathbf{H}$, i.e., $\mathbf{H} = \mathbf{U} \cdot \mathbf{\Lambda} \cdot \mathbf{V}$. Let $\mathbf{X} = \mathbf{V} \cdot \mathbf{U}^T$, then $\mathbf{X}$ must be orthogonal and $\mathbf{X} \cdot \mathbf{H} = \mathbf{V} \cdot \mathbf{U}^T \cdot \mathbf{U} \cdot \mathbf{\Lambda} \cdot \mathbf{V}^T = \mathbf{V} \cdot \mathbf{\Lambda} \cdot \mathbf{V}^T$ is symmetrical and positive definite. According to Lemma 2, for any orthogonal matrix $\mathbf{B}$, $trace(\mathbf{X} \cdot \mathbf{H}) \geq trace(\mathbf{B} \cdot \mathbf{X} \cdot \mathbf{H})$ must hold. Therefore, we have:

$$\mathbf{R}^k_{ab} = \mathbf{X} = \mathbf{V} \cdot \mathbf{U}^T.$$

Accordingly, the optimal translation vector is as follows:

$$\mathbf{t}^k_{ab} = \frac{1}{NS}\sum_{i=1}^{NS} \mathbf{p}^b_{(\mathbf{c}^k_{ab}(i))} - \frac{1}{NS}\sum_{i=1}^{NS} \mathbf{V} \cdot \mathbf{U}^T \cdot \mathbf{q}^a_{(i)}.$$

By now, the proof of Theorem 2 is completed.

References

[1] X. Xue, K. Zhang, R. Li, L. Zhang, C. Yao, J. Wang, J. Yao, A topology-based single-pool decomposition framework for large-scale global optimization, Appl. Soft Comput. 92 (2020) 106295, http://dx.doi.org/10.1016/j.asoc.2020.106295.

[2] M. Gong, H. Li, E. Luo, J. Liu, J. Liu, A multiobjective cooperative coevolutionary algorithm for hyperspectral sparse unmixing, IEEE Trans. Evol. Comput. 21 (2) (2017) 234–248, http://dx.doi.org/10.1109/TEVC.2016.2598858.

[3] Y. Liang, Z. Ren, L. Wang, H. Liu, W. Du, Surrogate-assisted cooperative signal optimization for large-scale traffic networks, Knowl.-Based Syst. 234 (2021) 107542, http://dx.doi.org/10.1016/j.knosys.2021.107542.

[4] D. Połap, M. Woźniak, Meta-heuristic as manager in federated learning approaches for image processing purposes, Appl. Soft Comput. 113 (2021) 107872, http://dx.doi.org/10.1016/j.asoc.2021.107872.

[5] I.A. Zamfirache, R.E. Precup, R.C. Roman, E.M. Petriu, Policy iteration reinforcement learning-based control using a grey wolf optimizer algorithm, Inform. Sci. 585 (2022) 162–175, http://dx.doi.org/10.1016/j.ins.2021.11.051.

[6] M. Köppen, K. Yoshida, K. Ohnishi, M. Tsuru, Meta-heuristic approach to proportional fairness, Evol. Inter. 5 (2012) 231–244, http://dx.doi.org/10.1007/s12065-012-0084-5.

[7] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359, http://dx.doi.org/10.1109/TKDE.2009.191.

[8] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, Proc. IEEE 109 (1) (2021) 43–76, http://dx.doi.org/10.1109/JPROC.2020.3004555.

[9] B. Gong, Y. Shi, F. Sha, K. Grauman, Geodesic flow kernel for unsupervised domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2066–2073, http://dx.doi.org/10.1109/CVPR.2012.6247911.

[10] B. Fernando, A. Habrard, M. Sebban, T. Tuytelaars, Unsupervised visual domain adaptation using subspace alignment, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 2960–2967, http://dx.doi.org/10.1109/ICCV.2013.368.

[11] A. Gupta, Y.S. Ong, L. Feng, Multifactorial evolution: Toward evolutionary multitasking, IEEE Trans. Evol. Comput. 20 (3) (2016) 343–357, http://dx.doi.org/10.1109/TEVC.2015.2458037.

[12] L. Feng, L. Zhou, J. Zhong, A. Gupta, Y.S. Ong, K.C. Tan, A.K. Qin, Evolutionary multitasking via explicit autoencoding, IEEE Trans. Cybern. 49 (9) (2019) 3457–3470, http://dx.doi.org/10.1109/TCYB.2018.2845361.

[13] K.C. Tan, L. Feng, M. Jiang, Evolutionary transfer optimization - A new frontier in evolutionary computation research, IEEE Comput. Intell. Mag. 16 (1) (2021) 22–33, http://dx.doi.org/10.1109/MCI.2020.3039066.

[14] T. Wei, S. Wang, J. Zhong, D. Liu, J. Zhang, A review on evolutionary multitask optimization: Trends and challenges, 26, (5) 2022, pp. 941–960, http://dx.doi.org/10.3390/10.1109/TEVC.2021.3139437.

[15] A. Gupta, Y.S. Ong, L. Feng, K.C. Tan, Multiobjective multifactorial optimization in evolutionary multitasking, IEEE Trans. Cybern. 47 (7) (2017) 1652–1665, http://dx.doi.org/10.1109/TCYB.2016.2554622.

[16] K.K. Bali, Y.S. Ong, A. Gupta, P.S. Tan, Multifactorial evolutionary algorithm with online transfer parameter estimation: MFEA-II, IEEE Trans. Evol. Comput. 24 (1) (2020) 69–83, http://dx.doi.org/10.1109/TEVC.2019.2906927.

[17] K.K. Bali, A. Gupta, L. Feng, Y.S. Ong, T.P. Siew, Linearized domain adaptation in evolutionary multitasking, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2017, pp. 1295–1302, http://dx.doi.org/10.1109/CEC.2017.7969454.

[18] Y. Chen, J. Zhong, L. Feng, J. Zhang, An adaptive archive based evolutionary framework for many-task optimization, IEEE Trans. Emerg. Topics Comput. Intell. 4 (3) (2020) 369–384, http://dx.doi.org/10.1109/TETCI.2019.2916051.

[19] Z. Liang, X. Xu, L. Liu, Y. Tu, Z. Zhu, Evolutionary many-task optimization based on multi-source knowledge transfer, IEEE Trans. Evol. Comput. 26 (2) (2022) 319–333, http://dx.doi.org/10.1109/TEVC.2021.3101697.

[20] G. Li, Q. Lin, W. Gao, Multifactorial optimization via explicit multipopulation evolutionary framework, Inform. Sci. 512 (2020) 1555–1570, http://dx.doi.org/10.1016/j.ins.2019.10.066.

[21] J. Li, H. Li, Y. Liu, M. Gong, Multi-fidelity evolutionary multitasking optimization for hyperspectral endmember extraction, Appl. Soft Comput. 111 (2021) 107713, http://dx.doi.org/10.1016/j.asoc.2021.107713.

[22] T.B. Thang, N.B. Long, N.V. Hoang, H.T.T. Binh, Adaptive knowledge transfer in multifactorial evolutionary algorithm for the clustered minimum routing cost, Appl. Soft Comput. 105 (2021) 107253, http://dx.doi.org/10.1016/j.asoc.2021.107253.

[23] B. Da, A. Gupta, Y.S. Ong, Curbing negative influences online for seamless transfer evolutionary optimization, IEEE Trans. Cybern. 49 (12) (2019) 4365–4378, http://dx.doi.org/10.1109/TCYB.2018.2864345.

[24] Y. Luo, Y. Wen, D. Tao, Heterogeneous multitask metric learning across multiple domains, IEEE Trans. Neural Netw. Learn. Syst. 29 (9) (2018) 4051–4064, http://dx.doi.org/10.1109/TNNLS.2017.2750321.

[25] J. Zhang, W. Zhou, X. Chen, W. Yao, L. Cao, Multisource selective transfer framework in multiobjective optimization problems, IEEE Trans. Evol. Comput. 24 (3) (2020) 424–438, http://dx.doi.org/10.1109/TEVC.2019.2926107.

[26] L. Zhou, L. Feng, A. Gupta, Y.S. Ong, Learnable evolutionary search across heterogeneous problems via kernelized autoencoding, IEEE Trans. Evol. Comput. 25 (3) (2021) 567–581, http://dx.doi.org/10.1109/TEVC.2021.3056514.

[27] R. Lim, A. Gupta, Y.S. Ong, L. Feng, A.N. Zhang, Non-linear domain adaptation in transfer evolutionary optimization, Cogn. Comput. 13 (2021) 290–307, http://dx.doi.org/10.1007/s12559-020-09777-7.

[28] J. Ding, C. Yang, Y. Jin, T. Chai, Generalized multitasking for evolutionary optimization of expensive problems, IEEE Trans. Evol. Comput. 23 (1) (2019) 44–58, http://dx.doi.org/10.1109/TEVC.2017.2785351.

[29] X. Xue, K. Zhang, K.C. Tan, L. Liang, J. Wang, G. Chen, X. Zhao, L. Zhang, J. Yao, Affine transformation-enhanced multifactorial optimization for heterogeneous problems, IEEE Trans. Cybern. 52 (7) (2022) 6217–6231, http://dx.doi.org/10.1109/TCYB.2020.3036393.

[30] Z. Chen, Y. Zhou, X. He, J. Zhang, Learning task relationships in evolutionary multitasking for multiobjective continuous optimization, IEEE Trans. Cybern. 52 (6) (2022) 5278–5289, http://dx.doi.org/10.1109/TCYB.2020.3029176.

[31] Z. Tang, M. Gong, Y. Wu, W. Liu, Y. Xie, Regularized evolutionary multitask optimization: Learning to intertask transfer in aligned subspace, IEEE Trans. Evol. Comput. 25 (2) (2021) 262–276, http://dx.doi.org/10.1109/TEVC.2020.3023480.

[32] W. Gao, J. Cheng, M. Gong, H. Li, J. Xie, Multiobjective multitasking optimization with subspace distribution alignment and decision variable transfer, IEEE Trans. Emerg. Topics Comput. Intell. 5 (4) (2021) 818–827, http://dx.doi.org/10.1109/TETCI.2021.3115518.

[33] P.J. Besl, N.D. McKay, A method for registration of 3-D shapes, IEEE Trans. Pattern Anal. Mach. Intell. 14 (2) (1992) 239–256, http://dx.doi.org/10.1109/34.121791.

[34] S. Du, G. Xu, S. Zhang, X. Zhang, Y. Gao, B. Chen, Robust rigid registration algorithm based on pointwise correspondence and correntropy, Pattern Recognit. Lett. 132 (2020) 91–98, http://dx.doi.org/10.1016/j.patrec.2018.06.028.

[35] Y. Yang, D. Fan, S. Du, M. Wang, B. Chen, Y. Gao, Point set registration with similarity and affine transformations based on bidirectional KMPE loss, IEEE Trans. Cybern. 51 (3) (2021) 1678–1689, http://dx.doi.org/10.1109/TCYB.2019.2944171.

[36] P. Larrañaga, J.A. Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Springer Science & Business Media, 2001.

[37] P.A.N. Bosman, J. Grahl, D. Thierens, Enhancing the performance of maximum–likelihood Gaussian EDAs using anticipated mean shift, in: Proceedings of the International Conference on Parallel Problem Solving from Nature Springer, 2008, pp. 133–143, http://dx.doi.org/10.1007/978-3-540-87700-4_14.

[38] Z. Ren, Y. Liang, L. Wang, A. Zhang, B. Pang, B. Li, Anisotropic adaptive variance scaling for Gaussian estimation of distribution algorithm, Knowl.-Based Syst. 146 (2018) 142–151, http://dx.doi.org/10.1016/j.knosys.2018.02.001.

[39] D. Kalyanmoy, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, http://dx.doi.org/10.1109/4235.996017.

[40] X. Zheng, A.K. Qin, M. Gong, D. Zhou, Self-regulated evolutionary multitask optimization, IEEE Trans. Evol. Comput. 24 (1) (2020) 16–28, http://dx.doi.org/10.1109/TEVC.2019.2904696.

[41] J. Lin, H. Liu, K.C. Tan, F. Gu, An effective knowledge transfer approach for multiobjective multitasking optimization, IEEE Trans. Cybern. 51 (6) (2021) 3238–3248, http://dx.doi.org/10.1109/TCYB.2020.2969025.

[42] D. Wu, X. Tan, Multitasking genetic algorithm (MTGA) for fuzzy system optimization, IEEE Trans. Fuzzy Syst. 28 (6) (2020) 1050–1061, http://dx.doi.org/10.1109/TFUZZ.2020.2968863.

[43] J.Y. Li, Z.H. Zhi, K.C. Tan, J. Zhang, A meta-knowledge transfer-based differential evolution for multitask optimization, IEEE Trans. Evol. Comput. 26 (4) (2022) 719–734, http://dx.doi.org/10.1109/TEVC.2021.3131236.

[44] R. Liaw, C. Ting, Evolutionary manytasking optimization based on symbiosis in biocoenosis, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 4295–4303, http://dx.doi.org/10.1609/aaai.v33i01.33014295.

[45] L.T. Thanh, L.V. Cuong, T.B. Thang, H.T.T. Binh, Multi-armed bandits for many-task evolutionary optimization, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2021, pp. 1664–1671, http://dx.doi.org/10.1109/CEC45853.2021.9504691.

[46] Y. Jiang, Z.H. Zhan, K.C. Tan, J. Zhang, A bi-objective knowledge transfer framework for evolutionary many-task optimization, IEEE Trans. Evol. Comput. (2022) http://dx.doi.org/10.1109/TEVC.2022.3210783, in press.

[47] B. Da, Y.S. Ong, L. Feng, A.K. Qin, A. Gupta, Z. Zhu, C.K. Ting, K. Tang, X. Yao, Evolutionary multitasking for single-objective continuous optimization: Benchmark problems, performance metrics and baseline results, Tech. Rep., Nanyang Technol. Univ, 2016, Available at: http://www.bdsc.site/websites/MTO/index.html.

[48] Y. Yuan, Y.S. Ong, L. Feng, A.K. Qin, A. Gupta, B. Da, Q. Zhang, K.C. Tan, Y. Jin, H. Ishibuchi, Evolutionary Multitasking for Multiobjective Continuous Optimization: Benchmark Problems, Performance Metrics and Baseline Results, Tech. Rep., Nanyang Technol. Univ, 2016, Available at http://www.bdsc.site/websites/MTO/index.html.

[49] D.J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, CRC Press, 2003.

[50] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V.G. Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, IEEE Trans. Evol. Comput. 7 (2) (2003) 117–132, http://dx.doi.org/10.1109/TEVC.2003.810758.

[51] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V.G. Fonseca, WCCI2020 Competition on Evolutionary Multi-task Optimization, http://www.bdsc.site/websites/MTO_competition_2020/MTO_Competition_WCCI_2020.html.