This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

An Effective Knowledge Transfer Approach for Multiobjective Multitasking Optimization

Jiabin Lin, Hai-Lin Liu, *Senior Member, IEEE*, Kay Chen Tan, *Fellow, IEEE*, and Fangqing Gu

***Abstract*—Multiobjective multitasking optimization (MTO), which is an emerging research topic in the field of evolutionary computation, was recently proposed. MTO aims to solve related multiobjective optimization problems at the same time via evolutionary algorithms. The key to MTO is the knowledge transfer based on sharing solutions across tasks. Notably, positive knowledge transfer has been shown to facilitate superior performance characteristics. However, how to find more valuable transferred solutions for the positive transfer has been scarcely explored. Keeping this in mind, we propose a new algorithm to solve MTO problems. In this article, if a transferred solution is nondominated in its target task, the transfer is positive transfer. Furthermore, neighbors of this positive-transfer solution will be selected as the transferred solutions in the next generation, since they are more likely to achieve the positive transfer. Numerical studies have been conducted on benchmark problems of MTO to verify the effectiveness of the proposed approach. Experimental results indicate that our proposed framework achieves competitive results compared with the state-of-the-art MTO frameworks.**

***Index Terms*—Evolutionary algorithm, knowledge transfer, multiobjective optimization, multitasking learning.**

I. Introduction

EVOLUTIONARY multiobjective optimization has been a crucial technique to solve a wide range of problems in the fields of science and engineering [1]–[8]. Recently, with the development of evolutionary multitasking optimization [9], multiobjective multitasking optimization (MTO) is gradually stimulating the attention of researchers.

An MTO problem consists of multiple multiobjective optimization problems [10] which have relatedness in their Pareto-optimal solutions or fitness landscapes [11].

Manuscript received August 11, 2019; revised November 23, 2019 and January 16, 2020; accepted January 21, 2020. This work was supported in part by the Natural Science Foundation of Guangdong Province under Grant 2020A151501989, in part by the Projects of Science and Technology of Guangzhou under Grant 201804010352, and in part by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Grant CityU11209219 and Grant CityU11202418. This article was recommended by Associate Editor Y. S. Ong. *(Corresponding author: Hai-Lin Liu.)*

Jiabin Lin, Hai-Lin Liu, and Fangqing Gu are with the Guangdong University of Technology, Guangzhou 510006, China (e-mail: jiabinlin00@gmail.com; hlliu@gdut.edu.cn; fqgu@gdut.edu.cn).

Kay Chen Tan is with the City University of Hong Kong, Hong Kong (e-mail: kaytan@cityu.edu.hk).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2020.2969025

Without loss of generality, an MTO problem can be formulated as follows:

$$\text{Minimize}: \begin{cases} F_1(\mathbf{x}_1) = \left(f_1^1(\mathbf{x}_1), \ldots, f_1^{m_1}(\mathbf{x}_1)\right) \\ F_2(\mathbf{x}_2) = \left(f_2^1(\mathbf{x}_2), \ldots, f_2^{m_2}(\mathbf{x}_2)\right) \\ \ldots, \\ F_k(\mathbf{x}_k) = \left(f_k^1(\mathbf{x}_k), \ldots, f_k^{m_k}(\mathbf{x}_k)\right) \end{cases}$$

$$\text{subject to } \mathbf{x}_i \in \prod_{s=1}^{n_i}\left[a_i^s, b_i^s\right], \quad i = 1, 2, \ldots, k \tag{1}$$

where F_k denotes the kth multiobjective optimization task. $\prod_{s=1}^{n_k}[a_k^s, b_k^s]$ denotes the decision space of the kth task. n_k denotes the dimensionality of the decision space of the kth task. $f_k : \prod_{s=1}^{n_k}[a_k^s, b_k^s] \rightarrow R^{m_k}$ consists of m_k real-valued continuous objective functions $f_k^1, \ldots, f_k^{m_k}$. R^{m_k} denotes the objective space of the kth task.

The pioneering work of Gupta *et al.* [12] introduced an evolutionary multitasking paradigm (MFEA) which uses a single population to solve several related optimization problems simultaneously. In MFEA, each optimization task is referred as a factor influencing the evolution of a population. As one of the most representative evolutionary multitasking optimization algorithms, MFEA has been successfully developed to solving complex real-world problems. Presently, there are mainly five research directions in the field of evolutionary multitasking optimization, which can be described as follows. First, how to reduce negative transfer [13]–[15] by adjusting the intensity of knowledge transfer has been studied. To solve this problem, an evolutionary multitasking optimization algorithm based on online learning was proposed in [16]. In this method, the correlation between different tasks is utilized to determine how much knowledge to transfer across tasks. Furthermore, Zheng *et al.* [17] also proposed an algorithm to capture and share useful knowledge between tasks. In this article, the intensity of knowledge transfer is adapted according to the degree of task relatedness. Second, how to allocate computational resources to different optimization problems has also been studied. Since different tasks cost different amounts of computational resources, it has been demonstrated that the rational allocation of computational resources is able to improve the performance of evolutionary multitasking. With the aim of dealing with this problem, Gong *et al.* [18] proposed an algorithm in which computational resources are dynamically allocated to different tasks according to their requirements. Moreover, a resource allocation mechanism was proposed in [19] to reallocate fitness evaluations on offsprings of different tasks. Third, how to build mapping among tasks for meaningful knowledge transfer also gradually attracted the attention of researchers. With the aim of achieving positive transfer across tasks, a linearized domain adaptation was proposed in [20]. This approach transforms the search space of a simplex task to the search space similar to its complex constitutive task. Feng *et al.* [21] recently proposed an MTO algorithm (EMEA) with the incorporation of multiple search mechanisms for solving different tasks. In this article, first, two sets of solutions are uniformly sampled from the search space of two different tasks. Then, these solutions are used to build the linear mapping between two tasks by the denoising autoencoders. Fourth, real-world application is also one of the research directions in evolutionary multitasking optimization. Toward applying evolutionary multitasking optimization to deal with NP-hard capacitated vehicle routing problems, a permutation-based unified representation and a split-based decoding operator were proposed in [22] and [23]. Finally, as the number of optimization tasks increases, knowledge transfer among tasks becomes more complex and more time consuming. Based on this, rationally simplifying the connection among tasks is also a crucial issue in evolutionary multitasking. An evolutionary framework of many-tasking optimization was proposed in [24]. In this article, each task is assigned with an assisted task based on the analysis of the correlation between tasks, which significantly reduces the unnecessary connections between tasks and improves the efficiency of knowledge transfer.

MTO is based on the assumption that there exists some useful knowledge in common for solving related tasks. Transferring such knowledge of value among tasks, known as the positive knowledge transfer, has a significant effect on improving the performance of MTO. However, one crucial issue, finding more solutions containing valuable knowledge for achieving the positive transfer, has been given little attention. In MFEA and its variants [16]–[18], [25]–[27], the number of solutions exchanging information among tasks is controlled by the parameter rmp. In each task, each solution will be selected as a transferred solution based on the same probability. However, some of them do not help to optimize the other tasks, thereby leading to the low efficiency of achieving the positive transfer. In EMEA, transferred solutions are selected from the nondominated solutions in each task, while the performance of this method may primarily rely on the high degree of underlying intertask similarities [16]. Consequently, in the absence of any information about intertask relationships, proposing a strategy selecting solutions for achieving the positive transfer, could be a promising avenue to improve the effectiveness of MTO algorithms.

Based on the above facts, this article proposes an effective algorithm (EMT/ET) to solve MTO problems. In this article, if a transferred solution is nondominated in its target task, it achieves positive transfer. Furthermore, in the original search space of this positive-transfer solution, its closest (based on the Euclidean distance) solutions will be selected as the transferred solutions in the new generation. The reason is that these solutions are more likely to achieve positive transfer.

The main contributions of this article can be summarized as follows.

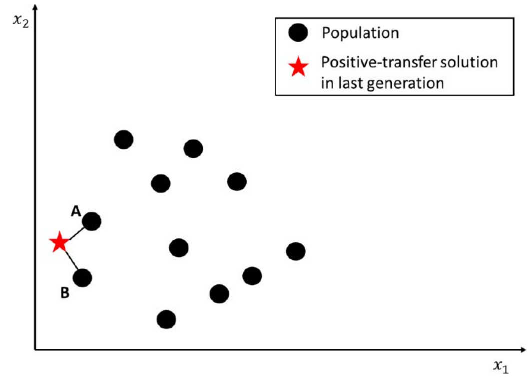

Fig. 1. Illustration of the transferred solutions which are selected by the proposed method.

1) An approach to finding valuable solutions for positive transfer is proposed. In this method, transferred solutions are selected from the neighbors of solutions that achieved the positive transfer, thereby enhancing the convergence characteristic of their target tasks.
2) An optimization framework for MTO is developed. Systematic experiments have demonstrated the effectiveness of our proposed framework.

The remainder of this article is organized as follows. The details of the proposed algorithm are presented in Section II. Section III reports and analyzes experimental results. Finally, the conclusions are drawn in Section IV.

II. Proposed Algorithm

This section starts with the description of our proposed strategy for selecting transferred solutions. Then, the main framework of EMT/ET will be given in the final section.

A. Selection of Transferred Solutions

For the knowledge transfer among tasks, transferred solutions are randomly selected from each task in the first generation. When the generation $g > 1$, transferred solutions will be selected based on the solutions which achieved the positive transfer in generation $g - 1$. In this article, a transferred solution achieves positive transfer if it is nondominated in its target task. Then, in the search space of the original task of this positive-transfer solution, its several closest (based on the Euclidean distance) solutions will be the transferred solutions, since these solutions are more likely to achieve positive transfer.

Fig. 1 illustrates how to select transferred solutions by the proposed method. As can be observed, in Fig. 1, the red star represents a solution that achieved positive transfer in the last generation, and the black dots denote the population in the current generation. Based on our proposed method, the neighbors of the red star (denoted as A and B) will be selected as transferred solutions in the current generation.

For any two optimization tasks, the number of solutions that are transferred from one task to the other task is n, where $n \geq 1$. If none of the transferred solutions in one task achieves positive transfer in generation $g-1$, new transferred solutions will be selected from the nondominated solutions of this task in generation g.

Algorithm 1 Pseudocode of the Proposed Algorithm

Input: $T_1, T_2, \ldots, T_k, \ldots, T_K$: K multiobjective optimization problems.
n: The number of solutions transferred from one task to the other task.
p: The probability that the transferred solutions are changed.
λ: Distrubed factor.

Output: K nondominated solution sets for K optimization problems.

1: **Initialize** K populations, $P_1^t, P_2^t, \ldots, P_k^t, \ldots, P_K^t$, $t = 0$.
2: **Evaluate** $P_1^t, P_2^t, \ldots, P_k^t, \ldots, P_K^t$.
3: **Select** n solutions from task j for task k ($k, j = 1, 2, \ldots, K, k \neq j$), denoted as $S_{j\rightarrow k}$, via the approach presented in Section II-A.
4: **for** each solution of $S_{j\rightarrow k}$ **do**
5: **if** $random(0, 1) < p$ **then**
6: Multiple this solution and λ.
7: **end if**
8: **end for**
$S_{j\rightarrow k}$ is changed to $CS_{j\rightarrow k}$.
9: **for** $k = 1, 2, \ldots, K$ **do**
10: Use P_k^t and $\bigcup_{j=1}^{K}(CS_{j\rightarrow k}), j \neq k$, to generate offsprings C_k^t.
11: **end for**
12: **for** $k = 1, 2, \ldots, K$ **do**
13: Evaluate the result population R_k^t, where $R_k^t = P_k^t \bigcup C_k^t \bigcup (\bigcup_{j=1}^{K}(CS_{j\rightarrow k}))$.
14: **end for**
15: **for** $k = 1, 2, \ldots, K$ **do**
16: Select N fittest solutions from R_k^t into P_k^{t+1}.
17: **end for**
18: **Set** $t = t + 1$.
19: **If** stopping conditions are not satisfied, go to step 3; else, go to step 20.
20: **return** $P_1^t, P_2^t, \ldots, P_k^t, \ldots, P_K^t$.

B. Main Framework of the Proposed Algorithm

The pseudocode of our proposed algorithm is presented in Algorithm 1. In particular, $T_1, T_2, \ldots, T_k, \ldots, T_K$ are K multiobjective optimization problems of an MTO problem. First, the decision spaces of these tasks should be combined into a unified space Y in order to achieve communication among tasks. As stated in [12], the unified space Y is $[0, 1]^D$, where $D = \max_k\{D_k\}$ and D_k is the decision space dimensionality of T_k. Suppose the jth variable (x_k^j) of T_k is box constrained as $[l_k^j, u_k^j]$. Therefore, $x_k^j \in [l_k^j, u_k^j]$, and the encoding formula to normalize is $[(x_k^j - l_k^j)/(u_k^j - l_k^j)]$. Suppose $L_i = (l_k^1, l_k^2, \ldots, l_k^j, \ldots, l_k^{D_k})$ denotes the lower bound of T_k. $U_k = (u_k^1, u_k^2, \ldots, u_k^j, \ldots, u_k^{D_k})$ denotes the upper bound of T_k. The formula of transferring a solution $\mathbf{x}$ from T_k to T_j, denoted as $\mathbf{x}'$, is shown as follows:

$$\mathbf{x}' = \frac{\mathbf{x} - L_k}{U_k - L_k} \cdot \left(U_j - L_j\right) + L_j. \tag{2}$$

When the decision space dimensionalities of these two tasks are different, this formula can be modified and referred to the strategy proposed in [27]. As shown in Algorithm 1, K populations ($P_1^t, P_2^t, \ldots, P_k^t, \ldots, P_K^t$) are used to solve an MTO problem containing K optimization tasks. The size of each population is N. Then, for any two optimization tasks (T_j and T_k), n transferred solutions are selected from one task (T_j) for helping to optimize the other task (T_k), denoted as $S_{j\rightarrow k}$, via the method proposed in Section II-A.

However, the optimal solutions of T_j and T_k may be similar but not completely identical in the unified search space; hence, $S_{j\rightarrow k}$ sometimes may not help to optimize T_k. To further improve the performance of knowledge transfer, $S_{j\rightarrow k}$ is changed with a certain probability. In this article, solutions of $S_{j\rightarrow k}$ are multiplied by a distributed factor λ with a probability p; thus, $S_{j\rightarrow k}$ is changed to $CS_{j\rightarrow k}$. Furthermore, the offspring population of each task is created, denoted as C_k^t. In this step, the parent population P_k^t injected with $CS_{j\rightarrow k}$ is used as the input for the offspring generation to obtain offsprings C_k^t. Finally, based on environmental selection, the N best individuals from the combined parent and offspring population R_k^t are selected into P_k^{t+1}, which allows us to preserve elite individuals of the parent population.

III. Experimental Setup and Discussion of the Results

In this section, we comprehensively evaluate the performance of our proposed algorithm (EMT/ET) in comparison of EMEA, MFEA, SPEA2 [28], and NSGA-II [29] on nine benchmark problems, and make an in-depth analysis of the selection of transferred solutions in EMT/ET to demonstrate the following.

1) EMT/ET can significantly outperform EMEA, MFEA, SPEA2, and NSGA-II.
2) The selection of transferred solutions in EMT/ET is capable of selecting the transferred solutions containing valuable knowledge to improve the performance of MTO.

The following details the experimental setup, experimental results, and discussion.

A. Experimental Settings

We will conduct five sets of experiments to: 1) evaluate the performance of EMT/ET in comparison to the recently proposed EMEA, MFEA, NSGA-II, and SPEA2, where NSGA-II and SPEA2 are used to solve one task of each test problem, separately; 2) analyze the effectiveness of the selection of transferred solutions in EMT/ET; 3) illustrate how the transferred solutions (selected by our proposed method) affect the optimization of their target tasks; 4) evaluate the proportion of positive transfer of the proposed method during the evolution process; and 5) further, evaluate the effectiveness of the proposed method during evolution.

For the second experiment, the validity of the selection of transferred solutions in EMT/ET will be compared with the other two strategies widely used in existing MTO algorithms: 1) selecting some nondominated solutions in each task as the transferred solutions and 2) randomly selecting the transferred solutions in each task. Therefore, three compared algorithms are constructed, namely, MTO-K, MTO-N, and MTO-P. MTO-K denotes an MTO algorithm using our proposed selection method, and MTO-N and MTO-P denote the algorithms with the first strategy and the second strategy, separately. First, in these three algorithms, multiple populations are used to solve multiple optimization tasks of an MTO problem by using NSGA-II. Then, for any two optimization tasks, n solutions in one task will be transferred to the other task based on (2) during the evolution process.

For the third experiment, we analyze whether the transferred solutions selected by the proposed method achieve positive transfer on all test problems. First, the populations of each task and the solutions which are selected for knowledge transfer in these populations will be shown. Second, the distribution of the transferred solutions in their target tasks will also be presented, which can demonstrate whether these solutions achieve positive transfer. Finally, the effect of the transferred solutions on their target tasks will be evaluated.

For the fourth experiment, we compare the proportion of positive transfer respectively obtained (during the evolution) by our proposed method and MFEA. In this experiment, the proportion of positive transfer is calculated every 50 generations. The proportion of positive transfer is equal to the number of positive-transfer solutions divided by the total number of transferred solutions. The independent number of runs is 30. Finally, averaged proportion of positive transfer (over 30 runs) will be plotted.

In the fifth experiment, the effectiveness of the proposed method for selecting transferred solutions during the evolution process is analyzed. There are three compared algorithms in this experiment. The first one is the algorithm using the proposed method to select transferred solutions. In the second algorithm, nondominated solutions are transferred across tasks, while the third one is the algorithm without transferring solutions between tasks. The inverted generation distance (IGD) values obtained by these three algorithms in every generation will be recorded. The following two points will be demonstrated.

1) Transferring useful solutions between tasks will improve the performance of optimizing related tasks.
2) The proposed method is capable of finding more value solutions for knowledge transfer.

1) Test Problems: In this article, the CEC 2017 Evolutionary MultiTask Optimization Competition [11] benchmarks contain nine problems that were used. The benchmarks containing nine multiobjective multitasking optimization problems are designed by considering the degree of the intersection of the Pareto-optimal solutions and the similarity in the fitness landscape between optimization tasks. There are three degrees of the intersection, which are complete intersection, partial intersection, and no intersection, respectively. There are also three degrees of similarity, including high, medium, and low. For example, CIHS denotes a multiobjective multitasking problem in which the fitness landscape of two optimization tasks is highly similar to each other, and the Pareto-optimal solutions of them are a complete intersection. Each multiobjective multitasking optimization problem is combined with two multiobjective optimization problems (with two or three objectives), which have been commonly studied in the literature. Table I shows the detailed properties of these nine test problems, which includes Pareto set and Pareto front.

TABLE I
PROPERTIES OF MULTIOBJECTIVE MULTITASKING OPTIMIZATION PROBLEMS

Problem	Task NO.	Pareto Set	Pareto Front	Properties
CIHS	T1	$x_1 \in [0,1]$, $x_i = 0, i = 2:50$	$f_1^2 + f_2^2 = 1$, $f_1 \geq 0, f_2 \geq 0$	concave,unimodal, separable
	T2	$x_1 \in [0,1]$, $x_i = 0, i = 2:50$	$f_2 = 1 - f_1^2$, $0 \leq f_1 \leq 1$	concave,unimodal, separable
CIMS	T1	$x_1 \in [0,1]$, $x_i = 1, i = 2:10$	$f_2 = 1 - f_1^2$, $0 \leq f_1 \leq 1$	concave,multimodal, nonseparable
	T2	$x_1 \in [0,1]$, $(x_2, ..., x_{10})^T = S_{cm2}$	$f_1^2 + f_2^2 = 1$, $f_1 \geq 0, f_2 \geq 0$	concave,unimodal, nonseparable
CILS	T1	$x_1 \in [0,1]$, $x_i = 0, i = 2:50$	$f_1^2 + f_2^2 = 1$, $f_1 \geq 0, f_2 \geq 0$	concave,multimodal, separable
	T2	$x_1 \in [0,1]$, $x_i = 0, i = 2:50$	$f_2 = 1 - \sqrt{f}_1$ $0 \leq f_1 \leq 1$	concave,multimodal, separable
PIHS	T1	$x_1 \in [0,1]$, $x_i = 0, i = 2:50$	$f_2 = 1 - \sqrt{f}_1$ $0 \leq f_1 \leq 1$	concave,unimodal, separable
	T2	$x_1 \in [0,1]$, $(x_2, ..., x_{50})^T = S_{ph2}$	$f_2 = 1 - \sqrt{f}_1$ $0 \leq f_1 \leq 1$	concave,multimodal, separable
PIMS	T1	$x_1 \in [0,1]$, $(x_2, ..., x_{50})^T = S_{pm1}$	$f_1^2 + f_2^2 = 1$, $f_1 \geq 0, f_2 \geq 0$	concave,unimodal, nonseparable
	T2	$x_1 \in [0,1]$, $x_i = 0, i = 2:50$	$f_2 = 1 - f_1^2$, $0 \leq f_1 \leq 1$	concave,multimodal, nonseparable
PILS	T1	$x_1 \in [0,1]$, $x_i = 0, i = 2:50$	$f_1^2 + f_2^2 = 1$, $f_1 \geq 0, f_2 \geq 0$	concave,multimodal, nonseparable
	T2	$x_1 \in [0,1]$, $(x_2, ..., x_{50})^T = S_{pl2}$	$f_1^2 + f_2^2 = 1$, $f_1 \geq 0, f_2 \geq 0$	concave,multimodal, nonseparable
NIHS	T1	$x_1 \in [0,1]$, $x_i = 1, i = 2:50$	$f_1^2 + f_2^2 = 1$, $f_1 \geq 0, f_2 \geq 0$	concave,multimodal, nonseparable
	T2	$x_1 \in [0,1]$, $x_i = 0, i = 2:50$	$f_2 = 1 - \sqrt{f}_1$ $0 \leq f_1 \leq 1$	convex,unimodal, separable
NIMS	T1	$x_1, x_2 \in [0,1]$, $x_i = 1, i = 3:20$	$\sum_{i=1}^{3} f_i^2 = 1$, $f_i \geq 0, i = 1,2,3$	concave,multimodal, nonseparable
	T2	$x_1 \in [0,1]$, $x_i = 0, i = 3:20$	$f_2 = 1 - f_1^2$, $0 \leq f_1 \leq 1$	concave,unimodal, nonseparable
NILS	T1	$x_1, x_2 \in [0,1]$, $(x_3, ..., x_{25})^T = S_{nl1}$	$\sum_{i=1}^{3} f_i^2 = 1$, $f_i \geq 0, i = 1,2,3$	concave,multimodal, nonseparable
	T2	$x_1 \in [0,1]$, $x_i = 0, i = 3:50$	$f_2 = 1 - f_1^2$, $0 \leq f_1 \leq 1$	concave,multimodal, nonseparable

2) Performance Indicator: Two performance metrics, that is, the hypervolume metric (HV) [30] and IGD metric, are used to evaluate the performance of all the compared algorithms. HV evaluates the diversity of the obtained population along the Pareto front. A larger HV value indicates that the obtained solutions have a better approximation to the Pareto front in the objective space. In this article, since some compared algorithms cannot obtain the solutions near the Pareto front, the reference points of HV for biobjective and triobjective problems are set to (5, 5) and (5, 5, 5), respectively. IGD is used to evaluate the average distance from the points in the

TABLE II
SMALLEST AND AVERAGE VALUES (SHOWN IN THE BRACKETS) OF IGD OBTAINED BY EMT/ET, EMEA, MFEA, SPEA2, AND NSGA-II ON THE NINE MTO BENCHMARKS

Problem	Task	EMT/ET	EMEA	MFEA	SPEA2	NSGA-II
CIHS	T1	1.90E-04(2.11E-04)	**1.67E-04(1.85E-04)**	1.60E-03(3.50E-03)	3.59E-03(6.18E-03)	-
	T2	1.77E-04(2.04E-04)	**1.80E-04(1.94E-04)**	5.90E-03(9.10E-03)	-	3.90E-03(5.20E-03)
CIMS	T1	**1.73E-04(1.94E-04)**	1.73E-04(4.41E-04)	1.78E-04(4.72E-04)	9.64E-04(4.63E-02)	-
	T2	**1.70E-04(3.16E-04)**	9.97E-04(0.0148)	2.04E-04(6.02E-04)	-	9.00E-04(1.08E-02)
CILS	T1	**1.70E-04(1.88E-04)**	1.73E-04(1.92E-04)	1.89E-04(1.93E-04)	1.17E-02(7.35E-02)	-
	T2	1.74E-04(1.92E-04)	**1.72E-04(1.90E-04)**	1.84E-04(1.96E-04)	-	8.96E-04(1.03E-03)
PIHS	T1	**1.70E-04(1.88E-04)**	7.12E-04(2.60E-03)	2.59E-02(5.06E-02)	2.58E-03(3.96E-03)	-
	T2	**1.67E-04(7.21E-04)**	3.12E-02(1.57E-01)	5.84E-01(1.12)	-	1.05E-01(1.94E-01)
PIMS	T1	**1.74E-04(2.81E-04)**	8.82E-04(2.80E-03)	2.40E-03(4.70E-03)	3.78E-02(7.15E-02)	-
	T2	**1.71E-04(2.06E-04)**	6.8369(7.527)	7(10.461)	-	2.14E+00(4.37E+00)
PILS	T1	**1.71E-04(1.89E-04)**	2.32E-04(3.44E-04)	9.91E-04(1.60E-03)	7.80E-04(1.98E-03)	-
	T2	**3.11E-04(1.34E-03)**	3.53E-02(6.01E-02)	6.27E-02(8.21E-02)	-	2.00E-01(2.01E-01)
NIHS	T1	**1.47E+00(1.48E+00)**	1.49E+00(1.49E+00)	2.03E+00(2.83E+00)	2.81E+00(9.19E+00)	-
	T2	1.77E-04(1.88E-04)	**1.72E-04(1.86E-04)**	3.20E-03(8.30E-03)	-	1.60E-03(2.60E-03)
NIMS	T1	1.48E-01(1.55E-01)	1.60E-01(1.64E-01)	**9.90E-02(1.53E-01)**	7.06E-02(6.22E-01)	-
	T2	**1.41E-04(1.50E-04)**	1.42E-04(1.55E-04)	3.31E-04(1.80E-02)	-	0.300E-03(4.29E-02)
NILS	T1	6.43E-04(7.53E-04)	6.83E-04(7.56E-04)	9.27E-04(1.6E-03)	**5.26E-04(5.60E-04)**	-
	T2	**4.51E-04(1.11E-02)**	6.42E-01(6.42E-01)	6.43E-01(6.44E-01)	-	2.02E-01(2.03E-01)

Pareto-optimal set to the obtained solutions. A smaller IGD value indicates that the obtained population is closer to Pareto front with good distribution. The IGD is calculated as follows:

$$\mathrm{IGD}(V, H_{\mathrm{eff}}) = \frac{1}{|H_{\mathrm{eff}}|}\sum_{i=1}^{|H_{\mathrm{eff}}|}\min_{j=1}^{|V|} d\big(h_i, v_j\big) \tag{3}$$

where $V = [v_1, v_2, \ldots]$ denotes the nondominated individuals obtained by a multiobjective optimization evolutionary algorithm in the objective space and $H_{\mathrm{eff}} = [h_1, h_2, \ldots]$ denotes a set of sampled solutions of the known Pareto front. $d(h_i, v_j) = ||h_i - v_j||$ denotes the Euclidean distance from h_i to v_j. The IGD metric is capable of measuring both diversity and convergence of V when $|H_{\mathrm{eff}}|$ is sufficiently large. Here, $|H_{\mathrm{eff}}|$ is set to 1000 for two objective problems and 10 000 for three objective problems.

3) Parameter Settings: The algorithms are implemented in MATLAB R2014a and run on a PC with an Intel Core I7-8550U 1.80-GHz CPU, and 8.00 GB of RAM.

1) *Population Size:*
   a) *For the First Experiment:* Population size of each solver, that is, NSGAII and SPEA2, EMT/ET, and EMEA is set to 100 for two-objective optimization problems and 120 for three-objective optimization problems, while the population size of MFEA is set to 200 for CIHS-NIHS and 240 for NIMS-NILS.
   b) *For the Second Experiment:* Population size of each solver, that is, MTO-K, MTO-N, and MTO-P, is set to 100 for two-objective optimization problems and 120 for three-objective optimization problems.
2) *Maximum Generations:* MaxGen = 500.
3) *Independent Number of Runs:* runs = 30.
4) The parameter settings of SBX and polynomial mutation for all algorithms are $p_c = 0.9$, $p_m = 1/D$, $\eta_c = 20$, and $\eta_m = 15$, where D is the dimensionality of the encoded solution.
5) The random mating probability (rmp) in MFEA is set to 0.3 according to [12].
6) $n = 8$ (for all experiments), $\lambda \sim U(0, 2)$, $p = 0.5$, since they give the best performance based on our initial experiments.

B. Experimental Results on the Test Suites

1) Comparison of EMT/ET, EMEA and MFEA, and NSGAII as well as SPEA2 on Nine Test Problems: Nine test problems are used to verify the performance of EMT/ET. The experimental results obtained by the proposed algorithm–EMEA, MFEA, NSGA-II, and SPEA2–are summarized in Tables II and III. In these two tables, the best result of each task is highlighted.

Table II shows the best and mean values (shown in the brackets) of the IGD metric obtained by all of the algorithms. As shown in this table, the overall performance of three MTO algorithms (i.e., EMT/ET, EMEA, and MFEA) is better than that of NSGA-II and SPEA2. These results confirm that the knowledge transfer among tasks can accelerate convergence in these problems. The advantages of the EMT/ET algorithm also can be observed intuitively from this table. Results of the nine test problems show that EMT/ET outperforms EMEA and MFEA in solving most problems.

For all problems except CIHS, CILS-T2, NIHS-T2, NIMS-T1, and NILS-T1, EMT/ET achieves smaller IGD values (best and average) than those of the other algorithms. For CIMS-T1, the best IGD value achieved by EMT/ET is not smaller than the other algorithms. However, the average IGD value of EMT/ET on this problem is significantly better than that of the other compared algorithms. That is, EMT/ET can find better solutions to these problems.

Table III gives the largest and mean HV values (shown in the brackets) obtained by all of the algorithms. The data show that EMT/ET performs better than the other algorithms in most test problems. Remarkably, the proposed algorithm can achieve larger HV values (best and average) than the other algorithms for all test problems except CIMS, CILS-T2, NIHS, NIMS-T1, and NILS-T1. The experimental results in this table demonstrate that EMT/ET is able to obtain solutions with better diversity and domination relation.

TABLE III
LARGEST AND AVERAGE VALUES (SHOWN IN THE BRACKETS) OF HV OBTAINED BY EMT/ET, EMEA, MFEA, SPEA2, AND NSGA-II ON THE NINE MTO BENCHMARKS

Problem	Task	EMT/ET	EMEA	MFEA	SPEA2	NSGA-II
CIHS	T1	24.20(24.19)	**24.21(24.21)**	24.1(24.0)	23.51(22.86)	-
	T2	24.31(24.30)	**24.33(24.33)**	22.53(21.70)	-	21.1(20.24)
CIMS	T1	**24.33(24.32)**	24.33(24.22)	24.33(24.19)	23.83(5.26)	-
	T2	**24.23(24.20)**	24.20(23.21)	24.21(24.18)	-	24.15(21.33)
CILS	T1	**24.23(24.22)**	24.21(24.21)	24.10(24.05)	11.24(0.926)	-
	T2	24.60(24.56)	**24.66(24.65)**	24.65(24.64)	-	24.60(24.60)
PIHS	T1	**24.66(24.66)**	24.66(24.33)	23.11(15.0)	23.53(22.12)	-
	T2	**22.77(24.56)**	22.06(8.11)	0(0)	-	0(0)
PIMS	T1	**24.26(24.20)**	24.15(24.04)	24.08(23.95)	3.797(0.3528)	-
	T2	**24.33(24.33)**	0(0)	0(0)	-	0(0)
PILS	T1	**24.21(24.21)**	24.20(24.20)	24.13(24.05)	24.20(24.19)	-
	T2	**23.14(24.13)**	20.33(17.35)	12.85(9.033)	-	0(0)
NIHS	T1	0(0)	0(0)	0(0)	0(0)	-
	T2	24.66(24.65)	**24.67(24.66)**	24.00(23.35)	-	23.83(22.75)
NIMS	T1	0(0)	0(0)	**117.2(13.44)**	103.2(3.44)	-
	T2	**24.33(24.33)**	24.32(24.31)	24.26(19.33)	-	22.58(4.811)
NILS	T1	124.35(124.32)	124.35(124.32)	124.27(124.10)	**124.48(124.39)**	-
	T2	**24.04(20.6)**	0(0)	0(0)	-	0(0)

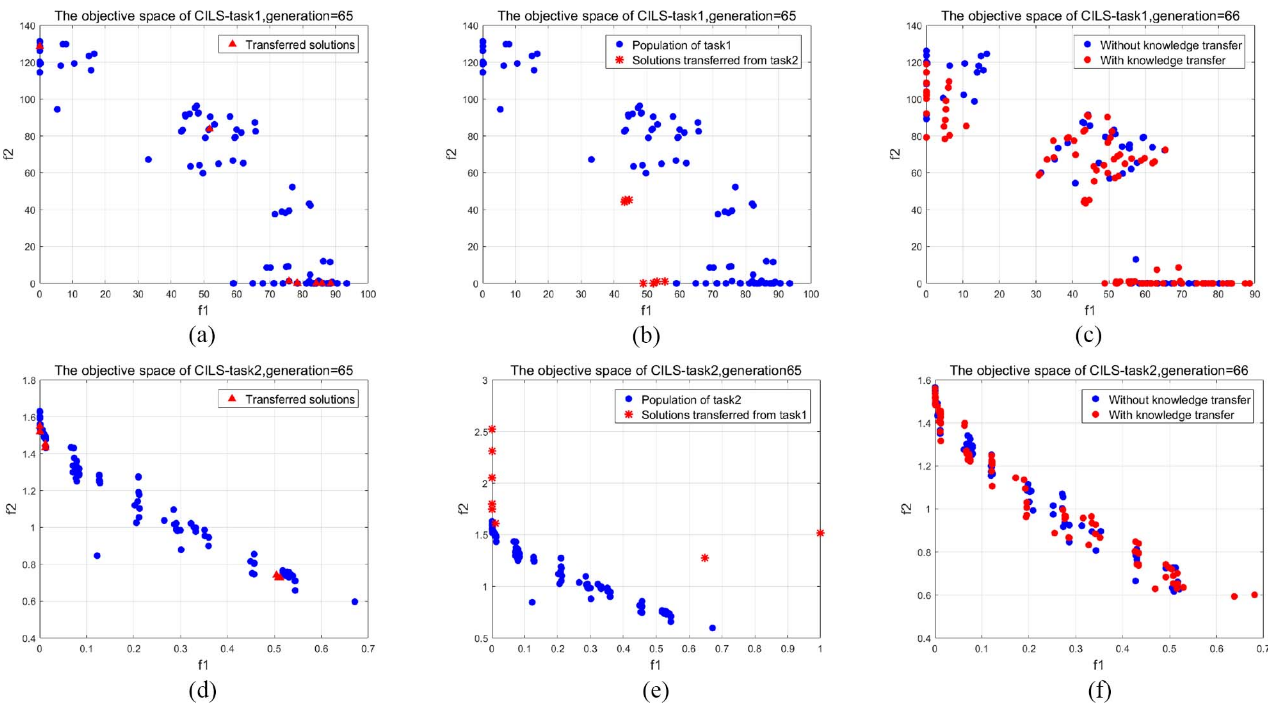

Fig. 2. Illustration of the transferred solutions and the population on representative benchmark—CILS. y-axis: f2; x-axis: f1. (a) Population of task1 including transferred solutions. (b) Distribution of solutions transferred from task2 to task1. (c) Comparison of the effect of knowledge transfer and without knowledge transfer. (d) Population of task2 including transferred solutions. (e) Distribution of solutions transferred from task1 to task2. (f) Comparison of the effect of knowledge transfer and without knowledge transfer.

EMT/ET performs worse than EMEA on problem CIHS both in terms of IGD value and HV value. CIHS consists of two tasks with the same Pareto-optimal solutions, and the high degree of similarity between the fitness landspace of $q(x)$ functions. Therefore, transferring nondominated solutions in each task to the other tasks is a useful way of knowledge transfer for solving this problem. Therefore, EMEA outperforms the other algorithms on problem CIHS.

Overall, these two tables demonstrate that EMT/ET is consistently more stable, and has faster convergence speeds and superior converged solutions, than the compared algorithms, over the majority of experiments.

We believe the promising performance of EMT/ET is mainly due to 1) the strategy of selecting transferred solutions and 2) multiplying some transferred solutions by a distributed factor. It is worth noting that each test problem consists of an easy task and a difficult task. First, since every problem of CIHS-CILS is composed of two tasks having the same Pareto set, these problems can be solved when the easy task converges to the Pareto set. Therefore, the key for solving CIHS-CILS is to accelerate the convergence of the easy tasks. However, although the Pareto sets of two tasks are the same, the nondominated solutions of the easy task may not always help to optimize the difficult task. The approach of selecting transferred solutions proposed in this article is capable of exploring some valuable area in the search space. As can be observed from the results of CIHS-CILS, EMT/ET outperforms other compared algorithms on most problems. Second,

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

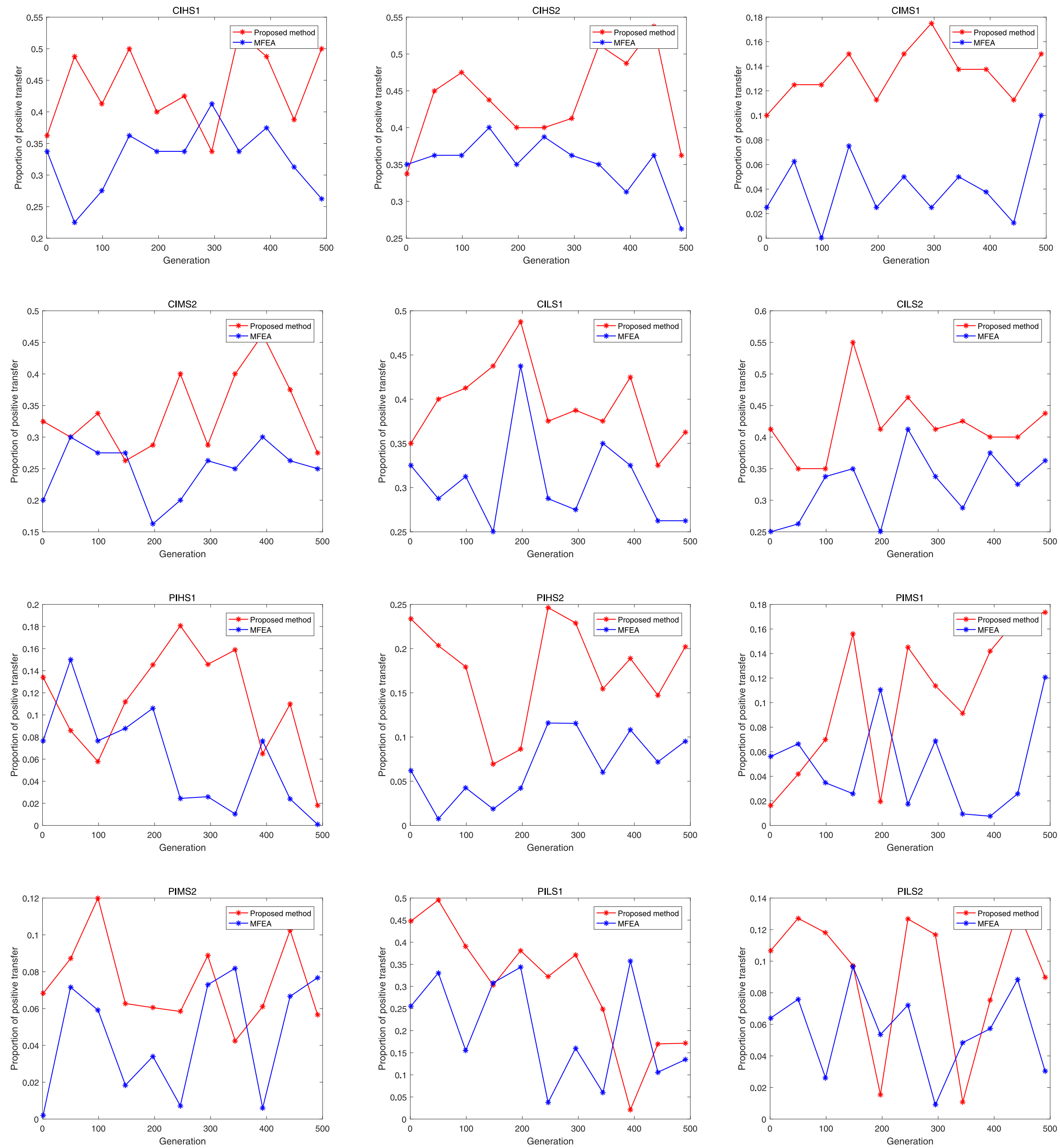

Fig. 3. Convergence curves of the averaged proportion of positive transfer (over 30 runs) obtained by the proposed method and MFEA on the nine multiobjective multitask problems—Part1. y-axis: Proportion of positive transfer; and x-axis: Generation.

every problem of PIHS-PILS consists of two tasks having a partial intersection in the Pareto sets. Since the Pareto sets of two tasks are different, the difficult tasks may be trapped in its local optima by the guide of the easy task. Hence, the easy task should be able to help the difficult task to jump out of the local traps. With the help of multiplying some transferred solutions and a distributed factor, the performance of EMT/ET in solving PIHS-PILS is significantly better than that of other compared algorithms. Finally, in every problem of NIHS-NILS, two optimization tasks do not have an intersection in the Pareto sets. As a result, knowledge transfer may not be effective in improving the performance of difficult tasks. As can be observed from the experimental results, EMT/ET still outperforms other compared algorithms on most of the problems of NIHS-NILS.

2) Analysis of the Selection of Transferred Solutions in EMT/ET: We conduct a further experiment to gain insight into the selection of transferred solutions in EMT/ET. The experimental results obtained by MTO-K, MTO-N, and MTO-P are summarized in Table IV. In this table, the best result of each task is highlighted.

Table IV shows the best and mean values (shown in the brackets) of the IGD metric obtained by three compared algorithms. The performance superiority of MTO-K over the other compared algorithms can be evidenced by the fact that MTO-K wins 11 out of the 18 tasks, and loses 7 out of the 18

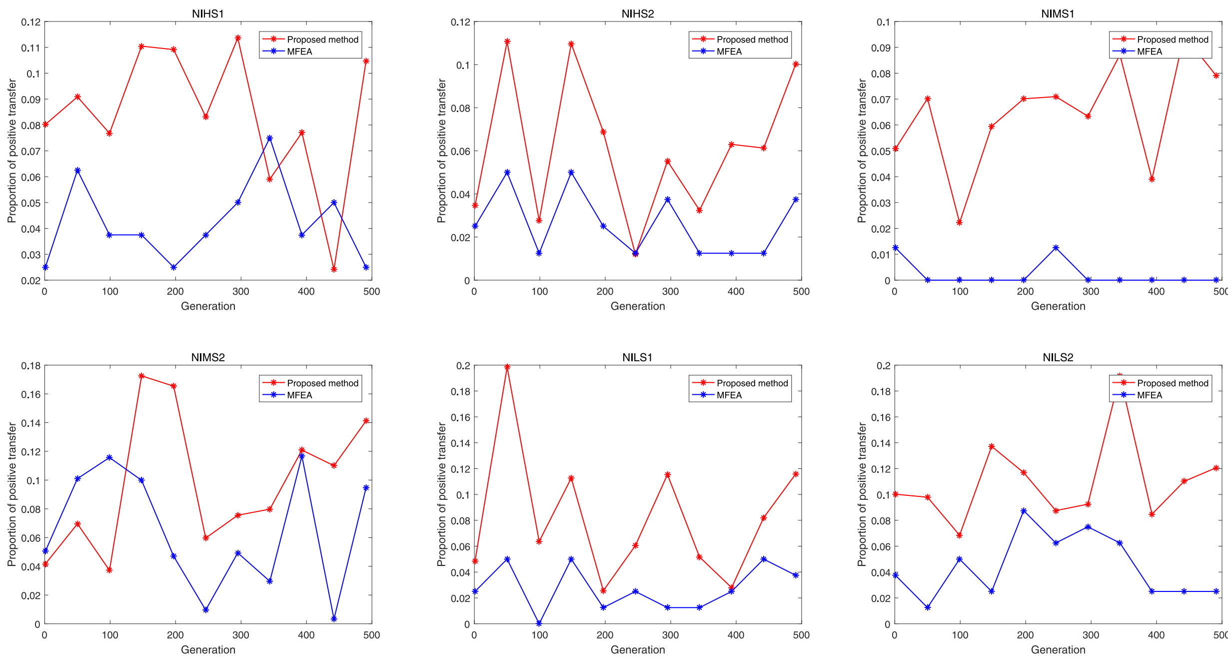

Fig. 4. Convergence curves of the averaged proportion of positive transfer (over 30 runs) obtained by the proposed method and MFEA on the nine multiobjective multitask problems—Part2. *y*-axis: Proportion of positive transfer; and *x*-axis: Generation.

TABLE IV
SMALLEST AND AVERAGE VALUES (SHOWN IN THE BRACKETS) OF IGD OBTAINED BY MTO-K, MTO-N, AND MTO-P ON THE NINE MTO BENCHMARKS

Problem	Task	MTO-K	MTO-N	MTO-P
CIHS	T1	**1.74E-04(6.18E-04)**	1.77E-04(6.24E-04)	1.79E-04(6.89E-04)
	T2	**4.80E-04(2.15E-03)**	5.64E-04(2.37E-03)	5.17E-04(2.44E-03)
CIMS	T1	**1.79E-04(2.04E-04)**	1.90E-04(2.36E-04)	1.74E-04(6.01E-04)
	T2	**1.91E-04(3.28E-04)**	1.93E-04(5.39E-04)	2.48E-04(1.60E-03)
CILS	T1	**1.75E-04(2.17E-04)**	1.85E-04(2.77E-04)	1.78E-04(3.07E-04)
	T2	1.83E-04(2.00E-04)	1.85E-04(1.97E-04)	**1.76E-04(1.88E-04)**
PIHS	T1	1.83E-04(3.91E-03)	**1.68E-04(2.23E-03)**	1.77E-04(3.29E-03)
	T2	**3.75E-04(1.20E-02)**	1.39E-03(7.93E-02)	4.91E-04(7.16E-02)
PIMS	T1	2.80E-04(1.31E-03)	4.15E-04(1.34E-03)	**2.83E-04(7.30E-04)**
	T2	7.42E+00(8.29E+00)	**6.77E+00(7.84E+00)**	9.64E+00(1.23E+01)
PILS	T1	1.74E-04(2.15E-04)	**1.74E-04(2.13E-04)**	3.64E-04(4.34E-04)
	T2	**7.92E-04(6.91E-03)**	7.97E-04(7.01E-03)	8.43E-04(7.11E-03)
NIHS	T1	**1.49E+00(1.57E+00)**	1.48E+00(1.59E+00)	1.49E+00(1.58E+00)
	T2	**1.80E-04(5.69E-04)**	1.85E-04(8.85E-04)	1.82E-04(7.15E-04)
NIMS	T1	1.57E-01(1.60E-01)	**9.82E-03(5.99E-02)**	7.68E-01(8.00E-01)
	T2	**1.97E-04(2.86E-04)**	8.70E-03(7.04E-02)	2.82E-03(1.19E-02)
NILS	T1	**6.21E-04(6.39E-04)**	6.54E-04(6.99E-04)	6.45E-04(6.96E-04)
	T2	6.42E-01(6.42E-01)	6.14E-02(6.50E-02)	**4.89E-04(1.20E-03)**

tasks. For CIMS-T1 and PILS-T2, although the best IGD values obtained by MTO-K are not smaller than that obtained by the other algorithms, MTO-K can obtain smaller average IGD values than the other algorithms.

As mentioned above, each problem of CIHS-CIMS consists of two optimization tasks with the same Pareto-optimal solutions. Therefore, the nondominated solutions in each task can help to optimize the other task during the process of solving these problems. As can be seen in Table IV, MTO-N outperforms MTO-P in solving CIHS-CIMS. However, only transferring the nondominated solutions in each task does not always help to optimize the other tasks. The results show that MTO-K is better than MTO-N at solving CIHS-CIMS. For PIHS-PIMS, all the compared algorithms do not perform well. The reason is that the key for solving problem PIHS-PIMS is to help the tasks to jump out of the local optima. Finally, MTO-K wins four out of the six tasks when solving NIHS-NILS. It can be demonstrated that a strategy adaptively explores some valuable area for the positive knowledge transfer is essential for MTO.

3) Analysis of the Distribution of Transferred Solutions (Selected by the Proposed Method) in Their Target Tasks: We analyze the positive transfer achieved by the proposed method on all the test problems to give a deeper insight into the performance obtained by it. For each problem, six figures [Fig. 2(a)–(f)] are presented, respectively. In Fig. 2(a) and (b), the populations, including blue dots and red triangles of each task in the objective space, are presented. The red triangles denote the solutions which are selected for knowledge transfer in these populations. In Fig. 2(b) and (e), the distribution of transferred solutions in the populations of their target tasks is shown. In these figures, blue dots denote the populations of the target tasks, while the red stars represented the solutions transferred from the other one task. Fig. 2(c) and (f) both illustrate the evolution results of with or without transferring the selected solutions between tasks. The red dots denote the populations which are evolved from the population injected with transferred solutions in the last generation. The blue dots in these two figures represent the evolution results of the populations which were not injected with transferred solutions in the last generation. Fig. 2 shows how the transferred solutions selected by our proposed method affect the evolution of the benchmark problem CILS. It should be noted that all the nine problems has been analyzed in this article. However, due to the space limit, the figures for the rest of the test problems are shown in the supplementary material. First, since each of these benchmarks consists of two related tasks (an easy task and a difficult task), evolutionary multitasking aims to let these tasks help each other during the evolution.

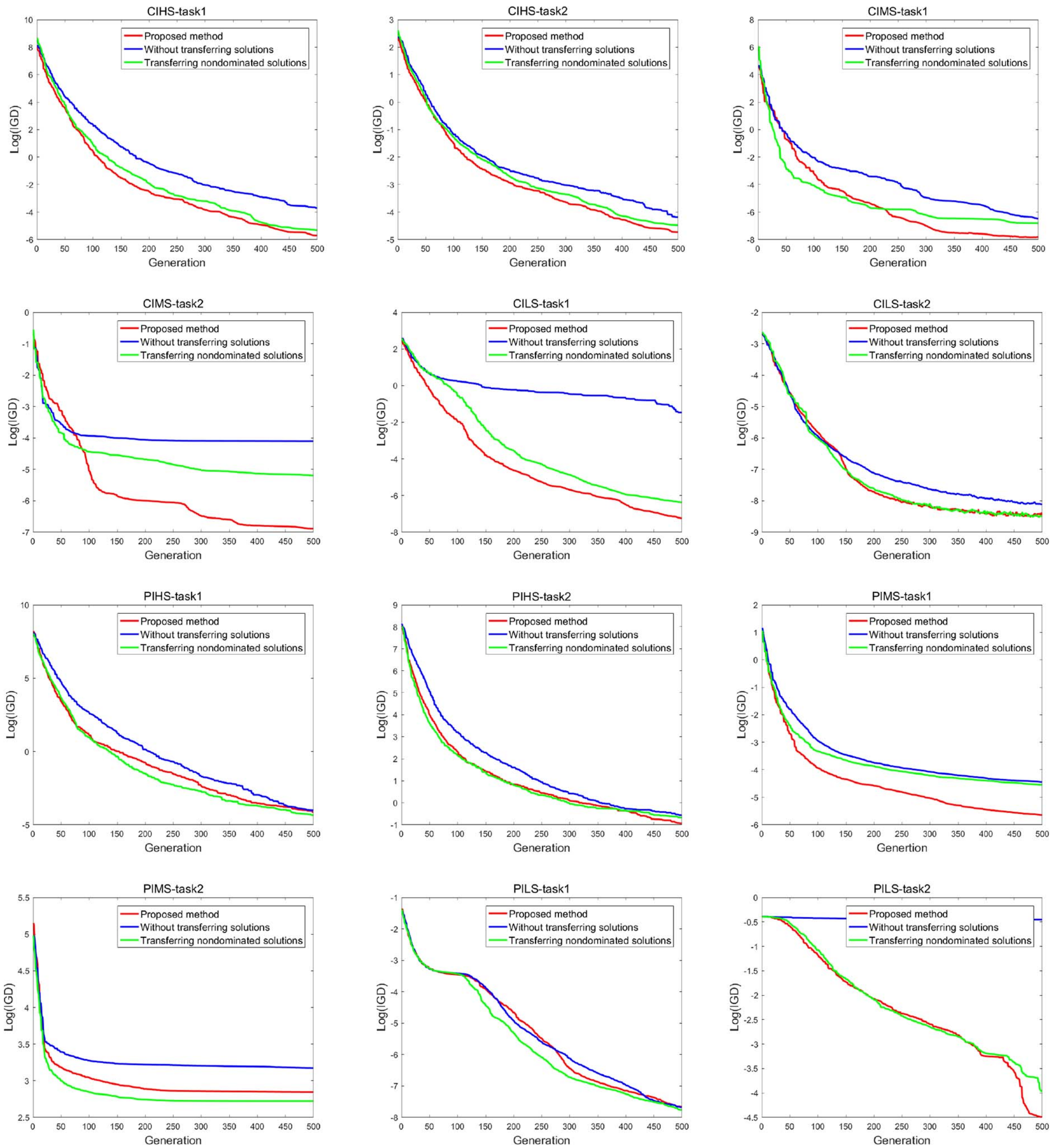

Fig. 5. Convergence curves of the averaged log(IGD) (over 30 runs) obtained by three algorithms on the nine multiobjective multitask problems—Part1. y-axis: Log(IGD); and x-axis: Generation.

As can be seen in Fig. 2(b), all the solutions transferred from CILS-task2 are nondominated in CILS-task1, thereby achieving positive transfer. Therefore, the proportion of positive transfer is 1. However, as can be observed in Fig. 2(e), all the solutions transferred from CILS-task1 are dominated in CILS-task2. The reason is that the CILS-task2 is the easy task in problem CILS, which makes it difficult for CILS-task2 to help the CILS-task1 (the difficult task). Furthermore, as can shown in Fig. 2(c), the population of CILS-task1 can be evolved better by the help of CILS-task2. For solving six problems (CIHS, CIMS, PIHS, PIMS, and NIHS), please see in the supplementary material, most of the transferred solutions from one task are nondominated in the other one task, thus achieving positive transfer. The populations which are injected with the positive-transfer solutions can be evolved better. However, the proposed method does not perform well on the problems PILS, NIMS, and NILS, which can be observed by the fact that the transferred solutions have a negative effect on their target tasks. For these three problems (PILS, NIMS, and NILS), all the transferred solutions are dominated in their target tasks. For the problems PILS and NILS, the Pareto set of the easy task is $x_1 \in [0, 1]$, $x_2 = \cdots = x_{50} = 0$, while the Pareto set of the difficult task is $x_1 \in [0, 1]$, $x_2 = \cdots = x_{25} = 0$, $x_{26} = \cdots = x_{50} = 20$. During the evolution, the population of the easy task will search in an area which is far from the Pareto set of the difficult task. Therefore, in the population of the easy task, there rarely exists some information that is useful for the difficult task. For the problem NIMS, the Pareto set of the easy task

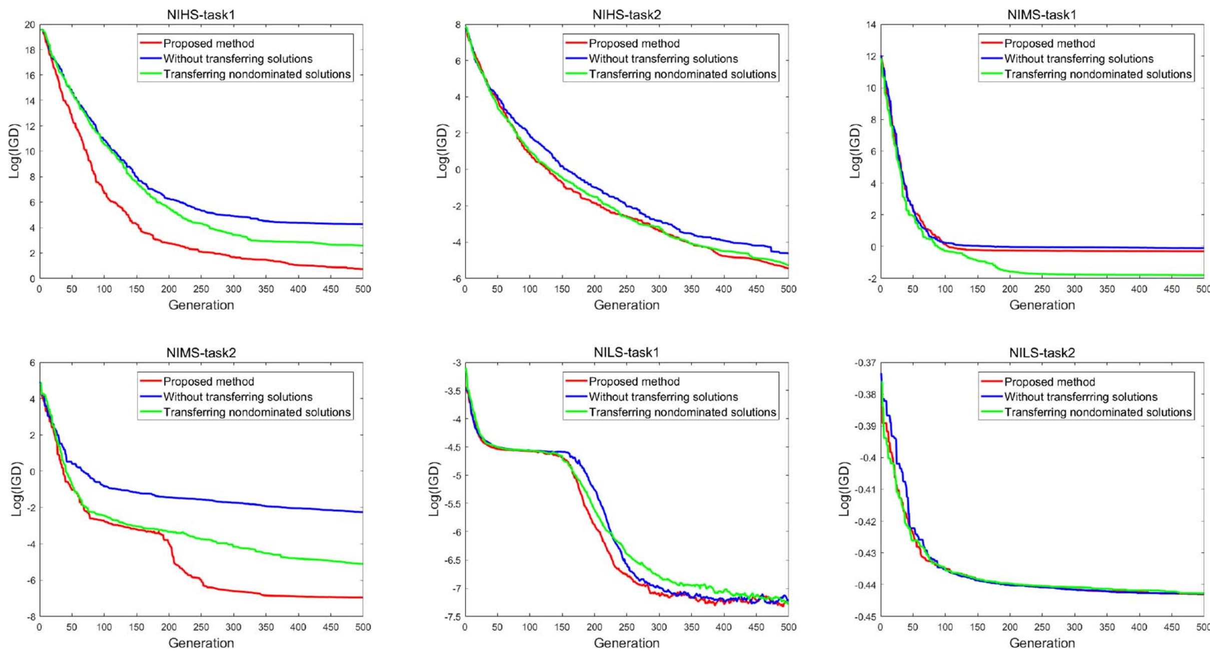

Fig. 6. Convergence curves of the averaged log(IGD) (over 30 runs) obtained by three algorithms on the nine multiobjective multitask problems—Part2. y-axis: Log(IGD); and x-axis: Generation.

is $x_1 \in [0, 1]$, $x_2 = \cdots = x_{20} = 0$, while the Pareto set of the difficult task is $x_1 \in [0, 1]$, $x_2 = \cdots = x_{20} = 1$. Although the Pareto sets of these two tasks are similar, the Pareto set of the easy task is the local Pareto set of the difficult task. Therefore, the population of the easy task also cannot provide some useful solutions for helping the evolution of the difficult task.

4) Analysis of the Proportion of Positive Transfer Obtained by Proposed Method and MFEA: We compare the proportion of positive transfer obtained by the proposed method and MFEA in order to compare their ability to select valuable solutions for knowledge transfer. Figs. 3 and 4, respectively, show the averaged proportion of positive transfer obtained by two algorithms during the evolutionary process on nine test problems. In these figures, the red stars denote the proportion of positive transfer obtained by the proposed method, while the blue stars represent the proportion of positive transfer obtained by MFEA. As can be observed, generally, our proposed method outperforms MFEA on most of the test problems in terms of the proportion of positive transfer. Especially for each of the problems CIHS1, CIHS2, and CILS2, the averaged proportions of positive transfer obtained by our proposed method are nearly 0.5, while that of MFEA is around 0.35. Furthermore, during the evolution, although there are some fluctuations in the proportion of positive transfer obtained by our proposed method, the proposed method still has better performance than MFEA. Compared with MFEA, our proposed method is capable of selecting more solutions from the source tasks to help the target tasks, thereby achieving more positive transfer.

5) Analysis of the Effectiveness of the Proposed Method During the Evolution Process: Further, the log(IGD) values obtained by the algorithm using the proposed method (for selecting transferred solutions) in each generation are compared with that obtained by the other two algorithms. In the first compared algorithm, n nondominated solutions in each task are transferred to the other one task. In the second algorithm, there is no solution transferred between tasks. Figs. 5 and 6 illustrate the log(IGD) values obtained by these three algorithms in every generation. In these figures, the red line denotes the log(IGD) values obtained by the algorithm which uses our proposed method to select transferred solutions. The green line denotes the log(IGD) values obtained by the algorithm in which nondominated solutions are transferred across tasks. The blue line denotes the log(IGD) values obtained by the algorithm without transferring solutions between tasks. It can be demonstrated by the experimental results that algorithms can converge much faster and obtain much better results with the help of transferring solutions across related tasks. Besides, as can be observed, our proposed method outperforms the other two compared algorithms on most of the test problems. It is also demonstrated that the proposed method is capable of not only achieving positive transfer but also significantly improve the performance of optimization.

IV. Conclusion

With the primary purpose of transferring valuable knowledge across tasks, this article proposed a new method for solving MTO problems. The proposed algorithm has been applied to nine MTO test problems. Each test problem consists of two different multiobjective optimization problems. These multiobjective optimization problems are designed based on the degree of the intersection of Pareto-optimal solutions and the similarity in the fitness landscape. In most of these problems, the proposed algorithm has been able to successfully find a well-converged and well-diversified set of points repeatedly over multiple runs. It is achieved mainly due to the strategy of finding more solutions containing the knowledge of value for the positive transfer among tasks. Furthermore, the performance of EMT/ET has been compared with several algorithms, which include EMEA, MFEA, SPEA2, and NSGA-II. The experimental results confirmed the validity of EMT/ET for solving MTO problems. Moreover, we analyzed the selection of transferred solutions in EMT/ET to gain an insight into its effectiveness.

It has been proven that positive transfer can be achieved by the proposed method so that the performance of the multiobjective multitasking optimization is improved.

REFERENCES

[1] C. A. Coello Coello and G. B. Lamont, *Applications of Multi-Objective Evolutionary Algorithms*, vol. 1. Singapore: World Sci., 2004.

[2] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, vol. 5. New York, NY, USA: Springer, 2007.

[3] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Struct. Multidiscipl. Optim.*, vol. 26, no. 6, pp. 369–395, 2004.

[4] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, vol. 16. Hoboken, NJ, USA: Wiley, 2001.

[5] F. Gu, H.-L. Liu, Y.-M. Cheung, and S. Xie, "Optimal WCDMA network planning by multiobjective evolutionary algorithm with problem-specific genetic operation," *Knowl. Inf. Syst.*, vol. 45, no. 3, pp. 679–703, 2015.

[6] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 264–283, Apr. 2015.

[7] T. M. Lakshmi, A. Martin, and V. P. Venkatesan, "A genetic bankrupt ratio analysis tool using a genetic algorithm to identify influencing financial ratios," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 38–51, Feb. 2016.

[8] V. A. Cicirello, "The permutation in a haystack problem and the calculus of search landscapes," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 434–446, Jun. 2016.

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

[10] K. Deb, "Multi-objective optimization," in *Search Methodologies*. New York, NY, USA: Springer, 2014, pp. 403–449.

[11] Y. Yuan *et al.*, "Evolutionary multitasking for multiobjective continuous optimization: Benchmark problems, performance metrics and baseline results," 2017. [Online]. Available: arXiv:1706.02766.

[12] 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, Jul. 2017.

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

[14] R. Caruana, "Multitask learning," in *Learning to Learn*. Boston, MA, USA: Springer, 1998, pp. 95–133.

[15] E. V. Bonilla, K. M. A. Chai, and C. K. I. Williams, "Multi-task Gaussian process prediction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 153–160.

[16] 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.

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

[18] M. Gong, Z. Tang, H. Li, and J. Zhang, "Evolutionary multitasking with dynamic resource allocating strategy," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 858–869, Oct. 2019.

[19] 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. (CEC)*, 2017, pp. 2266–2273.

[20] K. K. Bali, A. Gupta, L. Feng, Y. S. Ong, and T. P. Siew, "Linearized domain adaptation in evolutionary multitasking," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2017, pp. 1295–1302.

[21] L. Feng *et al.*, "Evolutionary multitasking via explicit autoencoding," *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3457–3470, Sep. 2019.

[22] L. Zhou, L. Feng, J. Zhong, Y.-S. Ong, Z. Zhu, and E. Sha, "Evolutionary multitasking in combinatorial search spaces: A case study in capacitated vehicle routing problem," in *Proc. IEEE Symp. Series Comput. Intell. (SSCI)*, 2016, pp. 1–8.

[23] Y. Yuan, Y.-S. Ong, A. Gupta, P. S. Tan, and H. Xu, "Evolutionary multitasking in permutation-based combinatorial optimization problems: Realization with TSP, QAP, LOP, and JSP," in *Proc. IEEE Region 10 Conf. (TENCON)*, 2016, pp. 3157–3164.

[24] Y. Chen, J. Zhong, L. Feng, and J. Zhang, "An adaptive archive-based evolutionary framework for many-task optimization," *IEEE Trans. Emerg. Topics Comput. Intell.*, early access, doi: 10.1109/TETCI.2019.2916051.

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

[26] C. Yang, J. Ding, K. C. Tan, and Y. Jin, "Two-stage assortative mating for multi-objective multifactorial evolutionary optimization," in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, 2017, pp. 76–81.

[27] 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.

[28] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evol. Methods Design Optim. Control Appl. Ind. Problems (EUROGEN)*, Athens, Greece, Sep. 2001.

[29] K. Deb, 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.

[30] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms: A comparative case study," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, 1998, pp. 292–301.

Jiabin Lin is currently pursuing the master's degree with the School of Applied Mathmatics, Guangdong University of Technology, Guangzhou, China.

His current research interests include evolutionary computation and multitask learning.

Hai-Lin Liu (Senior Member, IEEE) received the B.S. degree in mathematics from Henan Normal University, Xinxiang, China, the M.S. degree in applied mathematics from Xidian University, Xi'an, China, and the Ph.D. degree in control theory and engineering from the South China University of Technology, Guangzhou, China.

He is currently a Professor with the School of Applied Mathematics, Guangdong University of Technology, Guangzhou. He was a Postdoctoral Researcher with the Institute of Electronic and Information, South China University of Technology. His current research interests include evolutionary computation and optimization, wireless network planning and optimization, and their applications.

Kay Chen Tan (Fellow, IEEE) received the B.Eng. degree (First Class Hons.) in electronics and electrical engineering and the Ph.D. degree from the University of Glasgow, Glasgow, U.K., in 1994 and 1997, respectively.

He is a Full Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. He has published over 200 refereed articles and six books.

Dr. Tan is the Editor-in-Chief of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION. He was the Editor-in-Chief of *IEEE Computational Intelligence Magazine* from 2010 to 2013. He currently serves as an editorial board member of over ten journals.

Fangqing Gu received the B.S. degree from Changchun University, Changchun, China, in 2007, the M.S. degree from the Guangdong University of Technology, Guangzhou, China, in 2011, and the Ph.D. degree from the Department of Computer Science, Hong Kong Baptist University, Hong Kong, in 2016.

He joined the School of Applied Mathematics, Guangdong University of Technology as a Lecturer. His research interests include data mining, machine learning, and evolutionary computation.