This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2021.3101697, IEEE Transactions on Evolutionary Computation

Evolutionary Many-task Optimization Based on Multi-source Knowledge Transfer

Zhengping Liang, Xiuju Xu, Ling Liu, Yaofeng Tu, and Zexuan Zhu, *Senior Member, IEEE*

***Abstract*—Multi-task optimization aims to solve two or more optimization tasks simultaneously by leveraging inter-task knowledge transfer. However, as the number of tasks increases to the extent of many-task optimization, the knowledge transfer between tasks encounters more uncertainty and challenges, thereby resulting in degradation of optimization performance. To give full play to the many-task optimization framework and minimize the potential negative transfer, this paper proposes an evolutionary many-task optimization algorithm based on a multi-source knowledge transfer mechanism namely EMaTO-MKT. Particularly, in each iteration, EMaTO-MKT determines the probability of using knowledge transfer adaptively according to the evolution experience, and balances the self-evolution within each task and the knowledge transfer among tasks. To perform knowledge transfer, EMaTO-MKT selects multiple highly similar tasks in term of maximum mean discrepancy as the learning sources for each task. Afterward, a knowledge transfer strategy based on local distribution estimation is applied to enable the learning from multiple sources. Compared with the other state-of-the-art evolutionary many-task algorithms on benchmark test suites, EMaTO-MKT shows competitiveness in solving many-task optimization problems.**

***Index Terms*—Evolutionary many-task optimization, Maximum mean discrepancy, Local distribution estimation, Multi-source knowledge transfer.**

I. Introduction

EVOLUTIONARY algorithms (EAs) are population-based optimization algorithms capable of obtaining multiple solutions of a target problem in a single run [1]–[3]. They have achieved widely successes in various complex application problems [4]–[7]. Traditional EAs tend to solve one single problem from scratch by assuming zero prior knowledge. However, since complex real-world optimization problems seldom appear in isolation, knowledge learned from previous optimization exercises or related problems can be exploited to facilitate the solution of the target problems. Inspired by the parallel processing of multiple problems in human brain, Gupta et al. [8], [9] proposed a paradigm namely evolutionary multi-task optimization (EMTO) to solve multiple optimization problems simultaneously. Compared with the traditional evolutionary single-task optimization, EMTO can achieve better performance in solving correlated optimization problems by leveraging knowledge transfer among the problems [10]–[14]. Nevertheless, as the number of optimization tasks increases to the extent of many-task optimization (MaTO) [15] (the number of tasks exceeds three), the majority of EMTO algorithms face big challenges in computational resource allocation, larger-scale knowledge transfer, and task selection for knowledge transfer [16]–[18]. More specifically, in MaTO, more efforts should be put into balancing the computational budgets allocated to the intra-task optimization and inter-task knowledge transfer. New knowledge transfer mechanism is required to enable the efficient knowledge transfer among a larger number of tasks, where proper selection of participant tasks is the key to the efficiency of knowledge transfer.

A few specific evolutionary many-task optimization (EMaTO) algorithms have been proposed to solve the aforementioned issues. For example, GMFEA [16] uses clustering method to choose task for knowledge transfer. EEMTA [19] performs task selection for knowledge transfer via feedback based credit allocation method. SaEF-AKT [20] adopts Kullback-Leibler divergence (KLD) and pheromone based method to identify tasks for knowledge transfer. MaTEA [17] transfers knowledge across the tasks selected according to the feedback information of evolutionary process and KLD. To allocate computational resource, MaTEA also introduces a fixed probability to control the intra-task optimization and knowledge transfer among tasks. EBS [15] scales up the knowledge transfer by concatenating offspring to share the knowledge of all tasks. The existing EMaTO algorithms have made substantial progress in solving a portion of the challenges of MaTO, yet there remains much room for a more comprehensive solution that can take all the challenges into consideration.

To handle MaTO problems more efficiently, this paper proposes an evolutionary many-task optimization algorithm with multi-source knowledge transfer namely EMaTO-MKT. Particularly, the multi-source knowledge transfer mechanism consists of an adaptive mating probability (AMP) strategy, a maximum mean discrepancy (MMD) [21] based task se-

Manuscript received XX. XX. XXX; revised XX. XX. XXX; accepted XX. XX. XXX. This work was supported in part by the National Natural Science Foundation of China under Grant 61871272 and Grant 62001300; in part by the National Natural Science Foundation of Guangdong, China, under Grant 2020A1515010479, Grant 2021A1515011911, and Grant 2021A1515011679; in part by the Guangdong Provincial Key Laboratory under Grant 2020B121201001; in part by the Shenzhen Fundamental Research Program under Grant 20200811181752003 and Grant JCYJ20190808173617147; and in part by the BGI-Research Shenzhen Open Funds under Grant BGIRSZ20200002. This article was recommended by XXXXXXXX. *(Corresponding authors: Ling Liu; Zexuan Zhu.)*

Z. Liang, X. Xu, and L. Liu are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: liangzp@szu.edu.cn; xuxiuju2018@email.szu.edu.cn; liulingcs@szu.edu.cn).

Y. Tu is with Central R&D Institute, ZTE Corporation, Shenzhen 518057, China (e-mail: tu.yaofeng@zte.com.cn).

Z. Zhu is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, also with Shenzhen Pengcheng Laboratory, Shenzhen 518055, China, and also with the Guangdong Provincial Key Laboratory of Brain-Inspired Intelligent Computation, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: zhuzx@szu.edu.cn).

lection (MTS) strategy, and a local distribution estimation based knowledge transfer (LEKT) strategy. The AMP strategy estimates the current evolution trend of each task by learning the experience in the process of evolution, and calculates the probability of generating offspring for each task, which is conducive to the balance of self-evolution within each task and the knowledge transfer among tasks. The MTS strategy uses MMD to calculate the difference between the decision variable distributions of different task populations, and selects appropriate tasks to participate in knowledge transfer and relieve negative transfer [22]. The LEKT strategy supports knowledge transfer across any number of tasks. The union set of each task population participating in knowledge transfer is firstly divided by clustering method, and then a probability model is constructed for each sub-population through distribution estimation [23], based on which the offspring individuals are generated to accelerate the convergence and maintain the diversity of the population. To verify the effectiveness of EMaTO-MKT, EMaTO-MKT is compared with the other state-of-the-art EMaTO algorithms on two sets of single-objective MaTO problems and two sets of multi-objective MaTO problems. EMaTO-MKT shows good competitiveness in the comparison studies. The contributions of this paper are highlighted as follows:

1) The AMP strategy introduces adaptive frequency of knowledge transfer on the basis of evolutionary experience, which leads to better computational resource allocation and population convergence.
2) The MTS strategy based on MMD serves as a good solution for task selection in knowledge transfer and the experimental results demonstrate that it can substantially reduce negative transfer.
3) To the best of our knowledge, the LEKT strategy represents the first attempt to enable knowledge transfer of arbitrary number of tasks. The strategy can take full advantage of involving more tasks in MaTO.

The rest of this paper is structured as follows. Section II presents the preliminaries and literature review on related work. Sections III details the proposed algorithm. Section IV describes the experiment study. Finally, Section V concludes this paper and discusses the potential future work.

II. Preliminaries and Literature Review

To facilitate the understanding of the proposed EMaTO-MKT, the preliminaries of multi-task optimization (MTO) and the literature review on related work are provided in this section.

A. Multi-task Optimization

MTO [24]–[26] refers to the simultaneous optimization of multiple self-contained tasks or problems by exploiting synergies existing among them. It is closely related to transfer learning [22] and multi-task learning [27]. In machine learning, the optimization of learning model usually calls for a large volume of data. If the knowledge of other models in the related learning tasks can be reused, we can save efforts in data collection and learning process [28]. Accordingly, knowledge transfer is introduced in transfer learning and multi-task learning. Transfer learning uses the knowledge obtained from one or more source tasks to improve the learning performance of the target task, whereas multi-task learning establishes knowledge transfer among different tasks of equal priority aiming at improving the learning performance of all tasks at the same time. Extending the knowledge transfer principle of transfer learning and multi-task learning to the field of optimization leads to MTO, which focuses more on exploiting shared knowledge to improve problem-solving rather than learning [29].

Without loss of generality, a conventional optimization problem can be defined as follows:

$$\begin{aligned} &\min \mathbf{F}(\mathbf{x}) = \min \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})\right) \\ &\text{subject to } : \mathbf{x} \in \mathbf{R}^D \end{aligned} \tag{1}$$

where $\mathbf{x} = (x_1, x_2, ..., x_D)$ denotes a D-dimension decision variable in $\mathbf{R}$, $f_i(\mathbf{x})$ indicates the i-th objective function, and M is the number of objective functions. The problem is called a single objective optimization (SOO) problem if $M = 1$. If $M > 1$, the problem is referred to as a multi-objective optimization (MOO) problem. In MOO problem, a solution $\mathbf{x_1}$ is said to dominate $\mathbf{x_2}$, if $\forall i \in \{1, 2, \ldots, M\}, f_i(\mathbf{x_1}) \leq f_i(\mathbf{x_2})$ and $\exists j \in \{1, 2, \ldots, M\}, f_j(\mathbf{x_1}) < f_j(\mathbf{x_2})$. A solution not dominated by any other solution is called a Pareto optimal solution. All Pareto optimal solutions forms the Pareto optimal set (PS) of which the mapping in objective space is called Pareto front (PF).

Based on the definition of conventional optimization, an MTO problem can be formulated as:

$$\begin{aligned} \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_K\} = \{&\operatorname{argmin} \mathbf{F}_1(\mathbf{X}_1), \\ &\operatorname{argmin} \mathbf{F}_2(\mathbf{X}_2), \ldots, \operatorname{argmin} \mathbf{F}_K(\mathbf{X}_K)\} \end{aligned} \tag{2}$$

where $\mathbf{F}_i$ denotes the i-th optimization task defined in Eq. (1), $\mathbf{X}_i$ indicates the solution set of task i (for SOO tasks, the solution set might contain only one solution), and K is the number of tasks. An MTO problem is also called a many-task optimization (MaTO) problem if $K > 3$. To solve MTO/MaTO problems with EAs, some new properties should be defined, i.e.,

1) Factorial Rank: the factorial rank of an individual p on task i is the rank of p in all solutions in terms of $\mathbf{F}_i$ (in MOO problems, the sorting can be achieved via non-dominate sorting like NSGA-II [30]).
2) Skill Factor: the skill factor of an individual p is the task on which p obtains the best factorial rank.
3) Scalar fitness: given the factorial rank φ of an individual p on its skill factor task, the scalar fitness of p is defined as $1/\varphi$.

Moreover, to solve different tasks in the same environment, a unified representation space is usually established for all tasks in MTO/MaTO. Assuming D_i is the dimension of the search space of task i, then the dimension of the unified space $\mathbf{Y}$ is $D_{\mathbf{Y}} = \max\{D_i\}$, $i = 1, ..., K$, where $\mathbf{Y}$ is confined to the range $[0, 1]^D$, i.e.,

$$y_i = (x_i - L_i)/(U_i - L_i) \tag{3}$$

where x_i is the value of an individual in the i-th dimension of the original space, U_i and L_i represent the upper and lower bounds of the i-th dimension, respectively, and y_i is the mapped value of x_i in the unified space $\mathbf{Y}$. In fitness evaluation, y_i is decoded to x_i accordingly as follows:

$$x_i = L_i + y_i \times (U_i - L_i). \tag{4}$$

B. Related Work

In recent years, EMTO algorithms have attracted increasing attention and been widely used in solving different problems. Among them, MFEA [8] and MOMFEA [9] are the most widely used single-objective and multi-objective versions of implementation, respectively. Specifically, the knowledge transfer is achieved through the interaction of two features, namely, assortative mating and vertical cultural transmission. Based on MFEA and MOMFEA, many extensions have been proposed in the literature. For example, to solve the expensive optimization problem, Ding et al. [31] introduced a generalized evolutionary multi-task algorithm, which contains two strategies namely decision variable translation strategy and decision variable shuffling strategy. The algorithm achieves good results in promoting computationally expensive tasks (E-tasks) by computationally cheap tasks (C-tasks). Different from the previous EMTO algorithms using implicit knowledge transfer, Feng et al. [32] put forward explicit knowledge transfer by using different search mechanisms to construct mapping relationship via autoencoding. Gong et al. [33] presented an EMTO algorithm with dynamic computational resource allocation. Bali et al. [34] introduced a new multi-task framework (MFEA-II) capable of online learning and utilizing the similarities (and differences) of different tasks to enhance the optimization process. Zhou et al. [35] investigated the performance of different benchmark problems with different crossover operators and proposed MFEA-AKT, which adaptively configures the crossover operator for knowledge transfer based on the information collected in evolution process. Liang et al. [18] introduced MOMFEA-SADE that uses a subspace alignment strategy to minimize the difference between two task populations and employs evolutionary experience to select appropriate differential evolution (DE) strategy to balance the population convergence and diversity.

EMTO has also been successfully applied to many practical problems. For example, Zhou et al. [10] proposed P-MFEA to solve the constrained vehicle routing problem in multi-task environment. A unified representation method based on disturbance and a new decoding operator are designed to improve the performance of the algorithm in solving vehicle routing problem. Binh et al. applied MFEA to solve the cluster shortest path tree problem (CSTP) in [12]. Chandra et al. [36] introduced a Bayesian based approach to solve the dynamic time series prediction problem for multi-task environments. Tang et al. [37] suggested an EMTO algorithm to train neuron networks with different numbers of hidden neurons for classification problem.

Although many algorithms of promising performance have emerged in the field of EMTO, relatively few EMaTO algorithms have been proposed. For example, Liaw et al. [15] proposed an evolutionary symbiosis based many-task optimization framework (EBS). EBS adaptively adjusts the frequency of information transmission within the task according to the effect of knowledge transfer, and uses the set of offspring of all tasks to transfer knowledge. Tang et al. [16] introduced an EMaTO algorithm namely GMFEA that adopts the method of selecting tasks in groups for gene transfer. Chen et al. [17] designed an archive-based many-task evolutionary algorithm (MaTEA), which mainly includes a new crossover operator to achieve knowledge transfer and adaptive selection of candidate solutions based on auxiliary tasks. Huang et al. [20] put forward a surrogate-assisted many-task evolutionary framework (SaEF-AKT). SaEF-AKT updates the optimization function of Gaussian process based on historical experience. Meanwhile, it uses KLD and pheromone based method to select tasks for knowledge transfer, which can better deal with the problem of high computational cost [38]. Jin et al. [39] presented a general DE framework for many-task optimization, which mainly contains a knowledge transfer strategy based on elite set and a knowledge reuse strategy by using DE operator. These knowledge can accelerate the population convergence and expand the search space of the algorithm. Based on explicit knowledge transfer, Shang et al. [19] implemented the explicit EMT algorithm (EEMTA) for many-task optimization. The feedback based credit allocation method is used for task selection. The feedback information is dynamically updated during the search process to capture the useful knowledge between tasks, which can effectively guide the population search.

The existing EMaTO algorithms are focused on dealing part of the challenges of MaTO problems namely computational resource allocation, larger-scale knowledge transfer, and task selection for knowledge transfer. There is a lack of a method that can take care of all the challenges in a comprehensive fashion. This work proposes a new EMaTO algorithm EMaTO-MKT to fill the gap by addressing all the issues with the AMP, MTS, and LEKT strategies. The details of the proposed algorithm are provided in the following section.

III. Proposed Algorithm

A. General Framework

The general framework of EMaTO-MKT is shown in Fig. 1, where the key component operations including the three strategies are outlined. More details are available in the pseudo code shown in **Algorithm 1**.

Firstly, a population of size N is initialized and the individuals in the population are evaluated. The individuals of different tasks are mapped to a unified space as defined in Eq. (3) and each individual is randomly assigned with a skill factor. At the beginning of each iteration, the AMP strategy is used to calculate the probability amp_i of each task to generate offspring by knowledge transfer (line 3), and the MMD values between tasks are calculated (line 4). Before offspring generation, the MTS strategy is used to select l tasks for each task to prepare the knowledge transfer (line 6). On this basis, the LEKT strategy is used to construct the probability model of each subpopulation (line 7). Then, two different methods are used to generate N/K offspring for each task (line 8-18). If a generated random number $rand > amp_i$, two offspring are generated by crossover and mutation through self evolution within the task (line 11-12). If the random number $rand \leq amp_i$, the knowledge transfer between tasks is adopted, and two offspring are generated based on the probability model of the cluster corresponding to the selected parent (line 14). Finally, an environmental selection is carried out on the union of the parent population **P** and the offspring population **O** to generate the next generation population by environmental selection (line 20). Once the terminal condition is reached, the final solutions are obtained. The proposed three strategies including AMP, MTS, and LEKT are detailed in the following subsections.

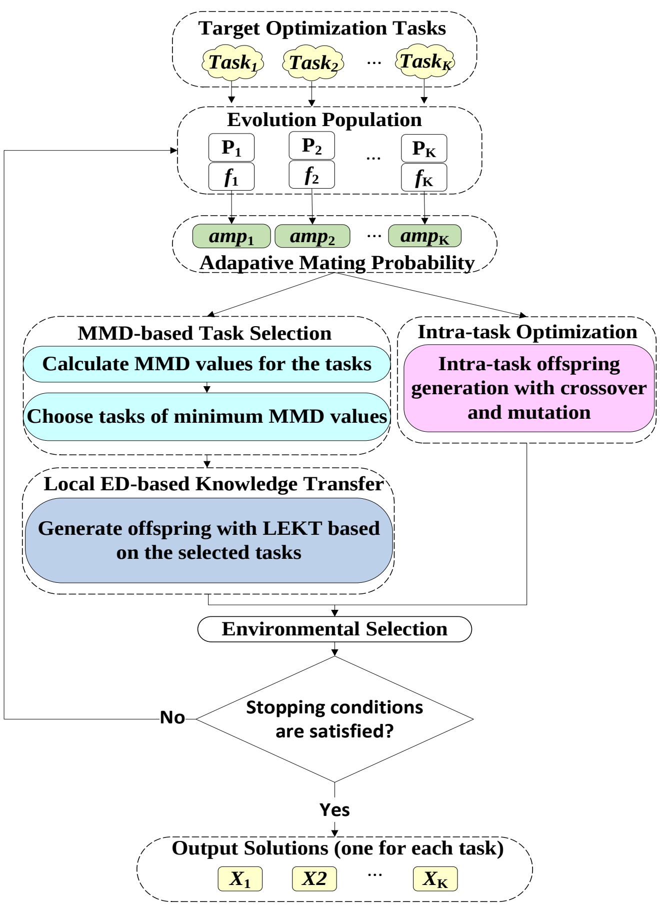


Fig. 1. The flowchart of the proposed algorithm.

Algorithm 1 The Procedure of EMaTO-MKT

Input: N (population size), K (number of tasks), C (number of clusters for LEKT), l (number of selected tasks for knowledge transfer)

Output: A series of best solutions

1: Initialize and evaluate population $\boldsymbol{P}$;
2: **while** *the termination conditions are not met* **do**
3: Obtain $\boldsymbol{amp}$ by AMP strategy;
4: Calculate MMD values between tasks based on Eqs. (9)-(14);
5: **for** $i = 1 \rightarrow K$ **do**
6: Choose l tasks to form the task list $\boldsymbol{L}$ for knowledge transfer using MTS strategy;
7: $(\bar{\boldsymbol{x}}_{\boldsymbol{i}}, \boldsymbol{Cov_i})$ = LEKT($\boldsymbol{P}$, i, C, $\boldsymbol{L}$) $\leftarrow$ **Algorithm 2**
8: **for** $j = 1 \rightarrow N/K/2$ **do**
9: Choose the $j\text{-}th$ individual p_i^j from task i as one parent and locate its cluster number in LEKT as c;
10: **if** $rand > amp_i$ **then**
11: Randomly select an individual from task i as another parent;
12: Perform crossover and mutation on the parents to generate two offspring o_1 and o_2;
13: **else**
14: Generate two offspring o_1 and o_2 via sampling from $\bar{\boldsymbol{x}}_{\boldsymbol{i}}^{\boldsymbol{c}}$ and $\boldsymbol{Cov}_{\boldsymbol{i}}^{\boldsymbol{c}}$;
15: **end if**
16: Evaluate o_1 and o_2 and let them inherit the skill factor of p_i^j;
17: Add o_1 and o_2 into the offspring population $\boldsymbol{O}$;
18: **end for**
19: **end for**
20: Apply environmental selection to $\boldsymbol{P} \cup \boldsymbol{O}$;
21: **end while**

B. Adaptive Mating Probability Strategy

In MTO/MaTO, if the search convergence of a task is accelerating, the odds of knowledge transfer should be appropriately reduced and the offspring should be generated mainly by self evolution within the current task. As such, the influence of potential negative transfer on the current convergence trend can be relieved. On the contrary, if the search convergence of a task slows down, the convergence of the task can be accelerated or the population can be dragged out of the potential local optimum by increasing the knowledge transfer probability. To reasonably control the probability of self evolution and knowledge transfer, empirical knowledge in the evolution process is used to estimate the convergence trend of the current task. Particularly, the current convergence trend of the task is estimated by comparing the best objective function value of the current population with that of the previous two generations. Taking the SOO problem as an example, let G denote the current generation, the best function values on task i in the current generation and the two previous generations are recorded as F_G^i, F_{G-1}^i, and F_{G-2}^i, respectively. We can get the differences of the best function values among three generations according to the equations

$$d_{1,G}^i = \left|F_G^i - F_{G-1}^i\right| \tag{5}$$

and

$$d_{2,G}^i = \left|F_{G-1}^i - F_{G-2}^i\right|, \tag{6}$$

where $|\cdot|$ calculates the absolute value. Note that for MOO problems, the F_G^i value is defined as the average Euclidean distance of all individuals to the origin in the objective space.

If $d_{1,G}^i \geq d_{2,G}^i$, the convergence of task i is accelerating, and staying in its own way tend to obtain better convergence.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2021.3101697, IEEE Transactions on Evolutionary Computation

Conversely, if $d^i_{1,G} < d^i_{2,G}$, the convergence of task i slows down, and thus transferring information from other tasks is more likely to improve the convergence of the task. It is the relative magnitude rather than the absolute values of $d_{1,G}$ and $d_{2,G}$ that matters in AMP strategy. The distance of the optimal solutions to the original could affect the values of $d_{1,G}$ and $d_{2,G}$, but it does not affect the use of the strategy.

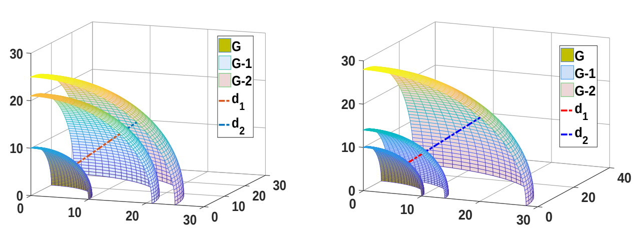
Based on $d_{1,G}$ and $d_{2,G}$, an adaptive probability amp^i_G of learning knowledge from other tasks in generation G for task i is defined as follows:

$$amp^i_G = \frac{d^i_{2,G}}{d^i_{1,G} + d^i_{2,G}}, i = 1, 2, \ldots, K \tag{7}$$

where amp^i_G is made up of $d^i_{1,G}$ and $d^i_{2,G}$ in proportion. The numerator represents the evolution speed of knowledge transfer. A large $d^i_{2,G}$ suggests that self evolution is shrinking and the probability of knowledge transfer should be raised for more communication. On the contrary, a small $d^i_{2,G}$ indicates that the search is converged toward a promising direction and the use of knowledge transfer should be refrained.

An MOO example is shown in Fig. 2 to illustrate the update of amp^i_G. According to Eqs. (5) and (6), $d^i_{1,G}$ and $d^i_{2,G}$ are obtained based on the population distributions of the successive three generations. The current probability of knowledge transfer amp^i_G can be calculated by Eq. (7). Fig. 2 (a) presents the scenario when $d^i_{2,G} \le d^i_{1,G}$. In this case, the convergence of this task is accelerating. Increasing the probability of self evolution may be conducive to its continued rapid convergence. Fig. 2 (b) displays the scenario when $d^i_{2,G} > d^i_{1,G}$. In this case, the convergence rate of the task is decreasing. Increasing the probability of knowledge transfer between tasks could help the subsequent evolution.

Unlike the rmp used in MFEA [8], AMP strategy selects appropriate knowledge transfer probability for each task according to the evolution trend in the many-task environment, which is simple and effective. The evolution processes of the tasks are different. The adaptive amp^i_G enables each task to find its own evolution pace and properly balances the frequency of knowledge transfer and self-evolution.



(a) (b)

Fig. 2. The illustration of $d^i_{1,G}$ and $d^i_{2,G}$ on MOO problem. (a) The scenario of $d^i_{2,G} \le d^i_{1,G}$. (b) The scenario of $d^i_{2,G} > d^i_{1,G}$.

C. MMD-based Task Selection Strategy

Compared with MTO, MaTO can solve more optimization tasks at the same time. Each task can obtain knowledge from more other tasks. Based on positive knowledge transfer between more tasks, the quality of solution space can be improved in theory. However, as the number of tasks increases, the consensus among tasks decreases, and the probability of negative transfer in the process of knowledge transfer increases. Therefore, to improve the performance of MaTO algorithms, it is necessary to give full play to the advantages of many tasks, which refers to efficient knowledge transfer between more tasks. The single knowledge transfer in the existing algorithms where the number of tasks is limited to 2-3 is not applicable to MaTo problems. Moreover, highly similar tasks should be selected for knowledge transfer rather than the low efficient random task selection.

The similarity between tasks can be measured based on the distribution of the decision variable set associated with the corresponding populations in the decision space. For low dimensional decision space, Euclidean distance can be used to approximately measure the distribution difference of two sets of decision variables. However, for the high-dimensional decision space faced in MaTO, it may not be able to accurately reflect the difference between high-dimensional decision variables. Maximum mean discrepancy (MMD) [21] has been successfully applied as distance measurement between high-dimensional distributions in various applications including feature extraction [40], target detection [41] and speaker verification system [42]. MMD measures the distance between two distributions in the reproducing kernel Hilbert space (RKHS), a kind of Hilbert space with reproducing kernel, which can regenerate the inner product of two functions. Calculating the distance in RKHS can more accurately reflect the difference between the two distributions in high-dimensional space.

Inspired by this, we use MMD to calculate the distance of the populations associated with different tasks in the decision space, and obtain the similarity between tasks based on the MMD value. More specifically, if the populations of two tasks have similar distributions, the two tasks may have similar evolutionary process and evolutionary trend to a certain extent. Therefore, the similarity of population distributions can reflect the similarity of tasks, which has also been justified in [17], [20]. Compared with the widely used measurements like KLD and Jensen-Shannon distance (JSD) in evolutionary computation, MMD works better in many-task evolutionary optimization, as demonstrated in Section IV-E. A smaller MMD value indicates less difference of the population distributions of two tasks, i.e., there are more similar search preferences between the two tasks.

In each iteration, the MMD value of every task pair is calculated as follows. Assume P and Q are the probability distributions of two tasks, two corresponding populations, i.e., $\mathbf{X} = (x_1, ..., x_u)$ and $\mathbf{Y} = (y_1, ..., y_v)$ containing identically distributed samples, are drawn from P and Q, respectively. Let $\mathcal{F}$ be a class of functions $f : \mathcal{X} \rightarrow \mathcal{R}$ that map the original sample space $\mathcal{X}$ to the set of real numbers $\mathcal{R}$. In this study, $\mathcal{F}$ is chosen as the unit ball of RKHS $\mathcal{H}$, as such, $f(x)$ is expressed as an inner product via

$$f(x) = \langle f, \phi(x) \rangle_{\mathcal{H}} \tag{8}$$

where $\phi : \mathcal{X} \rightarrow \mathcal{H}$ is a mapping from $\mathcal{X}$ to $\mathcal{H}$. The MMD value between P and Q is calculated as:

$$\begin{aligned} \mathrm{MMD}[\mathcal{F}, P, Q] &= \sup_{\|f\|_{\mathcal{H}} \leq 1} \mathbf{E}_P[f(x)] - \mathbf{E}_Q[f(y)] \\ &= \sup_{\|f\|_{\mathcal{H}} \leq 1} \mathbf{E}_p\left[\langle \phi(x), f\rangle_{\mathcal{H}}\right] - \mathbf{E}_q\left[\langle \phi(y), f\rangle_{\mathcal{H}}\right] \\ &= \sup_{\|f\|_{\mathcal{H}} \leq 1} \langle \mu_P - \mu_Q, f\rangle_{\mathcal{H}} = \|\mu_P - \mu_Q\|_{\mathcal{H}} \end{aligned} \tag{9}$$

where $\mathbf{E}_P[f(x)]$ ($\mathbf{E}_Q[f(y)]$) is the expectation of $f(x)$ ($f(y)$) with $x \in P$ ($y \in Q$). $\mu_P = \mathbf{E}_P[\phi(x)]$ and $\mu_Q = \mathbf{E}_Q[\phi(y)]$ represent the expected values after P and Q are mapped to the feature space, respectively. Eq. (9) squared leads to:

$$\begin{aligned} \mathrm{MMD}^2[\mathcal{F}, P, Q] =& \langle \mu_P - \mu_Q, \mu_P - \mu_Q\rangle_{\mathcal{H}} \\ =& \langle \mu_P, \mu_P\rangle_{\mathcal{H}} + \langle \mu_Q, \mu_Q\rangle_{\mathcal{H}} - 2\langle \mu_P, \mu_Q\rangle_{\mathcal{H}} \\ =& \mathbf{E}_P\langle \phi(x), \phi(x')\rangle_{\mathcal{H}} + \mathbf{E}_Q\langle \phi(y), \phi(y')\rangle_{\mathcal{H}} \\ & - 2\mathbf{E}_{P,Q}\langle \phi(x), \phi(y)\rangle_{\mathcal{H}} \end{aligned} \tag{10}$$

where x' is a random variable independent of x with distribution P and y' is a random variable independent of y with distribution Q.

Since the inner product of the mapping of two features is defined as the kernel, i.e.,

$$k(x, x') = \langle \phi(x), \phi(x')\rangle_{\mathcal{H}}, \tag{11}$$

we can further simplify Eq. (10) as follows

$$\begin{aligned} \mathrm{MMD}^2[\mathcal{F}, P, Q] =& \mathbf{E}_{x,x'}\left[k(x, x')\right] \\ & - 2\mathbf{E}_{x,y}[k(x, y)] + \mathbf{E}_{y,y'}\left[k(y, y')\right], \end{aligned} \tag{12}$$

which can be unbiasedly estimated through

$$\begin{aligned} \mathrm{MMD}^2[\mathcal{F}, X, Y] &= \frac{1}{u(u-1)} \sum_{i \neq j}^{u} k(x_i, x_j) \\ + \frac{1}{v(v-1)} \sum_{i \neq j}^{v} k(y_i, y_j) &- \frac{2}{uv} \sum_{i,j=1}^{u,v} k(x_i, y_j). \end{aligned} \tag{13}$$

At this point, the kernel function $k(x, x')$ can be used to calculate the MMD value of the two distributions. Since Gaussian kernel function can map data from original space to infinite dimension space, it can reflect the information of the original space more accurately regardless of the high dimension of the decision space. In this paper, the Gaussian kernel function

$$k(x, x') = \exp\left(-\|x - x'\|^2 / \left(2\sigma^2\right)\right) \tag{14}$$

where σ is the width parameter of Gaussian kernel function, is applied to calculate the MMD value.

To proceed knowledge transfer in the following LEKT strategy, for each task, the MST strategy selects the l tasks of the smallest MMD values with the task. Fig. 3 illustrates the spatial distance between two distributions measured with MMD. The lower left part depicts the distributions of the decision variables of two task populations in the decision space, and the upper right part shows the corresponding mapping image and the MMD on the unit ball of the RKHS. It can be seen that MMD can better measure the distance of high-dimensional decision variables. Since the mapping of MMD occurs before the knowledge transfer, the MMD distance can measure the knowledge similarity and the MTS strategy based on MMD can choose more suitable tasks for knowledge transfer to reduce negative transfer.

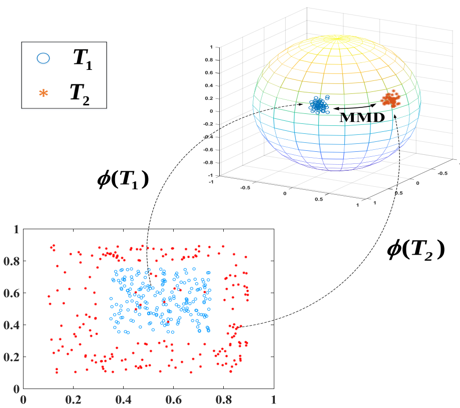


Fig. 3. The illustration of MMD in RKHS.

D. Local ED-based Knowledge Transfer Strategy

Different from the traditional EAs, estimation of distribution algorithm (EDA) uses probability model sampling to generate offspring instead of crossover and mutation operators. Since the probability model can capture the distribution of population variables from a global perspective, it is conducive to accelerating population convergence [43]. However, in the late stage of EDA, the population diversity could be poor, which tends to cause premature [44]. To make up for the defects of EDA, the decision space can be divided into multiple local probability models. At each time, only one local probability model is used to generate the offspring, which can reasonably balance the convergence and diversity of the population.

Based on the above considerations, this paper proposes a local ED-based knowledge transfer (LEKT) strategy. First of all, the related task groups are merged to fuse the knowledge from different tasks. Then, the merged population is clustered in the decision space, where each cluster may contain many individuals from different tasks but close to each other. Finally, the idea of EDA is used to build a probability model for the subpopulation in each cluster, and the offspring are generated by sampling from the related probability model of the parents. Through the local EDA model, the search scope and search preference of a highly-similar task selected by MTS strategy can be estimated and transferred to the target task to promote the evolution.

Fig. 4 shows the basic idea of the LEKT strategy. Fig. 4 (a) is a schematic diagram of dividing the union set of four tasks into three clusters in two-dimensional decision space. The probability models constructed for the corresponding subpopulations are then constructed in Fig. 4 (b). The distribution curves in each cluster are established in different dimensions according to the decision variables of the corresponding subpopulation. Fig. 4 (c) shows a case of generating new

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2021.3101697, IEEE Transactions on Evolutionary Computation

individuals by random sampling under a probability model obtained in Fig. 4 (b). In Fig. 4 (c), x_1^1 is the first dimension decision variable of the new individual sampled from the Gaussian probability model on the first dimension, and x_2^1 is the second dimension decision variable from the second dimensional Gaussian probability model.

The pseudo code of LEKT strategy is shown in **Algorithm 2**. Given a target task i and the list of tasks L selected with the previous MST strategy. Firstly, the corresponding subpopulations of the tasks in L and task i are merged into a set $\mathbf{Q_i}$ (line 1). Afterward, the K-means clustering method is used to divide $\mathbf{Q_i}$ into C groups $\{q_i^1, ..., q_i^C\}$ (line 2). Then, probability models are constructed on each dimension within each cluster (line 3-6). In this paper, we choose the simple Gaussian probability model and use Eqs. (15) and (16) to calculate the mean value $\bar{x}_i^j(k)$ and standard deviation $\mathrm{Cov}_i^j(k)$ of the k-th decision variable of cluster j and task i.

$$\bar{x}_i^j(k) = \frac{1}{\left|q_i^j\right|} \sum_{x \in q_i^j} x(k) \tag{15}$$

$$\mathrm{Cov}_i^j(k) = \frac{1}{\left|q_i^j\right| - 1} \sum_{y \in q_i^j} \left(y(k) - \bar{x}_i^j(k)\right) \left(y(k) - \bar{x}_i^j(k)\right)^T \tag{16}$$

Based on the LEKT strategy, new offspring are generated by sampling from the Gaussian probability models according to the task and the cluster of the selected parent (line 14, **Algorithm 1**).

Algorithm 2 Local ED-based Knowledge Transfer

Input: $\boldsymbol{P}$ (population), i (current task), C (number of groups after clustering), $\boldsymbol{L}$ (list of selected tasks for knowledge transfer)

Output: $\bar{\boldsymbol{x}}_{\boldsymbol{i}}$(mean of Gaussian probability model for task i with all clusters), $\mathbf{Cov}_{\boldsymbol{i}}$(variance of Gaussian probability model for task i with all clusters)

1: Select subpopulations of the tasks in $\boldsymbol{L}$ and task i from $\boldsymbol{P}$ to form the new union $\boldsymbol{Q_i}$;
2: Obtain C clusters with $\{q_i^1, ..., q_i^C\}$=K-means($\boldsymbol{Q_i}$,C);
3: **for** $j = 1 \rightarrow C$ **do**
4:     Calculate $\bar{\boldsymbol{x}}_{\boldsymbol{i}}^{\boldsymbol{j}}$=$(\bar{x}_i^j(1), ..., \bar{x}_i^j(D))$ with Eq. (15);
5:     Calculate $\mathbf{Cov}_{\boldsymbol{i}}^{\boldsymbol{j}}$=$(\mathrm{Cov}_i^j(1), ..., \mathrm{Cov}_i^j(D))$ with Eq. (16);
6: **end for**
7: $\bar{\boldsymbol{x}}_{\boldsymbol{i}}$=$(\bar{\boldsymbol{x}}_i^1, ..., \bar{\boldsymbol{x}}_i^C)$;
8: $\mathbf{Cov}_{\boldsymbol{i}}$=$(\mathbf{Cov}_i^1, ..., \mathbf{Cov}_i^C)$;

Compared with the existing knowledge transfer mechanism used in many-task optimization, the LEKT strategy supports any number of tasks to participate in a single knowledge transfer, which is conducive to enriching the source of knowledge transfer and obtaining more information to help with the evolution of each task. At the same time, because the task groups are selected by the MTS strategy, they are highly similar and the populations of different tasks have close spatial distribution. The spatial distribution of individuals in each subpopulation formed after merging and clustering gets closer. The generation mode of offspring based on probability model sampling is beneficial for the rapid convergence of the population. Each generation of offspring can be derived from different local subpopulation, which is advantageous to maintaining the diversity of the whole population. LEKT strategy uses multiple EDAs to complete knowledge transfer, which not only gains the advantage of fast convergence of EDA, but also avoids the defects of premature.

E. Complexity Analysis

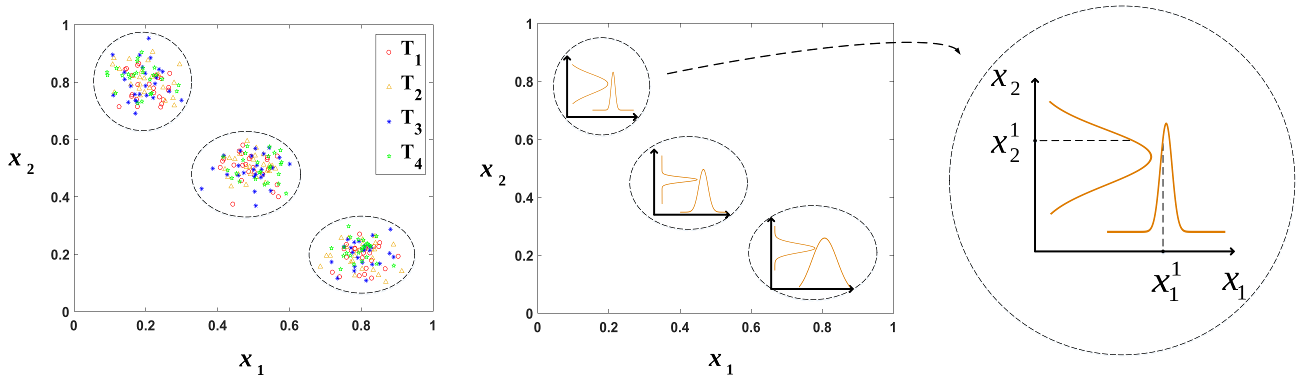
In EMaTO-MKT, the calculation involved in AMP strategy is relatively simple and with a complexity of $O(KN)$, where K denotes the number of tasks and N is the population size. Most of the computation in the MTS strategy is consumed by the MMD value calculation that has a time complexity of $O(N^2D)$ where D is the dimension of the decision variables. The LEKT strategy uses local distribution estimation to generate offspring, which calls for a time complexity of $O(KDC)$ where C is the number of clusters. The inter-task optimization mainly includes the MTS strategy and the LEKT strategy, so the time complexity of inter-task optimization is $O(\max(N^2D, KDC))$. The time complexity of intra-task optimization is $O(ND)$ mainly costed by crossover and mutation operators. The time complexity of the elite selection part for SOO problems is $O(KN)$, while the environmental selection in MOO problems involves time complexity of $O(MN^2)$ and $O(MNlog(N))$ for the non-dominated sorting and the crowding distance computation, respectively [30], where M is the number of objectives. In conclusion, the computational complexity of EMaTO-MKT is $O(\max(KN, N^2D, MN^2))$, i.e., the computational complexity of EMaTO-MKT depends on the number of tasks, the dimension of decision variables, the population size, and the number of objectives.

IV. Experiment and Analysis

In this section, the performance of EMaTO-MKT is tested and compared with other state-of-the-art EMaTO algorithms. The experiment was conducted on two MaTO problem sets proposed in the CEC2019 competition on evolutionary multi-task optimization [45] and the WCCI2020 competition on evolutionary multi-task optimization [46], respectively. Each problem set includes an SOO problem test suite and an MOO problem test suite. In addition, the effectiveness of each strategy in the proposed algorithm and the sensitivity of the parameters are also investigated.

A. Test Problems and Compared Algorithms

The MaTO problem set used in the CEC2019 competition on evolutionary multi-task optimization includes six SOO problems and six MOO problems, which are referred to as CEC19-SOMATP and CEC19-MOMATP, respectively. CEC19-SOMATP involves six common SOO functions [29], namely Rosenbrock, Ackley, Rastrigin, Griewank, Weierstrass and Schwefel. By rotating and translating the optimal position of decision variables, introducing the correlation between variables and increasing the number of optimization tasks,



(a) (b) (c)

Fig. 4. The intuition behind the idea of the LEKT strategy. (a) Clustering the populations into multiple groups. (b) Constructing the probability model on each dimension of each cluster. (c) Sampling data on each dimension under the corresponding of probability model in a selected cluster.

each test problem generates 50 variant SOO functions. The objective optimization functions of CEC19-MOMATP come from the MOO multi-task basic test problem set [47]. Similar to the SOO problems, 50 variant MOO functions are generated for each test MOO problem. The details of CEC19-SOMATP and CEC19-MOMATP test suites can be found in [18].

There are ten problems in the SOO problem test suite of the WCCI2020 competition on evolutionary multi-task optimization, referred to as WCCI20-SOMATP. It involves seven common SOO functions [29], namely Sphere, Rosenbrock, Ackley, Rastrigin, Griewank, Weierstrass and Schwefel. Each problem in WCCI20-SOMATP is composed of 50 tasks. The function combination of each problem in WCCI20-SOMATP is shown in Table I, and the specific information of each function is listed in Table S1 in the Supplementary Materials. The MOO problem test suite of the WCCI2020 competition on evolutionary multi-task optimization contains ten problems, referred to as WCCI20-MOMATP. Each problem is also composed of 50 tasks, among which the objective function involves ZDT problem [48] and DTLZ problem [49]. Table II shows the problem composition of WCCI20-MOMATP. The details of the objective functions are provided in Table S2 of the Supplementary Materials.

Besides the MaTO problems, the competitiveness of EMaTO-MKT on problems of small task size is also demonstrated in Section IV of the Supplementary Materials due to the page limit.

According to [50], tasks with the same objective functions can be referred to as homogeneous tasks, and tasks with different functions can be called heterogeneous tasks. The probability of positive transfer between homogeneous tasks tends to be higher than that of heterogeneous tasks. Heterogeneous tasks are relatively harder to solve.

The compared algorithms contain both the latest EMaTO algorithms and the classic EMTO algorithms. For SOO problems, MFEA [8], EBS [15], MaTEA [17], EEMTA [19] and SaEF-AKT [20] are selected as representative comparison algorithms. Among them, MFEA is a classical single objective EMTO algorithm, EBS is a EMaTO framework based on evolutionary symbiosis theory, MaTEA is an archive-based

TABLE I
THE FUNCTION COMBINATION OF EACH PROBLEM IN WCCI20-SOMATP TEST SUITE.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Sphere	✓			✓						
Rosenbrock		✓		✓		✓		✓	✓	
Rastrigin			✓		✓		✓	✓	✓	✓
Ackley				✓			✓	✓	✓	✓
Griewank					✓	✓		✓	✓	✓
Weierstrass					✓		✓	✓	✓	✓
Schwefel						✓			✓	✓

EMaTO framework, EEMTA is a EMaTO algorithm based on explicit knowledge transmission and SaEF-AKT is a surrogate-assisted EMaTO framework. For MOO problems, NSGA-II [30], MOMFEA [9], EBS and MaTEA are considered in the comparison. Among them, NSGA-II is a classic multi-objective single task optimization algorithm, and MOMFEA is a EMTO algorithm derived from NSGA-II.

B. Parameter Settings and Performance Metric

According to the original parameter settings of the problem set [45], [46], the population size of each EMTO and EMaTO algorithms is set to 5000, and the population size of each task in NSGA-II is set to 100. The peculiar parameters in MaTEA, EEMTA and SaEF-AKT are set in accordance with the original references [17], [19], [20]. The remaining parameters are listed in Table III, and the common parameters refer to the general settings of EMTO algorithm [29], [47].

The objective function value is used to measure the performance of the algorithms in SOO problems. In MOO problems, the inverted generational distance (IGD) [52] is applied to evaluate the performance of the algorithms. The calculation of IGD is given as follows:

$$\mathrm{IGD}(\mathbf{A}, \mathbf{Z}) = \frac{1}{|\mathbf{Z}|} \sum_{j=1}^{|\mathbf{Z}|} \min_{\mathbf{a_i} \in \mathbf{A}} \sqrt{\sum_{k=1}^{M} \left(z_j^k - a_i^k\right)^2} \tag{17}$$

where $\mathbf{A} = \{\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a}_{|\mathbf{A}|}\}$ is the non-dominated solution set in the objective space with $\mathbf{a_i} = (a_i^1, a_i^2, \ldots, a_i^M)$ being

TABLE II
THE COMPONENT PROBLEMS OF WCCI20-MOMATP TEST SUITE.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
objective function	DTLZ	DTLZ	DTLZ	ZDT	ZDT	ZDT	ZDT	ZDT	ZDT	ZDT
convergence function	Sphere	Rastrigin	Griewank	Sphere Rosenbrock Ackley	Rastrigin Griewank Weierstrass	Rosenbrock Griewank Weierstrass	Sphere Ackley Rastrigin	Sphere Rosenbrock Ackley Rastrigin Weierstrass	Rosenbrock Ackley Rastrigin Griewank Weierstrass	Sphere Rosenbrock Ackley Rastrigin Griewank Weierstrass
PF	circle	circle	circle	concave	concave	concave	concave	convex	convex	convex

TABLE III
PARAMETER SETTING FOR EXPERIMENT.

Parameter	value
rmp	0.3
Population size	5,000
The total number of evaluations	5,000,000
SBX [51] η_c for single objective problems	2
SBX [51] η_c for multi-objective problems	20
Polynomial Mutation η_m for single objective problems	5
Polynomial Mutation η_m for multi-objective problems	20
The initial value of *amp*	0.1
The number of selected tasks for knowledge transfer *l*	5
The number of sub-districts for proposed algorithm *C*	10

the M objective function values of solution i, and $\mathbf{Z} = \{\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z}_{|\mathbf{Z}|}\}$ is the reference point set in the objective space with $\mathbf{z_j} = (z_j^1, z_j^2, \ldots, z_j^M)$. IGD can measure the convergence and diversity of the solution set simultaneously. The smaller the IGD value is, the better the performance of the algorithm is.

Another performance metric namely mean standard score (MSS) [29], [47] is applied to compare the overall performance of different algorithms. Suppose there are K tasks $T_1, ..., T_K$ in a problem, and $\mathbf{I} = \{I_1, I_2, \ldots, I_K\}$ represents the best result (in terms of objective function value for SOO problems, and IGD for MOO problems) of an algorithm in each task over a few independent runs. MSS is calculated as

$$\mathrm{MSS}(\mathbf{I}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \frac{1}{K}\sum_{i=1}^{K}\frac{I_i - \mu_i}{\sigma_i} \tag{18}$$

where $\boldsymbol{\mu} = \{\mu_1, \mu_2, \ldots, \mu_K\}$ and $\boldsymbol{\sigma} = \{\sigma_1, \sigma_2, \ldots, \sigma_K\}$ denote the mean value and standard deviation of the results of all algorithms on every task over all runs, respectively. Smaller MSS value indicates better overall performance of the algorithm.

C. Results on Single Objective Many-task Optimization Problems

Both CEC19-SOMATP and WCCI20-SOMATP are composed of SOO problems. The statistics of the best results obtained by MFEA, EBS, MaTEA, EEMTA, SaEF-AKT and EMaTO-MKT over 30 independent runs on CEC19-SOMATP test suite are summarized in Table IV. Particularly, for each algorithm on each test problem, the number of tasks on which the algorithm obtains the best average function values over 30 runs is reported. The best result on each problem is highlighted in bold face. The average convergence traces of all algorithms over 30 independent runs on CEC19-SOMATP are also depicted in Fig. 5.

According to the experiment results, EMaTO-MKT shows superiority to other algorithms by winning on $40+$ out of the total 50 tasks in CEC19-SOMATP1-5. On CEC19-SOMATP6, the performance of EMaTO-MKT is inferior to that of MaTEA. The main reason is that the basin around the local optimal in Schwefel function is wider and deeper, and it is far away from the global optimal. EMaTO-MKT relying on local distribution estimation might get trap in local optima in this function. In contrast, MaTEA uses crossover operator to generate offspring by randomly combining the genes of the parents, leading to widely distributed offspring solutions. More unknown regions can be explored and therefore MaTEA can achieve better overall performance. Since all tasks of each problem in this test suite are homogeneous tasks. The knowledge transfer between tasks can greatly help boost the converge and achieve better performance. Due to the page limite, the detailed results of all algorithms on the CEC19-SOMATP test suit are shown in the Supplementary Materials.

The statistics of the best results obtained by the compared algorithms over 30 independent runs on WCCI20-SOMATP test suite are summarized in Table V. It is shown that EMaTO-MKT consistently achieves better results than the other algorithms in all problems of this test suite. More details of the experimental results are provided in the Supplementary Materials due to the page limit. Wilcoxon rank sum test with 95% confidence level is used to test the significant difference of the algorithm performance. All tasks in WCCI20-SOMATP1-3 are homogeneous tasks while the rest of the problems consist of heterogeneous tasks. In the problems consisting of homogeneous tasks, the algorithms get more similar results on all tasks. In the problems of heterogeneous tasks, EMaTO-MKT selects similar tasks for knowledge transfer according to the evolution process, which could help to better solve the tasks. On WCCI20-SOMATP, EMaTO-MKT manages to achieve competitive overall performance in both homogeneous and heterogeneous tasks thanks to the introduction of AMP, MTS, and LEKT strategies. On WCCI20-SOMATP9, the composition of tasks is complex and diverse, which makes such problem more challenging. On WCCI20-SOMATP9, EMaTO-MKT achieves the best performance in all tasks except those based on Schwefel function, i.e., similar to the observation in CEC19-SOMATP6.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2021.3101697, IEEE Transactions on Evolutionary Computation

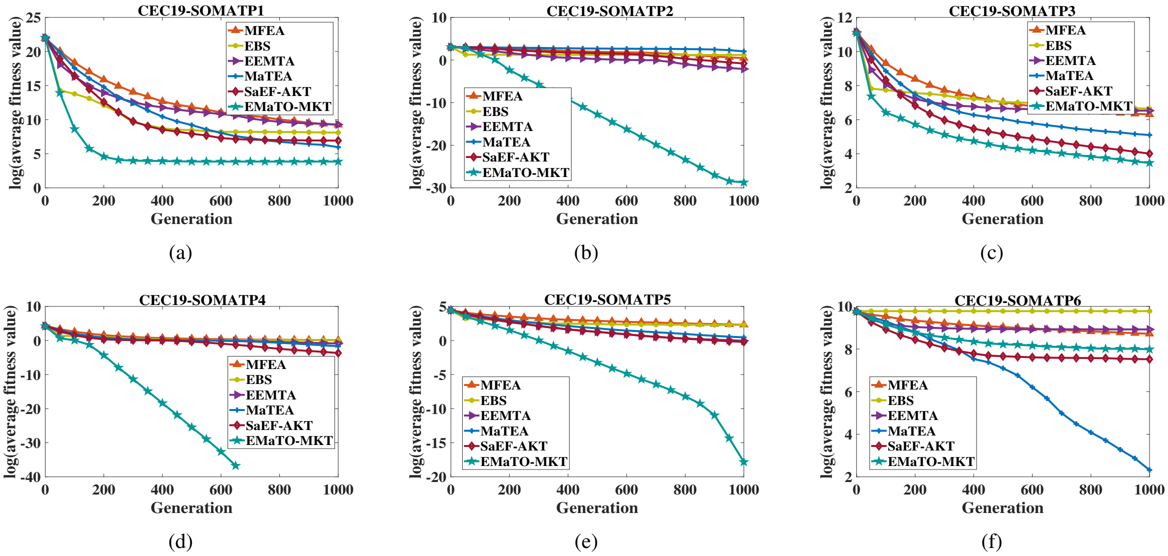


Fig. 5. The log(average fitness value) numerical curves of MFEA, EBS, EEMTA, MaTEA, SaEF-AKT and EMaTO-MKT after running 30 times independently on (a) CEC19-SOMATP1, (b) CEC19-SOMATP2, (c) CEC19-SOMATP3, (d) CEC19-SOMATP4, (e) CEC19-SOMATP5 and (f) CEC19-SOMATP6.

TABLE IV
THE NUMBER OF BEST RESULT FOR ALL TASKS IN EACH PROBLEM OBTAINED BY MFEA, EBS, EEMTA, MATEA, SAEF-AKT AND EMATO-MKT ON CEC19-SOMATP TEST SUITE.

problem	MFEA	EBS	EEMTA	MaTEA	SaEF-AKT	EMaTO-MKT
CEC19-SOMATP1	0	0	3	2	1	**44**
CEC19-SOMATP2	0	0	6	0	0	**44**
CEC19-SOMATP3	0	0	0	0	2	**48**
CEC19-SOMATP4	0	0	0	0	2	**48**
CEC19-SOMATP5	0	0	0	0	0	**50**
CEC19-SOMATP6	0	0	0	**50**	0	0

D. Results on Multi-objective Many-task Optimization Problems

The statistics of the best results obtained by NSGA-II, MOMFEA, EBS, MaTEA and EMaTO-MKT over 30 independent runs in CEC19-MOMATP test suite are reported in Table VI. Wilcoxon rank sum test with 95% confidence level is used to test the significant difference. The average convergence traces of all algorithms over 30 independent running on CEC19-MOMATP test suite are shown in Fig. 6.

According to Table VI, EMaTO-MKT achieves the best results on CEC19-MOMATP1-2 and CEC19-MOMATP5-6, yet its performance on CEC19-MOMATP3-4 is poorer than that of MaTEA. The reason is that the range of decision variables in CEC19-MOMATP3-4 is relatively small, i.e., from the second dimension to the last dimension the ranges are [-5, 5] and [-2, 2] for CEC19-MOMATP3 and CEC19-MOMATP4, respectively. Since the decision space is narrow, the variance of Gaussian distribution in the distribution estimation is small, and the solution would be distributed near the mean value, which leads to the decrease of exploration and population diversity. MaTEA can also play a better role in small search space by using its crossover operator to randomly combine parent genes to generate offspring. Nevertheless, in terms of MSS value, EMaTO-MKT is still competitive to the best algorithm on CEC19-MOMATP3-4. More detailed experimental results on CEC19-MOMATP test suite are provide in the Supplementary Materials.

TABLE V
THE NUMBER OF BEST RESULT FOR ALL TASKS ON EACH PROBLEM OBTAINED BY MFEA, EBS, EEMTA, MATEA, SAEF-AKT AND EMATO-MKT ON WCCI20-SOMATP TEST SUITE.

problem	MFEA	EBS	EEMTA	MaTEA	SaEF-AKT	EMaTO-MKT
WCCI20-SOMATP1	0	0	0	0	0	**50**
WCCI20-SOMATP2	0	0	0	0	0	**50**
WCCI20-SOMATP3	0	0	0	0	0	**50**
WCCI20-SOMATP4	0	0	0	0	0	**50**
WCCI20-SOMATP5	0	0	0	0	0	**50**
WCCI20-SOMATP6	0	0	13	0	2	**35**
WCCI20-SOMATP7	0	0	0	0	0	**50**
WCCI20-SOMATP8	0	0	0	1	0	**49**
WCCI20-SOMATP9	0	0	2	1	3	**44**
WCCI20-SOMATP10	0	0	1	4	4	**41**

TABLE VI
THE NUMBER OF BEST RESULT FOR ALL TASKS IN EACH PROBLEM OBTAINED BY NSGA-II, MOMFEA, EBS, MATEA AND EMATO-MKT ON CEC19-MOMATP TEST SUITE.

problem	NSGA-II	MOMFEA	EBS	MaTEA	EMaTO-MKT
CEC19-MOMATP1	0	0	0	0	**50**
CEC19-MOMATP2	0	0	0	0	**50**
CEC19-MOMATP3	16	15	0	**19**	0
CEC19-MOMATP4	0	0	0	**50**	0
CEC19-MOMATP5	0	0	0	0	**50**
CEC19-MOMATP6	0	0	0	0	**50**

Table VII tabulates the statistics of the best results obtained by the compared algorithms on the WCCI20-MOMATP test suite over 30 independent runs. As can be seen from Table VII, EMaTO-MKT shows superiority over all the problems in this test suite. More detailed experimental results of WCCI20-MOMATP are shown in the Supplementary Materials where

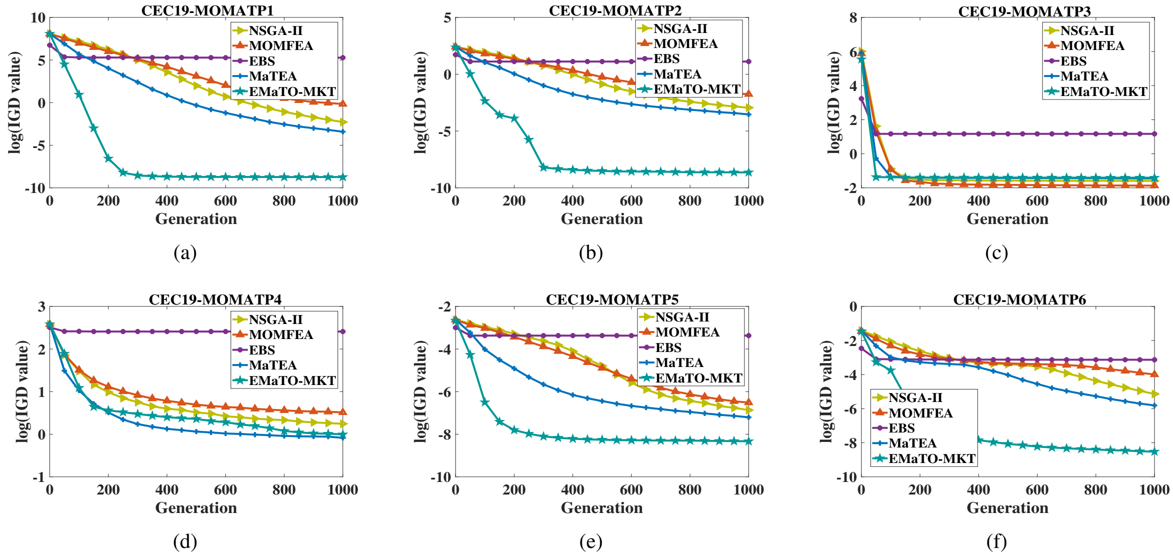


Fig. 6. The log(IGD value) numerical curves of NSGA-II, MOMFEA, EBS, MaTEA and EMaTO-MKT after running 30 times independently on (a) CEC19-MOMATP1, (b) CEC19-MOMATP2, (c) CEC19-MOMATP3, (d) CEC19-MOMATP4, (e) CEC19-MOMATP5 and (f) CEC19-MOMATP6.

the Wilcoxon rank-sum test with a 95% confidence level is also conducted and the best results are highlighted in boldface on each problem. From the results in the experiments, EMaTO-MKT achieves superiority for almost all function types in this test suite.

TABLE VII
THE NUMBER OF BEST RESULT FOR ALL TASKS IN EACH PROBLEM OBTAINED BY NSGA-II, MOMFEA, EBS, MATEA, EMATO-MKT-ST AND EMATO-MKT ON WCCI20-MOMATP TEST SUITE.

problem	NSGA-II	MOMFEA	EBS	MaTEA	EMaTO-MKT
WCCI20-MOMATP1	0	0	0	0	**50**
WCCI20-MOMATP2	0	0	0	0	**50**
WCCI20-MOMATP3	0	0	0	0	**50**
WCCI20-MOMATP4	0	0	0	3	**47**
WCCI20-MOMATP5	0	0	0	0	**50**
WCCI20-MOMATP6	0	0	0	1	**49**
WCCI20-MOMATP7	0	0	0	0	**50**
WCCI20-MOMATP8	1	0	0	1	**48**
WCCI20-MOMATP9	1	0	0	1	**48**
WCCI20-MOMATP10	0	0	0	2	**48**

E. Effects of the strategies

To explore the impact of each strategy in EMaTO-MKT, this section investigates the effect of each strategy independently. Compared with CEC19 test suite, WCCI20 test suite is more complex and can better test the performance of the algorithm. Moreover, since the Ackley function has more local optima and the global optimal value is located in a narrow valley, it is conducive to test the search ability of the algorithm. Therefore, WCCI20-SOMATP9 with Ackley function is selected as the test problem. The counterpart algorithms using only one strategy are denoted as EMaTO-AMP, EMaTO-MTS, and EMaTO-LEKT, respectively. EMaTO-MKT is equipped with all the three strategies. The knowledge transfer in EMaTO-AMP is similar to MFEA, i.e., implemented via SBX operator. In EMaTO-MTS, the knowledge transfer is conducted using SBX operator between the current task and the most similar task in terms of MMD value. EMaTO-LEKT randomly chooses one task to complete the knowledge transfer. Fig. 7 shows the results of the comparison experiments between the four algorithms and the baseline MFEA on WCCI20-SOMATP9. It can be seen that EMaTO-MKT performs the best with the help of the three strategies together. The three algorithms EMaTO-AMP, EMaTO-MTS and EMaTO-LEKT with only one strategy manage to outperform the baseline algorithms.

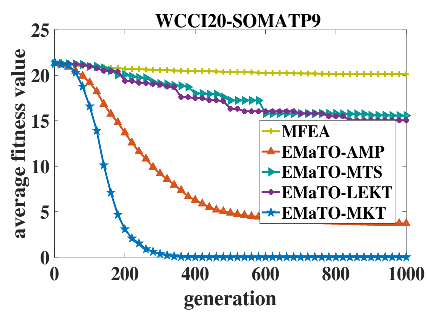


Fig. 7. The result of MFEA, EMaTO-AMP, EMaTO-MTS, EMaTO-LEKT and EMaTO-MKT after running 30 times independently on the selected problem.

Compared with the classical MFEA and MOMFEA, EMaTO-AMP achieves better results as the AMP strategy can adaptively control the probability of knowledge transfer according to the experience in the evolution process and accelerate the population convergence. EMaTO-MTS can reduce the probability of negative transfer by selecting proper tasks with small MMD values in knowledge transfer. EMaTO-LEKT can accelerate the population convergence and meanwhile maintain the population diversity by applying distribution estimations to different types of subpopulations. It is not surprising that EMaTO-MTS and EMaTO-LEKT perform better than MFEA and MOMFEA. EMaTO-MKT combines the advantages of all the three strategies, and therefore it gets the best performance under the comparison.

The effects of the strategies are further investigated by comparing with other methods. Particularly, the AMP strategy is compared with the learning strategy used in MFEA-II [34], which uses a data-driven approach for online rmp estimation (the results reported in Table S3 and Fig. S1 of the Supplementary Materials). The AMP strategy shows better performance than the competitor thanks to its simpleness and effectiveness. MMD in the MTS strategy is pitted against KLD and Jensen-Shannon distance (JSD) in Table S4 and Fig. S2 of the Supplementary Materials. KLD and JSD have achieved good results on MTO problems with small task size, yet MMD is more suitable for MaTO problems thanks to the mapping to RKHS. The local EDA used in the LEKT strategy is also competed with other EDAs to demonstrate the effectiveness of the local EDA method in Table S5 and Fig. S3 of the Supplementary Materials. The knowledge transfer strategy in EMaTO-MKT is compared with the corresponding methods proposed in [16]–[18] (the results are provided in Table S6 and Fig. S4 of the Supplementary Materials) where the proposed knowledge transfer strategy shows great advantages that attribute the success to targeted task selection and effective knowledge transfer.

We also investigate how the similarity of the selected tasks in MTS strategy, i.e., choosing highly or middle similar tasks, affect the performance of the algorithm. In extreme cases, two highly similar tasks have a relatively consistent pace of evolution, where the migration of information becomes similar and the algorithm might not achieve further improvement. However, it is less likely that tasks can reach the same pace at the same time. Therefore, the information from highly similar tasks is still more advanced. Experiments are conducted to compare the effect of selecting high similarity and middle similarity tasks for knowledge transfer. The variant algorithm choosing middle similar tasks, called EMaTO-MKT-mid, selects the tasks ranking 20 after sorting by the MMD value. The compared algorithms are tested on WCCI20-SOMATP test suite as it is more complicated than the CEC19 test suites. Table VIII shows the number of tasks on which the algorithms MFEA, EMaTO-MKT-mid and EMaTO-MKT obtained the best function values over 30 independent runs and Fig. 10 shows the average convergence trace of the compared algorithms over 30 independent runs on 7 single objective functions in WCCI20-SOMATP. The experimental results show that choosing medium similar tasks is helpful in some specific problems, yet overall choosing highly similar tasks is the better option.

A simple version of generating offspring for each task without selecting task (only generate individuals within the current task) is also tested to justify the effectiveness of performing task selection in the algorithm [53]. The variant algorithm without using MMD-based task selection and generating individuals based on the current task alone is denoted as EMaTO-MKT-single-task. Table IX tabulates the number of tasks on which the algorithms MFEA, EMaTO-MKT-single-task, and EMaTO-MKT obtained the best results over 30 independent runs on WCCI20-SOMATP test suite. More details of the results are shown in Section VI of the Supplementary Materials. The experimental results suggest the superiority of EMaTO-MKT with task selection in terms of both convergence speed and solution quality.

TABLE VIII
THE NUMBER OF BEST RESULT FOR ALL TASKS IN EACH PROBLEM OBTAINED BY MFEA, EMATO-MKT-MID AND EMATO-MKT ON WCCI20-SOMATP TEST SUITE.

problem	MFEA	EMaTO-MKT-mid	EMaTO-MKT
WCCI20-SOMATP1	0	0	**50**
WCCI20-SOMATP2	0	17	**33**
WCCI20-SOMATP3	0	2	**48**
WCCI20-SOMATP4	0	9	**41**
WCCI20-SOMATP5	0	5	**45**
WCCI20-SOMATP6	0	14	**36**
WCCI20-SOMATP7	0	5	**45**
WCCI20-SOMATP8	0	5	**45**
WCCI20-SOMATP9	0	9	**41**
WCCI20-SOMATP10	0	8	**42**

TABLE IX
THE NUMBER OF BEST RESULT FOR ALL TASKS IN EACH PROBLEM OBTAINED BY MFEA, EMATO-MKT-SINGLE-TASK AND EMATO-MKT ON WCCI20-SOMATP TEST SUITE.

problem	MFEA	EMaTO-MKT-single-task	EMaTO-MKT
WCCI20-SOMATP1	0	0	**50**
WCCI20-SOMATP2	0	7	**43**
WCCI20-SOMATP3	0	0	**50**
WCCI20-SOMATP4	0	5	**45**
WCCI20-SOMATP5	0	0	**50**
WCCI20-SOMATP6	1	15	**34**
WCCI20-SOMATP7	0	0	**50**
WCCI20-SOMATP8	0	5	**45**
WCCI20-SOMATP9	0	7	**43**
WCCI20-SOMATP10	0	10	**40**

We also verify the performance of the proposed algorithm in improving population diversity. The SPREAD [54] indicator is used to evaluate the distribution diversity of the solutions. A smaller SPREAD value indicates a better diversity of solutions. The details of SPREAD can be found in [54]. The experimental results on WCCI20-MOMATP test suite in terms of SPREAD values are provided in Table S11 and Fig. S5 of the Supplementary Materials. In most problems, EMaTO-MKT can maintain better population diversity than the compared algorithms.

F. Sensitivity Analysis of the Parameters

This section discusses the influence of two key parameters of EMaTO-MKT, namely the number of tasks l in the MTS strategy and the number of clusters C in the LEKT strategy. The problems containing Rastrigin function in WCCI20 test suite, including WCCI20-SOMATP3, WCCI20-SOMATP5, and WCCI20-SOMATP7-10, are selected for the investigation. Compared with CEC19 test suite, WCCI20 test suite is more complex and the Rastrigin function is featured by more local optima. The selected problems are suitable for challenging the convergence ability of the algorithm.

The number of selected tasks l in knowledge transfer is set to be 1,5,10,15,20,30,40, and 50. On the selected test problems, the results of EMaTO-MKT using different l values are shown in Fig. 8. It is shown that the algorithm achieves the best results when $l = 5$. As l increases, the performance of the algorithm degrades. The reason is that choosing too many tasks tend to causes more negative transfer, which reduces the benefits of knowledge transfer. With $l = 1$, the LEKT strategy chooses only one task with the minimum MMD value. In this configuration, the algorithm obtains promising performance but not as good as that of $l = 5$. This is because the negative transfer is relatively less with $l = 1$, but meanwhile, it does not give full play to the advantage of many-task optimization to obtain knowledge from more tasks.

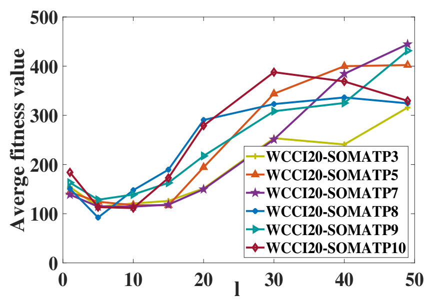


Fig. 8. The result of EMaTO-MKT with different l values after running 30 times independently on selected problems.

To investigate the effect of the cluster number C, it is set to 2,5,8,10,15,20,and 30, respectively, in the comparison study. On the selected test problems, the results of EMaTO-MKT with different C values are summarized in Fig. 9. It can be observed that as C increases form 2 to 10, the performance of the algorithm improves and remains relatively stable when $C > 10$. When C is small, the number of subpopulation is insufficient to maintain good diversity of the whole population. In contrast, if C is too large, the algorithm could waste computational resource in searching unpromising areas. The best configuration of C is observed to be around 10.

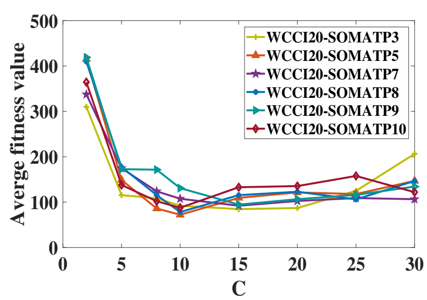


Fig. 9. The result of EMaTO-MKT with different C values after running 30 times independently on selected problems.

V. CONCLUSION AND FUTURE WORK

In the light of the shortcomings of the existing EMaTO algorithms, this paper proposes a new EMaTO algorithm based on multi-source knowledge transfer namely EMaTO-MKT. The multi-source knowledge transfer is featured by three strategies, i.e., AMP, MTS and LEKT. The AMP strategy adaptively adjusts the frequency of knowledge transfer by using the convergence trend of the population in the evolution process, which can reduce the unnecessary random knowledge transfer and improve computational resource allocation. The MTS strategy can reduce the probability of negative transfer by selecting highly similar tasks in terms of MMD values for knowledge transfer. The LEKT strategy is based on local estimation of distribution to transfer knowledge among multiple tasks, which can accelerate population convergence and maintain population diversity. On both single-objective and multi-objective MaTO test suites, EMaTO-MKT shows good competitiveness compared with other state-of-art EMaTO al-

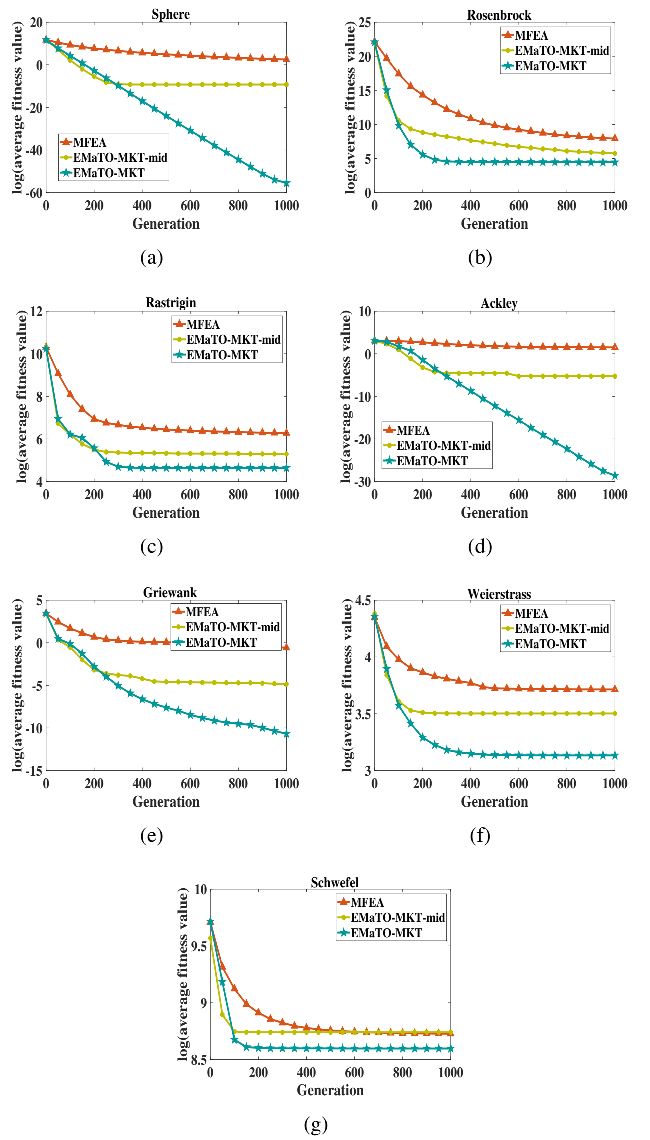


Fig. 10. The log(average fitness) numerical curve of MFEA, EMaTO-MKT-mid and EMaTO-MKT running 30 times independently on 7 single functions in WCCI20-SOMATP test suite. Convergence traces in (a) Sphere, (b) Rosenbrock, (c) Rastrigin, (d) Ackley, (e) Griewank, (f) Weierstrass and (g) Schwefel.

gorithms.

The knowledge transfer mechanism formed by the combination of the three strategies has achieved promising results, yet there are a few open questions left for potential future work. For example, the parameter sensitivity analysis on the benchmark test problems shows that five tasks involved in knowledge transfer can achieve the best performance, but this might not be the best choice for practical problems. An adaptive way to decide the number of tasks participating in knowledge transfer is more desirable. In selecting tasks for knowledge transfer, MMD measures the difference between populations for the recent generations, but the recent evolutionary trend could be biased, thus affecting the accuracy of task selection. Involving more historical experience of the population can more accurately reflect the evolutionary trend of the current tasks, and it deserves further investigation. In addition, handling more tasks inevitably costs more time and space consumptions. How to improve the efficiency of the algorithms in solving MaTO problems remains an critical issue. The source code of EMaTO-MKT written in MATLAB is available at https://github.com/CIA-SZU/XXJ.

REFERENCES

[1] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 440 – 462, 2017.

[2] M. Pourhassan and F. Neumann, "Theoretical analysis of local search and simple evolutionary algorithms for the generalized travelling salesperson problem," *Evol. Comput.*, vol. 27, no. 3, pp. 452 – 558, 2018.

[3] R. Tanabe and H. Ishibuchi, "A review of evolutionary multimodal multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 193 – 200, 2020.

[4] F. Sarro, F. Ferrucci, M. Harman, A. Manna, and J. Ren, "Adaptive multiobjective evolutionary algorithms for overtime planning in software projects," *IEEE Trans. Software. Eng*, vol. 43, no. 10, pp. 898 – 917, 2017.

[5] J. Wang, T. Weng, and Q. Zhang, "A two-stage multiobjective evolutionary algorithm for multiobjective multidepot vehicle routing problem with time windows," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2467 – 2478, Jul. 2019.

[6] J. P. Hanley, D. M. Rizzo, J. S. Buzas, and M. J. Eppstein, "A tandem evolutionary algorithm for identifying causal rules from complex data," *Evol. Comput.*, vol. 28, no. 1, pp. 87 – 114, 2020.

[7] Y.-J. Zheng, Y.-C. Du, H.-F. Ling, W.-G. Sheng, and S.-Y. Chen, "Evolutionary collaborative human-uav search for escaped criminals," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 217 – 231, Apr. 2020.

[8] A. Gupta, Y.-S. Ong, and L. Feng, "Multifactorial evolution: Toward evolutionary multitasking." *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 343–357, June 2016.

[9] A. Gupta, Y.-S. Ong, L. Feng, and K. C. Tan, "Multiobjective multifactorial optimization in evolutionary multitasking," *IEEE Trans. Cybern.*, vol. 47, no. 7, pp. 1652–1665, July 2017.

[10] L. Feng, Y. Huang, L. Zhou, J. Zhong, A. Gupta, K. Tang, and K. C. Tan, "Explicit evolutionary multitasking for combinatorial optimization: A case study on capacitated vehicle routing problem," *IEEE Trans. Cybern.*, pp. 1–14, 2020.

[11] D. Wu and X. Tan, "Multitasking genetic algorithm (mtga) for fuzzy system optimization," *IEEE Trans. Fuzzy. Syst*, vol. 28, no. 6, pp. 1050 – 1061, Jun. 2020.

[12] P. D. Thanh, H. T. T. Binh, and T. B. Trung, "An efficient strategy for using multifactorial optimization to solve the clustered shortest path tree problem," *Appl. Intell*, vol. 50, no. 4, pp. 1233 – 1258, Apr. 2020.

[13] Z. Wang and X. Wang, "Multiobjective multifactorial operation optimization for continuous annealing production process," *Ind. Eng. Chem. Res.*, vol. 58, no. 41, pp. 19 166 – 19 178, Oct. 2019.

[14] J. Zhong, L. Feng, W. Cai, and Y.-S. Ong, "Multifactorial genetic programming for symbolic regression problems," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 50, no. 11, pp. 4492–4505, Jul. 2018.

[15] R.-T. Liaw and C.-K. Ting, "Evolutionary many-tasking based on biocoenosis through symbiosis: A framework and benchmark problems," in *Proc. IEEE Congr. Evol. Comput.* IEEE, 2017, pp. 2266–2273.

[16] J. Tang, Y. Chen, Z. Deng, Y. Xiang, and C. P. Joy, "A group-based approach to improve multifactorial evolutionary algorithm," in *Int Joint Conf Art Int.*, 2018, pp. 3870–3876.

[17] Y. Chen, J. Zhong, L. Feng, and J. Zhang, "An adaptive archive-based evolutionary framework for many-task optimization," *IEEE Trans. Emerging Topics Comp. Intell.*, vol. 4, no. 3, pp. 369 – 384, Jun. 2020.

[18] Z. Liang, H. Dong, C. Liu, W. Liang, and Z. Zhu, "Evolutionary multitasking for multiobjective optimization with subspace alignment and adaptive differential evolution," *IEEE Trans. Cybern.*, pp. 1–14, 2020.

[19] Q. Shang, L. Zhang, L. Feng, Y. Hou, J. Zhong, A. Gupta, K. Tan, and H.-L. Liu, "A preliminary study of adaptive task selection in explicit evolutionary many-tasking," in *IEEE Congr. Evol.. Comput., CEC - Proc.*, Wellington, New zealand, Jun 2019, pp. 2153 – 2159.

[20] S. Huang, J. Zhong, and W. Yu, "Surrogate-assisted evolutionary framework with adaptive knowledge transfer for multi-task optimization," *IEEE Trans. Emerg. Top. Comput.*, 2019.

[21] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Scholkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, pp. 723 – 773, Mar 2012.

[22] K. Weiss, T. M. Khoshgoftaar, and D. D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, p. 9, May 2016.

[23] P. Larranga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2001.

[24] J. Lin, H. Liu, K. C. Tan, and F. Gu, "An effective knowledge transfer approach for multiobjective multitasking optimization," *IEEE Trans. Cybern.*, vol. 24, no. 5, pp. 1–11, 2020.

[25] K. K. Bali, A. Gupta, Y. Ong, and P. S. Tan, "Cognizant multitasking in multiobjective multifactorial evolution: Mo-mfea-ii," *IEEE Trans. Cybern.*, pp. 1–13, 2020, early access.

[26] X. Hao, R. Qu, and J. Liu, "A unified framework of graph-based evolutionary multitasking hyper-heuristic," *IEEE Trans. Evol. Comput.*, 2020.

[27] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41 – 75, Jul. 1997.

[28] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data. Eng.*, vol. 22, no. 10, pp. 1345 – 1359, 2010.

[29] B. Da, Y.-S. Ong, L. Feng, A. Qin, A. Gupta, Z. Zhu, C.-K. Ting, K. Tang, and X. Yao, "Evolutionary multitasking for single-objective continuous optimization: Benchmark problems, performance metric, and baseline results," *arXiv preprint arXiv:1706.03470*, June 2017.

[30] D. Kalyanmoy, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[31] J. Ding, C. Yang, Y. Jin, and T. Chai, "Generalized multitasking for evolutionary optimization of expensive problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 44–58, Feb 2019.

[32] L. Feng, L. Zhou, J. Zhong, A. Gupta, Y.-S. Ong, K.-C. Tan, and A. Qin, "Evolutionary multitasking via explicit autoencoding," *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3457 – 3470, Sep. 2019.

[33] X. Zheng, A. Qin, M. Gong, and D. Zhou, "Self-regulated evolutionary multitask optimization," *IEEE Trans. Evol. Comput*, vol. 24, no. 1, pp. 16 – 28, Feb. 2020.

[34] K. K. Bali, Y.-S. Ong, A. Gupta, and P. S. Tan, "Multifactorial evolutionary algorithm with online transfer parameter estimation: Mfea-ii," *IEEE Trans. Evol. Comput*, vol. 24, no. 1, pp. 69 – 83, Feb. 2020.

[35] L. Zhou, L. Feng, K. C. Tan, J. Zhong, Z. Zhu, K. Liu, and C. Chen, "Toward adaptive knowledge transfer in multifactorial evolutionary computation," *IEEE Trans. Cybern.*, pp. 1–14, 2020.

[36] R. Chandra and a. S. Cripps, "Bayesian multi-task learning for dynamic time series prediction," in *Proc Int Jt Conf Neural Networks*, vol. 2018-July, 2018.

[37] Z. Tang, M. Gong, and M. Zhang, "Evolutionary multi-task learning for modular extremal learning machine," in *IEEE Congr. Evol. Comput., CEC - Proc.*, 2017, pp. 474 – 479.

[38] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2005.

[39] C. Jin, P.-W. Tsai, and A. Qin, "A study on knowledge reuse strategies in multitasking differential evolution," in *IEEE Congr. Evol.. Comput., CEC - Proc.*, Wellington, New zealand, Jun. 2019, pp. 1564 – 1571.

[40] J. Gao, "A local domain adaptation feature extraction method," in *Proc. - Int. Conf. Fuzzy Syst. Knowl. Discov., FSKD*, Shenyang, China, 2013, pp. 526 – 530.

[41] Y. Liu, Y. Wang, and A. Sowmya, "Batch mode active learning for object detection based on maximum mean discrepancy," in *Int. Conf. Digit. Image Comput.: Tech. Appl., DICTA*, Adelaide, SA, Australia, 2015, pp. Australian Government Department of Defence, Defence Science and Technology Group; Bae System; Canon Information Systems Research Australia Pty Ltd (CISRA); Flinders University –.

[42] W.-W. Lin, M.-W. Mak, and J.-T. Chien, "Multisource i-vectors domain adaptation using maximum mean discrepancy based autoencoders," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 26, no. 12, pp. 2412 – 2422, Dec. 2018.

[43] B. Doerr and M. S. Krejca, "Significance-based estimation-of-distribution algorithms," in *GECCO - Proc. Genet. Evol. Comput. Conf.*, Kyoto, Japan, Jul. 2018, pp. 1483 – 1490.

[44] A. Zhou, J. Sun, and Q. Zhang, "An estimation of distribution algorithm with cheap and expensive local search methods," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 807 – 822, Dec. 2015.

[45] L. Feng, K. Qin, A. Gupta, Y. Yuan, Y.-S. Ong, and X. Chi, "IEEE CEC 2019 Competition on Evolutionary Multi-task Optimization," http://cec2019.org/programs/competitions.html#cec-02 and http://www.bdsc.site/websites/MTO_competiton_2019/MTO_Competition_CEC_2019.html, 2019.

[46] ——, "IEEE WCCI 2020 Competition on Evolutionary Multi-task Optimization," https://wcci2020.org/programs/competitions.html#cec-c7 and http://www.bdsc.site/websites/MTO_competition_2020/MTO_Competition_WCCI_2020.html, 2020.

[47] Y. Yuan, Y.-S. Ong, L. Feng, A. Qin, A. Gupta, B. Da, Q. Zhang, K. C. Tan, Y. Jin, and H. Ishibuchi, "Evolutionary multitasking for multiobjective continuous optimization: Benchmark problems, performance metrics and baseline results," *arXiv preprint arXiv:1706.02766*, June 2017.

[48] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, Feb 2000.

[49] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proc. Congr. Evol. Comput.*, vol. 1. IEEE, 2002, pp. 825–830.

[50] F. Zhang, Y. Mei, S. Nguyen, K. C. Tan, and M. Zhang, "Multitask genetic programming-based generative hyperheuristics: A case study in dynamic scheduling," *IEEE Trans. Cybern.*, pp. 1–14, 2021.

[51] R. B. Agrawal, K. Deb, and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Compl. Syst.*, vol. 9, no. 2, pp. 115–148, June 1995.

[52] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, April 2003.

[53] F. Zhang, Y. Mei, S. Nguyen, M. Zhang, and K. C. Tan, "Surrogate-assisted evolutionary multitask genetic programming for dynamic flexible job shop scheduling," *IEEE Trans. Evol. Comput.*, pp. 1–1, 2021.

[54] Y.-N. Wang, L.-H. Wu, and X.-F. Yuan, "Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure," *Soft Comput.*, vol. 14, no. 3, pp. 193–209, 2010.

Zhengping Liang received the B.S. degree in computer science and technology from Hunan Normal University, China, in 2001, and the Ph.D degree in computer science and technology from Wuhan University, China, in 2006. He is currently an associate professor with the college of computer science and software engineering, Shenzhen University, China.

His main research interest include computational intelligence, multi-objective optimization and big data analysis.

Xiuju Xu received the B.S. degree from Guangzhou Medical University, Guangzhou, China, in 2018, and the M.S. degree from Shenzhen University, Shenzhen, China, in 2021. She is currenly pursuing for the Ph.D degree from Sun Yat-sen University, Guangzhou, China.

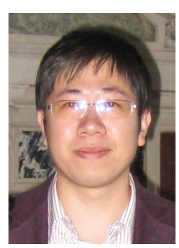
Her current research interests include transfer learning, multi-task optimization and adaptive cyber defense.

Ling Liu received the B.S. degree in electronic engineering from Nanjing University, Nanjing, China, in 2008, the M.S. degree in electronic engineering from Peking University, Beijing, China, in 2012, and the Ph.D. degree in communication and signal processing from the Electrical and Electronic Engineering Department, Imperial College London, London, U.K., in 2016. He was a Research Assistant with the CSP Group, Imperial College London. From 2017 to 2019, he served as a Senior Engineer with the CTLab of Huawei Technologies, Shenzhen, China. He is currently an Assistant Professor with the Department of Computer Science and Software Engineering, Shenzhen University, Guangdong, China.

His research interests include coding theory, information theory, and machine learning.

Yaofeng Tu received the M.S.degree in aerospace manufacturing engineering from Nanjing University of Aeronautics and Astronautics, China, in 1997. He is currently a researcher with the Central R&D Institute, ZTE Corporation, China.

His main research interest include big data, distributed systems and machine learning.



Zexuan Zhu received the B.S. degree in computer science and technology from Fudan University, Shanghai, China, in 2003, and the Ph.D. degree in computer engineering from Nanyang Technological University, Singapore, in 2008. He is currently a Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.

His research interests include computational intelligence, machine learning, and bioinformatics. Prof. Zhu is an Associate Editor of IEEE Transactions on Evolutionary Computation and IEEE Transactions on Emerging Topics in Computational Intelligence. He is also the Chair of the IEEE CIS Emergent Technologies Task Force on Memetic Computing.