# Multiobjective Multitasking Optimization with Decomposition-based Transfer Selection

Qiuzhen Lin, *Member, IEEE*, Zhongjian Wu, Lijia Ma, Maoguo Gong, *Senior Member, IEEE*, Jianqiang Li, and Carlos A. Coello Coello, *Fellow, IEEE*

***Abstract*—Multiobjective multitasking optimization needs to solve a set of multiobjective optimization problems simultaneously, and tries to speed up their solution by transferring useful search experiences across tasks. However, the quality of transfer solutions will significantly impact the transfer effect, which may even deteriorate the optimization performance with an improper selection of transfer solutions. To alleviate this issue, this paper suggests a new multiobjective multitasking evolutionary algorithm with decomposition-based transfer selection, called MMTEA-DTS. In this algorithm, all tasks are first decomposed into a set of subproblems, and then the transfer potential of each solution can be quantified based on the performance improvement ratio of its associated subproblem. Only high-potential solutions are selected to promote knowledge transfer. Moreover, to diversify the transfer of search experiences, a hybrid transfer evolution method is designed in this paper. In this way, more diverse search experiences are transferred from high-potential solutions across different tasks to speed up their convergence. Three well-known benchmark suites suggested in the competition of evolutionary multitasking optimization and one real-world problem suite are used to verify the effectiveness of MMTEA-DTS. The experiments validate its advantages in solving most of the test problems when compared to five recently proposed multiobjective multitasking evolutionary algorithms.**

***Index Terms*—Multitasking Optimization, Multiobjective Optimization, Knowledge Transfer, Decomposition.**

## I. Introduction

Optimization problems are ubiquitous in real-world applications [1]-[3]. These problems are challenging for traditional optimization methods to solve, owing to their complex characteristics with many practical constraints [4]-[6]. In particular, multiobjective optimization problems (MOPs) [7]-[9] pose a significant challenge since their objectives often conflict with each other, and thus a single solution cannot guarantee their optimality simultaneously. Evolutionary algorithms, which rely on population-based search, have been recognized as an efficient and effective tool for addressing MOPs [10]-[12]. However, with the emergence of cloud computing services, an on-demand optimization service is in high demand, which needs to solve a large number of optimization tasks requested by several customers concurrently [13]. In such cases, separately solving the tasks is inefficient, as there are common optimization experiences in tackling multiple tasks. Consequently, the study of multitasking optimization (MTO) has received significant attention in recent years [14]-[18].

As an emerging paradigm in the field of evolutionary computation, MTO focuses on solving multiple self-contained tasks simultaneously [19], [20], which is inspired by transfer learning and multitask learning in predictive analytics [21]. In recent studies of MTO, evolutionary computation has been widely employed, as this method can exploit the implicit parallelism of population-based search to simultaneously tackle multiple optimization tasks [19]. One pioneering work of Gupta *et al.* [20] introduces a multifactorial evolution paradigm and proposes the first MTO algorithm called MFEA, which aims to speed up convergence when solving several single-objective optimization tasks by transferring useful search experiences across them. Since then, some subsequent studies have been conducted to further improve MFEA, such as MTO-DRA [22], MFMP [23], MFEA-GSMT [24], and AT-MFEA [25]. More recently, the idea of MTO has been extended to solve MOPs, in which two or more MOPs can be tackled simultaneously in an effective way. However, these multiobjective multitasking optimization problems (MMTOPs) are more complicated and challenging, as their solution requires consideration of both convergence and diversity of different objectives within each task during the knowledge transfer process. Although many multiobjective evolutionary algorithms (MOEAs) [26]-[28] with very promising performance have been proposed recently, they are not sufficiently powerful for tackling MMTOPs due to the lack of knowledge transfer across tasks. Thus, embedding a multitasking engine into traditional MOEAs is a promising approach to transfer useful search experiences across tasks [19].

In recent years, a number of multiobjective multitasking evolutionary algorithms (MMTEAs) [29]-[41] have been designed based on multifactorial optimization, inspired by the multifactorial inheritance between organisms. In [29], Gupta *et al.* first proposed a multiobjective multifactorial evolutionary algorithm called MO-MFEA, which can solve several MOP tasks simultaneously using a single population. In this algorithm, exploration knowledge is extracted from different

This work was supported by the Shenzhen Science and Technology Program under Grants JCYJ20220531101411027 and JCYJ20190808174801673, in part by the Guangdong Regional Joint Foundation Key Project under Grant 2022B1515120076, and in part by the National Natural Science Foundation of China under Grant 62173236. Prof. Carlos A. Coello Coello was partially supported by CONACyT grant no. 2016-01-1920 (Investigacion en Fronteras de la Ciencia 2016). (*Corresponding author: Lijia Ma*)

Q.Z. Lin, Z.J. Wu, L.J. Ma, and J.Q. Li are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China.

M.G. Gong is with Key Laboratory of Collaborative Intelligence Systems, Ministry of Education, Xidian University, Xi'an, Shanxi Province 710071, China.

C.A. Coello Coello is with the Department of Computer Science, CINVESTAV-IPN (Evolutionary Computation Group), Mexico, D.F. 07300, MEXICO. He is also with the Faculty of Excellence at School of Engineering and Sciences, Tecnologico de Monterrey, Monterrey, N.L., Mexico.

tasks and implicitly transferred via assortative mating and vertical cultural transmission. Zhou *et al.* [30] proposed a modified MO-MFEA with adaptive knowledge transfer to obtain robust and improved performance. In this algorithm, the crossover operators as the carriers for knowledge transfer are adaptively adjusted based on the search experiences gained in the evolutionary process. Feng *et al.* [31] suggested an explicit knowledge transfer model via a denoising autoencoder so that different search mechanisms with unique biases can be incorporated to tackle various multitasks. Bali *et al*. [32] designed an enhanced MO-MFEA called MO-MFEA-II to alleviate the effect of negative transfer. In this algorithm, an online transfer parameter estimation scheme is designed to control the extent of knowledge transfer among tasks. Similarly, the probability of knowledge transfer is also adaptively adjusted in [33]. Moreover, to adapt the domains of different tasks for solving MMTOPs, Liang *et al.* [34] used a mapping matrix obtained by subspace alignment to transform the search spaces of different tasks, and Gao *et al.* [35] introduced a subspace distribution alignment method with decision variable transfer. To adaptively assign computational resources for various tasks, Yao *et al.* [36] directly embedded a dynamic resource allocation strategy [37] into MO-MFEA, and Wei *et al.* [38] designed a generalized resource allocation framework. To select promising solutions in knowledge transfer, Lin *et al.* suggested transferring only the neighbours of historical solutions with a positive transfer effect in EMT/ET [39] and further designed an incremental Bayes classifier to identify positive-transfer solutions in EMTIL [40], while Chen *et al.* [41] defined a transfer rank for each solution and designed a K-nearest neighbors classifier to further distinguish the solutions with the same rank for knowledge transfer. In some of our recent work [42], the found nondominated solutions from all tasks in each generation are used to train a discriminative reconstruction network model via backpropagation with gradient descent, which helps to transfer the solutions of the source task directly to the target task.

In most of the above mentioned MMTEAs, the Pareto-dominance method [43] is used for environmental selection. However, in our previous studies [44]-[46], the experiments have shown that the decomposition methods used in MOEAs perform significantly better than the Pareto-dominance method in most cases for solving complicated MOPs with variable linkages. To the best of our knowledge, only two MMTEAs have been designed based on the decomposition method. As a pioneering work, Yao *et al.* [36] designed a new MMTEA called MFEA/D-DRA, which is directly extended from MOEA/D-DRA [37]. This algorithm can dynamically assign different computational resources to various tasks but ignores the improvement of knowledge transfer methods. Recently, Wang *et al.* [47] suggested a new decomposition-based MMTEA, called MTEA/D-DN. This algorithm uses neighborhood as a bridge for knowledge transfer among different tasks, but its multitasking optimization performance is unsatisfactory, as shown in our experimental study. Following this research direction, this paper suggests an improved MMTEA with decomposition-based transfer selection, called MMTEA-DTS. In our algorithm, all tasks are first decomposed into a set of subproblems, and then the transfer potential of each solution can be properly quantified based on the performance improvement ratio of its associated subproblem. Only solutions with higher transfer potentials will be selected for a hybrid transfer evolution to share diverse search experiences across tasks. Compared to the above decomposition-based MMTEAs (MOEA/D-DRA and MTEA/D-DN), our algorithm improves both the selection and evolution of transfer solutions, which enhances the final multitasking optimization performance.

To summarize, the main contributions of this paper are the following:

(1) A novel decomposition-based transfer selection is designed to identify high-potential solutions for knowledge transfer. The transfer potential of each solution is quantified using the performance improvement ratio of its associated subproblem, and only solutions with higher transfer potentials will be selected as transfer solutions. This way, the negative transfer caused by randomly selected transfer solutions can be effectively alleviated.

(2) A hybrid transfer evolutionary mechanism is designed to run on the above high-potential solutions. The offspring generated by the parents in the same task will be transferred into other tasks, while the offspring generated by the parents from different tasks will be transferred into one random task. This way, knowledge transfer is more diversified in our algorithm, and the positive transfer effect is promoted.

(3) A new decomposition-based MMTEA (MMTEA-DTS) is proposed with the above decomposition-based transfer selection and hybrid transfer evolutionary mechanism. In this algorithm, solutions with a high transfer potential will be selected to transfer more diverse search experiences across tasks, which can better solve a variety of MMTOPs, as experimentally validated in our empirical studies.

To assess the efficacy of MMTEA-DTS, a number of empirical studies were conducted to solve three commonly used MMTOP test suites, i.e., CIHS-NILS and CPLX suggested in the CEC2017 [48] and the CEC2019 [30] competition on evolutionary multitasking optimization, respectively, ETMO suggested in the CEC2021 competition on evolutionary transfer multiobjective optimization [49], and one real-world problem suite for training neural networks [42]. Compared to five recently proposed MMTEAs (MO-MFEA-II [32], MO-MFEA-AKT [30], MO-MFEA-SADE [34], EMT/ET [39], and MTEA/D-DN [47]), our experimental results validate that MMTEA-DTS performs better in most cases in terms of both solution quality and convergence speed.

The remainder of this paper is organized as follows. Section II provides background information related to the paper. Then, Section III details the MMTEA-DTS algorithm. Experimental results are presented and analyzed in Section IV. Finally, Section V concludes this paper and discusses our future work.

## II. Background

This section first gives the formulation of MMTOPs in Section II. A and then introduces a state-of-the-art MOEA

(MOEA/D [10]) in Section II.B. Finally, two extensions of MOEA/D for solving MMTOPs are introduced, and our improvements over them are clarified in Section II.C.

### A. Definition of MMTOPs

Generally, an MMTOP with $K$ MOPs can be mathematically formulated as follows:

$$\begin{aligned} &\min\ \mathbf{Q}(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_K) \\ &\quad = (\mathbf{F}_1(\mathbf{x}_1), \mathbf{F}_2(\mathbf{x}_2), ..., \mathbf{F}_K(\mathbf{x}_K)), \\ &\text{s.t. } \mathbf{x}_i \in \mathbf{\Omega}_i,\ i \in \{1,2,...,K\} \end{aligned} \tag{1}$$

where $\mathbf{x}_i = \{x_i^1, x_i^2, \ldots, x_i^{D_i}\}$ is the $D_i$-dimensional decision variable vector, $\mathbf{\Omega}_i$ denotes the search space of $\mathbf{x}_i$, and $\mathbf{F}_i$ represents the $i$-th MOP to be optimized, as follows:

$$\min\ \mathbf{F}_i(\mathbf{x}_i) = (f_i^1(\mathbf{x}_i), f_i^2(\mathbf{x}_i), ..., f_i^m(\mathbf{x}_i)), \tag{2}$$

where $m$ is the number of objectives and $f_i^j(\mathbf{x}_i)$ ($j = 1, \ldots, m$) is the $j$-th objective function to be optimized in the $i$-th MOP. Inspired by the way that the human brain solves new tasks by leveraging previously acquired knowledge and experiences instead of starting from scratch, MMTOPs can be effectively tackled by identifying the correlations between different MOPs and exploiting the potential parallelism of evolutionary algorithms [48].

### B. The MOEA/D Framework for Tackling MOPs

MOEA/D [10] uses a decomposition method for converting an MOP into a set of single-objective optimization problems and then solves them simultaneously. Here, a single-objective optimization subproblem decomposed from an MOP ($\mathbf{F}(\mathbf{x})$) by the Tchebycheff method [44] is defined as follows:

$$\begin{aligned} &g^{tch}(\mathbf{x} \mid \boldsymbol{\lambda}, \mathbf{z}) = \min\left[\max_{1 \le j \le m}\left\{\lambda^j \mid f^j(\mathbf{x}) - z^j \mid\right\}\right], \\ &\text{s.t. } \mathbf{x} \in \mathbf{\Omega} \end{aligned} \tag{3}$$

where $\boldsymbol{\lambda} = \{\lambda^1, \lambda^2, ..., \lambda^m\}$ is a weight vector, i.e., $\lambda^j \ge 0$ for all $j = 1, \ldots, m$, and $\sum_{j=1}^m \lambda^j = 1$. $f^j(\mathbf{x})$ is the $j$-th objective function, and $\mathbf{z} = \{z^1, z^2, ..., z^m\}$ is an ideal point to record the minimal value of each objective function.

To clearly introduce the running of MOEA/D, its framework is given in **Algorithm 1** with five inputs: 1) $N$ (the number of subproblems); 2) $\boldsymbol{\lambda} = \{\lambda^1, \lambda^2, ..., \lambda^m\}$ (a set of $N$ weight vectors); 3) $T$ (the number of weight vectors in the neighborhood); 4) $\delta$ (the probability that parent solutions are selected from the neighborhood); and 5) $n_r$ (the maximum number of solutions replaced by each offspring). The output of MOEA/D is an approximation set ($\mathbf{x}^1, \ldots, \mathbf{x}^N$) to the Pareto-optimal set (PS). From the start, in lines 1, 9, 13, and 15 of **Algorithm 1**, MOEA/D can be divided into four functional components:

*(1) Initialization in lines 2-6*: First, the Euclidean distances between any pair of weight vectors are computed, and then the $T$ closest weight vectors to each weight vector are found and saved into a set **B** in lines 2-4. Second, an initial population is generated uniformly and randomly in the search space, and the current minimum value of each objective composes the ideal point **z**. This process can be denoted as a function: (**B**, **x**, **z**) = **Initialization**($N$, **λ**, $T$).

*(2) Evolutionary process in lines 10-12:* For each subproblem $i = 1, 2, \ldots, N$, $rand_1$ is a real number randomly generated in [0, 1]. If it is smaller than $\delta$, the range of mating selection and replacement **P** is set as $\mathbf{B}^i$; otherwise, **P** is set as $\{1, \ldots, N\}$ in line 10. After that, two indexes $r_1$ and $r_2$ are randomly selected from **P** in line 11. Then, differential evolution (DE) and polynomial-based mutation (PM) are used to generate the offspring **y** by using $\mathbf{x}^i$, $\mathbf{x}^{r_1}$, $\mathbf{x}^{r_2}$ in line 12. In detail, each element $y_k'$ ($k = 1, ..., n$, and $n$ is the number of decision variables in $\mathbf{x}^i$) of the preliminary offspring $\mathbf{y}'$ is generated by DE as follows:

$$y_k' = \begin{cases} x_k^i + F \times \left(x_k^{r_1} - x_k^{r_2}\right) & \text{if } rand_2 < CR \\ x_k^i & \text{otherwise} \end{cases}, \tag{4}$$

where $rand_2$ is a real number randomly generated in [0, 1], while $CR$ and $F$ are two control parameters in DE. Then, PM generates $\mathbf{y} = (y_1, ..., y_n)$ from $\mathbf{y}'$ in the following way:

$$y_k = \begin{cases} y_k' + \sigma_k \times (b_k - a_k) & \text{if } rand_3 < p_m \\ y_k' & \text{otherwise} \end{cases}, \tag{5}$$

with

$$\sigma_k = \begin{cases} (2 \times rand_4)^{\frac{1}{\eta+1}} - 1 & \text{if } rand_4 < 0.5 \\ 1 - (2 - 2 \times rand_4)^{\frac{1}{\eta+1}} & \text{otherwise} \end{cases}, \tag{6}$$

where $rand_3$ and $rand_4$ are two real numbers randomly generated in [0, 1]. $a_k$ and $b_k$ ($k = 1, 2, \ldots, n$) are the lower and upper bounds of the $k$-th decision variable, respectively. The distribution index $\eta$ and the mutation rate $p_m$ are two control parameters in PM. The above process of generating offspring can be denoted as a function: (**y**, **P**) = **Evolution** ($\delta$, $i$, **x**, **B**).

*(3) Update the ideal point in line 14:* For each $j = 1, \ldots, m$, if $z^j > f^j(\mathbf{y})$, then set $z^j = f^j(\mathbf{y})$. This process can be denoted as a function: **z** = **Update-Ideal-Point**(**y**, **z**).

*(4) Update the population in lines 16-23:* First, the number of solution replacements $c$ is initialized as 0, and an index $j$ is randomly selected from **P**. Then, the aggregated function values of $\mathbf{x}^j$ and **y** are calculated by (3). When $g^{tch}(\mathbf{y} \mid \lambda^j, \mathbf{z})$

**Algorithm 1: MOEA/D** ($N$, **λ**, $T$, $\delta$, $n_r$)

```
1: //(B, x, z) = Initialization(N, λ, T)
2: for i = 1 : N do
3:    B^i = {i_1,..., i_T} where λ^{i_1},..., λ^{i_T} are the T closest weight vectors of λ^i;
4: end
5: generate an initial population x^1,..., x^N and evaluate the objectives;
6: initialize z = {z^1, z^2, ..., z^m} by setting z^j = min_{1≤i≤N} f^j(x^i);
7: while the termination condition is not met do
8:    for i = 1 : N do
9:       //(y, P) = Evolution(δ, i, x, B);
10:      P = rand_1 < δ? B^i : {1,..., N};
11:      r_1, r_2 ← randomly select two indexes from P;
12:      sequentially run DE and PM on solutions x^i, x^{r_1}, x^{r_2} to get y;
13:      // z =Update-Ideal-Point(y, z);
14:      for each j = 1 ,..., m, if z^j > f^j(y), then set z^j = f^j(y);
15:      // x = Update-Population(P, y, x, n_r);
16:      set c = 0;
17:      while c < n_r and P ≠ Ø do
18:         j ← randomly select an index j from P;
19:         if g^tch(y | λ^j, z) < g^tch(x^j | λ^j, z)
20:            x^j = y;
21:            c = c + 1 and remove j from P;
22:         end if
23:      end while
24: end while
Output: an approximation set (x^1,..., x^N) to the PS;
```

$< g^{tch}\left(\mathbf{x}^j \mid \lambda^j, \mathbf{z}\right)$, $\mathbf{x}^j$ is replaced by $\mathbf{y}$, $c$ is incremented by 1, and index $j$ is removed from $\mathbf{P}$. Please note that the maximum number of solution replacements is not more than $n_r$. This process can be denoted as a function: $\mathbf{x}$ = **Update-Population**($\mathbf{P}$, $\mathbf{y}$, $\mathbf{x}$, $n_r$).

### C. Extensions of MOEA/D for MMTOPs

In a pioneering work, Yao *et al*. [36] proposed the first decomposition-based MMTEA, called MFEA/D-DRA, which is an extension of MOEA/D-DRA [37]. However, MFEA/D-DRA adopts a simple knowledge transfer method that cannot fully leverage the useful search experiences among different tasks. To further enhance MFEA/D-DRA, Wang *et al.* designed MTEA/D-DN [47]. In this algorithm, the internal and external neighborhoods of each subproblem are employed as a bridge to explore the correlations and potentially useful information among different tasks to further improve the efficiency of tackling various tasks. Compared to most MMTEAs based on the Pareto-dominance method [43], the aforementioned decomposition-based MMTEAs show more promising performance in solving complex MMTOPs with variable linkages.

Inspired by the two aforementioned studies, this paper also employs a decomposition-based approach to simultaneously solve all tasks as single-objective optimization subproblems. However, our algorithm stands out by improving the selection and evolution of transfer solutions. To facilitate knowledge transfer and speed up the convergence of the associated subproblems, our algorithm records the performance improvement ratio of each solution on its associated subproblem as an indicator of its transfer potential. Solutions with a higher transfer potential are then selected as potential carriers for knowledge transfer, which are likely to provide positive transfer for other related subproblems. Moreover, to further explore the potential synergy among various tasks, our algorithm incorporates a hybrid transfer evolutionary method that operates in two different ways: (1) transferring offspring evolved in the same task to another task and (2) transferring offspring evolved from different tasks into one task. This approach diversifies the information exchange in knowledge transfer and facilitates the discovery of new solutions that can benefit multiple tasks. This way, our algorithm delivers superior performance for solving different MMTOPs.

## III. Our Proposed Algorithm

In this section, the framework of MMTEA-DTS is first introduced in Section III. A, which is extended from MOEA/D introduced in Section II.B. As clarified in Section II. C, MMTEA-DTS has two main components different from the two existing decomposition-based MMTEA, i.e., a hybrid transfer evolution and a decomposition-based transfer selection, which are respectively introduced in Section III. B and Section III.C.

### A. The Framework of MMTEA-DTS

To clarify the running of MMTEA-DTS, its framework is provided in **Algorithm 2** based on MOEA/D [10] with four inputs: 1) $K$ (the number of tasks); 2) $Tp$ (the probability of knowledge transfer); 3) $a$ (a parameter to control the size of the potential transfer solution set); and 4) **TV** (transfer potentials). The output of MMTEA-DTS is an approximation set ($\mathbf{x}_1$, …, $\mathbf{x}_K$) to the PSs of all tasks. Here, the functional components of MOEA/D introduced in Section II are reused in **Algorithm 2**. Specifically, each task $i$ from 1, …, $K$ maintains the following parameters: 1) $\mathbf{B}_i$ (a set of neighborhoods); 2) $\mathbf{z}_i$ (an ideal point); 3) $\boldsymbol{\lambda}_i$ (a set of weight vectors); and 4) $\mathbf{x}_i$ (a set of solutions).

**Algorithm 2: MMTEA-DTS ($K$, $Tp$, $a$, TV)**

```
1:  for i = 1 : K do
2:     (B_i, x_i, z_i) = Initialization(N, λ_i, T);//function in Algorithm 1
3:     for each j = 1,…, N, set TV_i^j = 1.0;
4:  end for
5:  set FES = N × K;
6:  E= {(1, 1),…, (1, N),…, (K, 1),…,(K, N)};
7:  while FES < MaxFES
8:     for each (i, j) in E
9:        if rand < tp
10:          (y, P) = HTE (x, B, i, j);//Algorithm 3
11:          target ← randomly select an index different from i in{1, …, K};
12:       else
13:          (y, P) = Evolution(δ, j, x_i, B_i);//function in Algorithm 1
14:          target = i;
15:       end if
16:       evaluate y for the target task and FES = FES + 1;
17:       z_target = Update-Ideal-Point(y, z_target);//function in Algorithm 1
18:       x_target=Update-Population(P, y, x_target, n_r);//function in Algorithm 1
19:    end for
20:    E = DTS (TV, a);//Algorithm 4
21: end while
Output: an approximation set (x_1, …, x_K) to the PSs of all tasks
```

MMTEA-DTS starts from an initialization process in lines 1-6. For each task $i$ from 1, …, $K$ in line 2, its neighborhood set $\mathbf{B}_i$, population $\mathbf{x}_i$, and ideal point $\mathbf{z}_i$ are initialized by the function **Initialization**($N$, $\boldsymbol{\lambda}_i$, $T$) introduced in **Algorithm 1** (here, $N$ is the population size and $T$ is the neighborhood size for each task). In particular, each initial solution in $\mathbf{x}_i$ is randomly sampled in [0, 1] to facilitate knowledge transfer across tasks, and the vector length of each solution is set the same as the maximum number of decision variables among all tasks. In line 3, for each $j$-th solution ($j$ = 1, …, $N$) in task $i$, its transfer potential $TV_i^j$ is initialized as 1.0. In the above initialization process, the number of function evaluations is set to $N \times K$ in line 5, where $\times$ means the multiplication operator. As seen in line 6, $\mathbf{E}$ is initialized as a set of $N \times K$ tuples that are used to track high-potential solutions for knowledge transfer, where the first index is the associated task and the second index is the associated subproblem.

While the current number of maximum times has not been reached, the evolutionary process in lines 7-21 is run. Each high-potential solution in $\mathbf{E}$ is used to generate an offspring $\mathbf{y}$ for the current task $i$ or another task, as controlled by a transfer probability $Tp$. When $rand$ is smaller than $Tp$ in line 9, $\mathbf{y}$ is evolved by our hybrid transfer evolution (**Algorithm 3**) in line 10 as a transfer solution into other tasks, and the index of the target task (*target*) different from $i$ is randomly selected in line 10. Otherwise, the same evolution in MOEA/D is run in line 13 to optimize its associated task, and the *target* is set to $i$ in line 14. Then, the offspring $\mathbf{y}$ is evaluated for the target task, and the counter of function evaluations is incremented by 1 in line 16. In lines 17-18, the evaluated solution $\mathbf{y}$ is used to

**Algorithm 3: HTE (x, B, *i*, *j*)//Hybrid Transfer Evolution**

1: **if** *rand* < 0.5
2: (**y**, **P**) = **Evolution**($\delta$, *j*, $\mathbf{x}_i$, $\mathbf{B}_i$);
3: **else**
4: $i_1$, $i_2$ ← randomly select two indexes from {1, …, *K*};
5: $j_1$, $j_2$ ← randomly select two indexes from {1, …, *N*};
6: sequentially run SBX and PM on solutions $\mathbf{x}_{i_1}^{j_1}$, $\mathbf{x}_{i_2}^{j_2}$ to get **y**;
7: **P** = {1, …, *N*};
8: **end if**

**Output**: the transfer child **y**, the range of population replacement **P**;

**Algorithm 4: DTS (TV, *a*)//Decomposition-based Transfer Selection**

1: set **S** = {(1, 1),…, (1, *N*),…, (*K*, 1),…,(*K*, *N*)};
2: **if** there is little improvement in last generation
3: **E** = **S**;
4: **else**
5: compute the performance improvement ratio of each subproblem in (9);
6: **for** *i* = 1 : *K* **do**
7: for each *j* = 1 ,…, *N*, update $TV_i^j$ by (10);
8: **end for**
9: sort **S** in a descending order based on their transfer potentials by (10);
10: save the first $\lfloor \frac{K\times N}{a} \rfloor$ tuples into **E**
11: **end if**

**Output**: the potential transfer solution set **E**;

update the ideal point and solutions of the target task. After all the selected solutions in **E** are evolved or transferred, the potential transfer solution set is updated by the decomposition-based transfer selection (**Algorithm 4**) in line 20.

### *B. A Hybrid Transfer Evolutionary Mechanism*

To clarify the behavior of our hybrid transfer evolutionary mechanism, its pseudocode is provided in **Algorithm 3** with four inputs: 1) **x** (all *K* sets of solutions); 2) **B** (all *K* sets of neighborhoods); 3) *i* (the index of the source task); and 4) *j* (the index of the subproblem). It is noted that the indexes *i* and *j* are used to track the transfer solution in **E**. Two options of knowledge transfer are randomly selected here: (1) the transfer solution is evolved in line 2 with the solutions in source tasks by **Evolution**($\delta$, *j*, $\mathbf{x}_i$, $\mathbf{B}_i$), as introduced in **Algorithm 1**, which can share the search experiences of the source task with other target tasks; and (2) the transfer solution is evolved in lines 4-6 with the solutions of other tasks by sequentially running simulated binary crossover (SBX) and PM, which further diversifies the sharing of search experiences across all tasks.

Here, in line 6, the SBX operator is introduced, which will generate a temporary solution $\mathbf{y}'$, as follows:

$$\mathbf{y}' = \begin{cases} \frac{1}{2}\cdot\left[(1-\beta)\cdot\mathbf{x}_{i_1}^{j_1} + (1+\beta)\cdot\mathbf{x}_{i_2}^{j_2}\right] & \text{if } rand_5 < 0.5 \\ \frac{1}{2}\cdot\left[(1+\beta)\cdot\mathbf{x}_{i_1}^{j_1} + (1-\beta)\cdot\mathbf{x}_{i_2}^{j_2}\right] & \text{otherwise} \end{cases}, \quad (7)$$

where $rand_5$ is a real number randomly generated in [0, 1], $i_1$ and $i_2$ are randomly selected from {1, …, *K*}, and $j_1$ and $j_2$ are randomly selected from {1, …, *N*}. An exponential probability distribution $\beta$ is defined by

$$\beta(u) = \begin{cases} (2u)^{1/(\eta_c+1)}, & \text{if } u \le \frac{1}{2} \\ [2(1-u)]^{-1/(\eta_c+1)}, & \text{otherwise} \end{cases}, \quad (8)$$

where *u* is a random number generated in the range [0, 1] and $\eta_c$ is the crossover distribution index. Finally, PM is run on $\mathbf{y}'$ to generate **y** = ($y_1$, …, $y_n$) by (5).

### *C. Decomposition-based Transfer Selection*

To clarify the behavior of our decomposition-based transfer selection mechanism, its pseudocode is provided in **Algorithm 4** with two inputs: 1) **TV** (transfer potentials of all solutions for *K* tasks) and 2) *a* (a parameter to control the size of the potential transfer solution set). First, in line 1, the set **S** is used to locate all the solutions where the first and second elements indicate the solution's index and its associated task, respectively. If there is little improvement from the last generation in line 2, all the solutions should be selected for potential knowledge transfer and evolution. Thus, the potential transfer solution set **E** is reinitialized as **S** in line 3. Otherwise, in line 5, the performance improvement ratio $\Delta_i^j$ of the *j*-th subproblem in the *i*-th task is computed based on (3) as follows:

$$\Delta_i^j = \frac{g^{tch}\left(\mathbf{x}_{i,old}^j \mid \boldsymbol{\lambda}_i^j, \mathbf{z}_i\right) - g^{tch}\left(\mathbf{x}_{i,new}^j \mid \boldsymbol{\lambda}_i^j, \mathbf{z}_i\right)}{g^{tch}\left(\mathbf{x}_{i,old}^j \mid \boldsymbol{\lambda}_i^j, \mathbf{z}_i\right)}, \quad (9)$$

where $\mathbf{x}_{i,old}^j$ and $\mathbf{x}_{i,new}^j$ are the old solution in the last generation and the new solution in the current generation associated with the subproblem, respectively. Then, in lines 6-8, the transfer potential of each solution associated with the *j*-th subproblem in the *i*-th task is updated by

$$TV_i^j = \begin{cases} 1 & \text{if } \Delta_i^j > \Delta_M \\ \text{Max}(TV_i^j - \Delta_M + \Delta_i^j,\ 0) & \text{otherwise} \end{cases}, \quad (10)$$

where $\Delta_i^j$ is the performance improvement ratio of its associated subproblem in (9) and $\Delta_M$ is their median improvement ratio collected from the last generation. After that, all the solutions are sorted in descending order based on the transfer potentials in line 9, and the top $\lfloor \frac{K\times N}{a} \rfloor$ tuples in **S** are added into **E** as the potential transfer solutions for the next generation in line 10, where $\lfloor x \rfloor$ is a floor function retuning the maximum integer no larger than $x$. Note that *a* is a preset parameter to control the size of **E,** and its impact will be experimentally studied in Section IV.E. This way, the solutions with performance improvement ratios larger than the median ratio are preferred for undertaking both evolution and knowledge transfer among the tasks.

## IV. Experimental Studies

In this section, MMTEA-DTS is compared to five recently proposed MMTEAs on three well-known test suites to assess its performance. Then, more experiments and discussions about the effectiveness of our hybrid transfer evolutionary mechanism and our decomposition-based transfer selection mechanism are presented. Finally, our performance on a real-world application is also studied.

### *A. The Three Used Test Suites*

The first is the classical multitasking multiobjective test suite proposed in the IEEE CEC2017 competition on evolutionary multitasking optimization [48]. These test problems have three degrees of intersection, i.e., complete intersection

(CI), partial intersection (PI), and no intersection (NI). Additionally, they have three degrees of similarity in the fitness landscapes, i.e., high similarity (HS), medium similarity (MS), and low similarity (LS). Thus, there are nine types of test problems, i.e., CIHS, CIMS, CILS, PIHS, PIMS, PILS, NIHS, NIMS, and NILS, which include two MOPs in one task. Their detailed definitions can be found in Table A. I of the Supplementary Material due to page limitations.

The second is the complex multitasking multiobjective optimization test suite CEC2019-CPLX proposed in the IEEE CEC2019 competition on evolutionary multitasking optimization, which contains ten MMTOPs constructed from F1-F9 of the LZ09 benchmarks [10]. Their detailed definitions can be found in Tables A.II-A. III of the Supplementary Material due to page limitations.

The third are the benchmark problems proposed in the CEC2021 competition on evolutionary transfer multiobjective optimization, where forty benchmark functions are designed, covering diverse types and properties of multitasks [49]. Only ETMOF1-ETMOF8 belonging to MMTOPs are used in this paper. Their detailed definitions can be found in Tables A. IV-A. VI of the Supplementary Material due to page limitations.

### *B. The Compared Algorithms and Parameters Settings*

In this study, MMTEA-DTS[1] is compared to five competitive MMTEAs, i.e., four Pareto-dominance-based MMTEAs (MO-MFEA-II [32], MO-MFEA-AKT [30], MO-MFEA-SADE [34], and EMT/ET [39]) and one recent decomposition-based MMTEA (MTEA/D-DN [47]). MFEA/D-DRA [36] is not included for comparison, as it has been found to be inferior than MTEA/D-DN. To ensure a fair comparison, all the compared MMTEAs were run using the original source codes provided by their authors with their recommended parameters settings. Due to page limitations, Table A. VII of the Supplementary Material summarizes their parameters settings, where these parameters should be set by the user and some randomly generated values are used in the mutation, crossover and knowledge transfer operators, such as (4)-(8).

The population size for each task is set as $N = 100$, and the maximum number of function evaluations (FEs) is set as $FEs = 50000$ in each task for all the compared MMTEAs. A relatively small number of FEs is more practical for real-world MMTOPs due to computational resource limitations. Each algorithm is run 30 times on the test problems independently to obtain the mean values and standard deviations. It is noted that the crossover operators for knowledge transfer across different tasks include DE and SBX. When generating offspring without knowledge transfer in each task, the DE operator is used in all the compared MMTEAs to ensure a fair comparison, as suggested in [47]. The weight vectors used in MTEA/D-DN and MMTEA-DTS are generated using the systematic method described in [10]. Other parameters settings in MMTEA-DTS are summarized at the end of Table A. VII in the Supplementary Material.

[1]The source codes of MMTEA-DTS and of all the test problems adopted are available at https://github.com/Linqiuzhen/DTS.

### *C. Performance Measure*

The goal of solving MOPs is to find a solution set with an even distribution that can approximate the true Pareto-optimal front (PF). The inverted generational distance (IGD) [50], [51] is a widely used performance indicator that can assess both the diversity and convergence of a solution set, and is calculated using:

$$\mathrm{IGD}\left(\mathbf{S},\mathbf{PF}^*\right)=\frac{1}{\left|\mathbf{PF}^*\right|}\sqrt{\sum_{\mathbf{x}\in\mathbf{PF}^*}\left(\min_{\mathbf{y}\in\mathbf{S}} dist(\mathbf{x},\mathbf{y})\right)^2}, \quad (11)$$

where $\mathbf{S}$ represents a set of solutions obtained by an MMTEA, $\mathbf{PF}^*$ represents a subset of the true PF, and $dist(\mathbf{x},\mathbf{y})$ is the Euclidean distance between two solutions $\mathbf{x}$ in $\mathbf{PF}^*$ and $\mathbf{y}$ in $\mathbf{S}$. IGD can evaluate the convergence and diversity performance by calculating the average value of the minimum distance from each reference point on the true PF to the solution set obtained by an MMTEA. Thus, a smaller IGD value indicates a better overall performance of the MMTEA in terms of convergence and diversity. In our study, the number of solutions sampled in $\mathbf{PF}^*$ ($|\mathbf{PF}^*|$) is set to 5000 and 10000 for biobjective problems and tri-objective problems, respectively.

Based on the above IGD values, a comprehensive indicator (i.e., the mean standard score (MSS) [48]), is further used to evaluate the overall multitasking optimization performance of an MMTEA. Suppose $N$ MMTEAs are compared for solving an MMTOP with $K$ optimization tasks $T_1$, $T_2$,..., $T_K$, each of which is executed independently with $L$ runs. Then, for each MMTEA, its MSS value on the test problem is computed by

$$MSS=\frac{1}{L\cdot K}\sum_{k=1}^{K}\sum_{l=1}^{L}\frac{I_k^l-\mu_k}{\delta_k}, \quad (12)$$

where $I_k^l$ denotes the IGD value obtained by an MMTEA in the $l$th run on task $T_k$ ($k = 1, 2,\ldots, K$), while $\mu_k$ and $\delta_k$ are the mean value and standard deviation of the IGD results obtained by each compared MMTEA in all runs, respectively. A smaller value of MSS indicates a better overall performance of an MMTEA on the minimized MMTOP [48].

### *D. Statistical Results and Discussion*

Tables I-III provide the IGD and MSS results obtained by all the compared MMTEAs for tackling three test suites of MMTOPs from the MTO computation in CEC2017, CEC2019, and CEC2021, respectively. In all the tables, the best mean value for each test instance is highlighted by boldface and gray background. Moreover, the Wilcoxon rank-sum test is conducted on the IGD values with a significance level of 5%, and the symbols '+', '-', and '≈' behind each mean IGD value indicate that MMTEA-DTS performs better than, worse than, and similarly to the corresponding MMTEA, respectively, based on the IGD indicator. The symbols '+' and '-' behind each MSS value indicate that MMTEA-DTS performs better than and worse than the corresponding MMTEA based on the MSS indicator, respectively. In the second to last row of each table, the values of "≈/+/-" summarize the numbers of test problems on which MMTEA-DTS performs significantly better than, worse than, and similarly to the corresponding MMTEA based on the IGD or MSS indicator, respectively. Moreover, to comprehensively show the performance of each

**TABLE I**
THE MEAN VALUES AND STANDARD DEVIATIONS OF IGD AND MSS RESULTS OBTAINED BY THE COMPARED MMTEAS IN CEC2017

| CEC2017 | MO-MFEA-II | | MO-MFEA-AKT | | MO-MFEA-SADE | | EMT/ET | | MTEA/D-DN | | MMTEA-DTS | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | IGD | MSS | IGD | MSS | IGD | MSS | IGD | MSS | IGD | MSS | IGD | MSS |
| CIHS1 | 2.61E-03 + (7.14E-04) | 1.29 + | 1.08E-03 + (3.20E-04) | 0.40 + | **1.67E-04 - (9.44E-06)** | -0.361 + | 4.43E-03 + (1.08E-03) | 2.18 + | 2.63E-04 + (5.89E-05) | -0.21 + | 1.76E-04 (5.08E-06) | **-0.363** |
| CIHS2 | 8.14E-03 + (1.36E-03) | | 4.56E-03 + (7.57E-04) | | 4.69E-04 + (2.87E-05) | | 1.11E-02 + (1.71E-03) | | 1.47E-03 + (2.85E-04) | | **4.42E-04 (7.70E-05)** | |
| CIMS1 | 1.10E-01 + (9.15E-02) | 1.30 + | 6.42E-02 + (8.62E-02) | 0.60 + | 3.30E-02 + (6.81E-02) | 0.29 + | 9.55E-02 + (8.40E-02) | 1.26 + | 1.49E-04 + (2.20E-05) | -0.051 + | **1.41E-04 (8.60E-08)** | **-0.052** |
| CIMS2 | 1.64E-02 + (1.67E-02) | | 6.30E-03 + (6.08E-03) | | 3.65E-03 + (5.06E-03) | | 1.77E-02 + (1.67E-02) | | 1.90E-04 + (2.88E-05) | | **1.74E-04 (1.53E-07)** | |
| CILS1 | 1.49E-03 + (3.42E-04) | 0.02 + | 7.49E-04 + (1.82E-04) | 0.00 + | **1.75E-04 - (7.32E-06)** | -0.02 + | 2.13E-03 + (4.02E-04) | 0.04 + | 1.25E-02 + (6.59E-02) | 0.45 + | 1.82E-04 (8.13E-06) | **-0.03** |
| CILS2 | 2.02E-04 + (8.56E-06) | | 1.94E-04 + (9.94E-06) | | 1.66E-04 + (6.46E-06) | | 2.27E-04 + (1.58E-05) | | 6.24E-04 + (2.33E-03) | | **1.62E-04 (1.52E-06)** | |
| PIHS1 | 1.66E-02 + (6.21E-03) | 1.13 + | 2.33E-02 + (7.04E-03) | 1.73 + | **1.81E-04 - (7.19E-06)** | **-0.34 -** | 2.16E-02 + (8.56E-03) | 1.64 + | 6.11E-04 ≈ (1.61E-04) | -0.26 + | 5.50E-04 (1.63E-04) | -0.32 |
| PIHS2 | 5.57E-01 + (1.52E-01) | | 7.89E-01 + (1.45E-01) | | **1.48E-03 - (3.37E-03)** | | 7.75E-01 + (1.68E-01) | | 5.33E-02 + (2.90E-02) | | 3.69E-03 (5.41E-03) | |
| PIMS1 | 4.97E-03 ≈ (2.41E-03) | -0.11 - | 6.44E-03 ≈ (1.91E-03) | 0.18 - | **1.39E-03 - (8.10E-04)** | **-1.18 -** | 4.03E-03 - (2.07E-03) | 0.01 - | 2.06E-03 - (8.12E-04) | -0.30 - | 1.45E-02 (1.70E-02) | 0.58 |
| PIMS2 | 1.21E+01 ≈ (4.20E+00) | | 1.44E+01 + (3.77E+00) | | **2.55E+00 - (1.85E+00)** | | 1.41E+01 + (3.46E+00) | | 1.19E+01 ≈ (5.26E+00) | | 1.33E+01 (3.76E+00) | |
| PILS1 | 7.48E-04 + (1.41E-04) | 0.50 + | 6.56E-04 + (1.82E-04) | 0.30 + | **1.77E-04 - (4.03E-05)** | 0.20 + | 6.69E-04 + (2.00E-04) | 0.35 + | 3.49E-04 - (1.82E-04) | 0.00 + | 3.86E-04 (1.46E-04) | **-0.41** |
| PILS2 | 4.63E-02 + (8.84E-03) | | 4.18E-02 + (9.93E-03) | | 2.25E-01 + (9.15E-02) | | 4.74E-02 + (1.09E-02) | | 1.06E-01 + (1.83E-01) | | **9.37E-04 (3.50E-04)** | |
| NIHS1 | 2.04E+00 + (2.01E-01) | 1.38 + | 1.73E+00 + (6.19E-02) | 0.41 + | 1.47E+00 + (6.85E-03) | -0.31 + | 2.25E+00 + (2.21E-01) | 2.06 + | 1.50E+00 + (1.44E-02) | -0.24 + | **1.46E+00 (2.00E-02)** | **-0.33** |
| NIHS2 | 3.35E-03 + (1.16E-03) | | 1.47E-03 + (4.25E-04) | | **1.84E-04 - (7.28E-06)** | | 4.73E-03 + (1.30E-03) | | 3.30E-04 + (6.88E-05) | | 2.12E-04 (1.72E-05) | |
| NIMS1 | 3.62E-01 + (3.19E-01) | 1.01 + | 2.10E-01 + (1.65E-01) | 0.14 + | 2.39E-01 + (2.24E-01) | 0.20 + | 3.64E-01 + (2.89E-01) | 0.77 + | 1.80E-01 ≈ (2.35E-01) | 0.05 + | **1.02E-01 (1.23E-02)** | **-0.11** |
| NIMS2 | 6.91E-02 + (9.96E-02) | | 3.28E-03 + (8.11E-03) | | 4.09E-03 + (8.65E-03) | | 4.04E-02 + (7.97E-02) | | 9.65E-04 + (1.24E-03) | | **1.89E-04 (3.81E-05)** | |
| NILS1 | 1.06E-03 + (2.00E-04) | 1.11 + | 9.81E-04 + (1.23E-04) | 0.88 + | **8.20E-04 ≈ (2.82E-05)** | **-0.77 -** | 8.21E-04 ≈ (4.32E-05) | 0.01 + | 8.96E-04 ≈ (1.68E-04) | -0.15 + | 8.36E-04 (2.12E-05) | -0.65 |
| NILS2 | 6.44E-01 + (5.42E-04) | | 6.44E-01 + (7.12E-04) | | 6.42E-01 ≈ (5.45E-04) | | 6.43E-01 + (2.64E-04) | | 6.42E-01 + (5.20E-04) | | **6.42E-01 (1.32E-04)** | |
| ≈/+/- | 2/16/0 | 0/8/1 | 1/17/0 | 0/8/1 | 2/8/8 | 0/6/3 | 1/16/1 | 0/8/1 | 4/12/2 | 0/8/1 | / | / |
| Rank | 4.96 | | 4.35 | | 2.41 | | 4.73 | | 2.90 | | **1.65** | |

MMTEA in each test suite, the average performance ranks are provided in the last row of each table, which are obtained by using Friedman's test [52] on the IGD results.

*1) Performance Comparisons on CIHS-NILS*

Table I provides the mean values and standard deviations of the IGD and MSS results obtained by all the compared MMTEAs in 30 independent runs for tackling nine test MMTOPs in CEC2017. From the IGD results in Table I, MMTEA-DTS performs significantly better than MO-MFEA-II, MO-MFEA-AKT, EMT/ET, and MTEA/D-DN on 16, 17, 16, and 12 cases out of a total of 18 tasks, respectively. It should be noted that our MMTEA-DTS can perform well on MMTOPs with varying degrees of similarity. Although MO-MFEA-II adopts a multi-Gaussian distribution model to estimate the similarity of two solution sets, its estimation is not always accurate, which may induce a negative transfer effect. This is mainly because the number of samples is relatively small in MO-MFEA-II for building the Gaussian model in a high-dimensional decision space. Although MO-MFEA-AKT uses an adaptive multiple-operator strategy to enhance the effect of knowledge transfer, its performance is still not satisfactory due to the randomly selected transfer parents. Similarly, MTEA/D-DN does not perform as well mainly due to this random mating strategy. In EMT/ET, only the neighbours of transfer solutions showing a positive effect before are selected for knowledge transfer, but it still does not work well on this test suite. In our algorithm, as the qualities of transfer solutions are estimated by the transfer potentials in (10), MMTEA-DTS can always select the transfer solutions with promising evolutionary directions, which are further diversified by our hybrid transfer evolution. Thus, MMTEA-DTS can perform better than the above mentioned MMTEAs. Due to the subspace alignment strategy used, MO-MFEA-SADE performs comparably with MMTEA-DTS on this test suite, but MMTEA-DTS still has some advantages. For example, in the IGD results of CIMS1 and CIMS2, where CIMS1 is a harder task with multimodality and both tasks have nonseparable decision variables, MMTEA-DTS can obtain a well-converged solution set for CIMS1 by transferring the search experiences from the relatively easy task CIMS2, while MO-MFEA-SADE may become trapped occasionally in the local optima of CIMS1. The performance ranks based on the IGD results (shown in the last row of Table I) also indicate that MMTEA-DTS performs best on this test suite among all the competitors, as it has the smallest rank (1.65).

Moreover, by observing another comprehensive indicator (MSS) in Table I, when compared with MO-MFEA-II, MO-MFEA-AKT, MO-MFEA-SADE, EMT/ET, and MTEA/D-DN, MMTEA-DTS performs significantly better on 8, 8, 6, 8, and 8 cases out of a total of 9 MMTOPs, while it is only outperformed on 1, 1, 3, 1 and 1 cases, respectively. Thus, these MSS results further confirm the advantages of MMTEA-DTS on this test suite.

*2) Performance Comparisons on CPLX1-CPLX10*

**TABLE II**
THE MEAN VALUES AND STANDARD DEVIATIONS OF IGD AND MSS RESULTS OBTAINED BY THE COMPARED MMTEAS IN CEC2019

| CEC2019 | MO-MFEA-II | | MO-MFEA-AKT | | MO-MFEA-SADE | | EMT/ET | | MTEA/D-DN | | MMTEA-DTS | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | IGD | MSS | IGD | MSS | IGD | MSS | IGD | MSS | IGD | MSS | IGD | MSS |
| CPLX1_1 | 4.08E-04 + (2.30E-05) | 0.39 + | 3.98E-04 + (1.49E-05) | 0.34 + | 3.03E-04 + (1.46E-05) | -0.28 + | 3.97E-04 + (2.11E-05) | 1.45 + | 2.44E-04 ≈ (1.39E-05) | -0.900 + | **2.50E-04 (2.54E-05)** | **-0.903** |
| CPLX1_2 | 2.44E-03 + (8.57E-04) | | 2.59E-03 + (7.17E-04) | | 2.91E-03 + (8.31E-04) | | 1.39E-02 + (4.91E-03) | | 7.83E-04 + (2.02E-04) | | **3.46E-04 (2.10E-05)** | |
| CPLX2_1 | 3.96E-04 + (2.10E-05) | 1.48 + | 3.10E-04 + (2.31E-05) | 0.15 + | 3.18E-04 + (8.79E-05) | 0.29 + | 3.81E-04 + (1.49E-05) | 0.65 + | **2.39E-04 ≈ (1.24E-05)** | -0.86 + | 2.44E-04 (1.96E-05) | **-0.95** |
| CPLX2_2 | 5.52E-03 + (1.23E-03) | | 2.97E-03 + (8.61E-04) | | 3.23E-03 + (9.00E-04) | | 2.98E-03 + (6.69E-04) | | 1.17E-03 + (2.91E-04) | | **7.17E-04 (2.21E-04)** | |
| CPLX3_1 | 5.11E-03 + (7.89E-04) | -0.10 + | 6.00E-03 + (2.32E-03) | 0.04 + | 5.21E-03 + (4.34E-04) | 0.09 + | 1.69E-02 + (1.73E-03) | 1.92 + | 6.70E-03 + (3.34E-03) | 0.16 + | **1.47E-03 (4.21E-04)** | **-0.65** |
| CPLX3_2 | 2.93E-03 + (7.41E-04) | | 3.34E-03 + (6.37E-04) | | 4.32E-03 + (8.72E-04) | | 9.85E-03 + (7.26E-03) | | 3.71E-03 + (7.04E-04) | | **1.35E-03 (1.95E-04)** | |
| CPLX4_1 | 5.57E-03 + (1.59E-03) | 0.06 + | 5.04E-03 + (6.76E-04) | 0.00 + | 4.99E-03 + (3.67E-04) | 1.07 + | 1.74E-02 + (1.07E-03) | 2.01 + | 6.24E-03 + (5.78E-03) | 0.15 + | **1.14E-03 (2.04E-04)** | **-0.42** |
| CPLX4_2 | 5.96E-03 + (1.96E-03) | | 5.30E-03 + (6.17E-04) | | 6.37E-02 + (9.34E-03) | | 5.51E-02 + (1.71E-02) | | 7.45E-03 + (6.75E-03) | | **1.15E-03 (2.35E-04)** | |
| CPLX5_1 | 2.65E-03 + (6.38E-04) | 0.02 + | 2.76E-03 + (4.79E-04) | 0.03 + | 3.85E-02 + (2.30E-03) | 1.26 + | 9.76E-03 + (5.93E-03) | 1.47 + | 3.27E-03 + (2.43E-03) | -0.01 + | **1.25E-03 (1.65E-04)** | **-0.07** |
| CPLX5_2 | 6.63E-03 + (8.72E-04) | | 6.83E-03 + (1.17E-03) | | **2.96E-03 - (3.25E-04)** | | 5.90E-02 + (2.15E-02) | | 4.26E-03 - (2.02E-03) | | 5.30E-03 (8.13E-04) | |
| CPLX6_1 | 2.86E-03 + (8.10E-04) | 0.04 + | 2.55E-03 + (7.39E-04) | -0.01 + | 2.89E-03 + (3.01E-04) | 1.06 + | 8.79E-03 + (6.80E-03) | 1.80 + | 3.37E-03 + (2.46E-03) | 0.11 + | **1.49E-03 (1.92E-04)** | **-0.21** |
| CPLX6_2 | 5.60E-03 + (1.17E-03) | | 5.63E-03 + (7.35E-04) | | 6.12E-02 + (1.14E-02) | | 5.93E-02 + (1.39E-02) | | 6.17E-03 + (4.56E-03) | | **1.99E-03 (2.90E-04)** | |
| CPLX7_1 | 3.06E-03+ (7.20E-04) | 0.09 + | 3.16E-03 + (5.11E-04) | 0.22 + | 3.55E-03 + (5.25E-04) | 0.25 + | 3.77E-03 + (5.51E-04) | 0.78 + | 2.70E-03 + (6.19E-04) | -0.16 + | **1.02E-03 (1.59E-04)** | **-1.19** |
| CPLX7_2 | 2.33E-03 + (9.94E-04) | | 2.59E-03 + (1.07E-03) | | 2.10E-03 + (4.18E-04) | | 3.48E-03 + (3.27E-03) | | 2.08E-03 + (6.04E-04) | | **1.34E-03 (2.61E-04)** | |
| CPLX8_1 | 2.33E-03 + (7.62E-04) | 0.68 + | 2.73E-03 (1.46E-03) | 0.64 + | 1.20E-02 + (4.30E-03) | 0.80 + | 5.06E-03 + (4.84E-03) | 0.73 + | 2.48E-03 + (1.03E-03) | -0.09 + | **1.66E-03 (2.39E-04)** | **-0.45** |
| CPLX8_2 | 1.53E-02 + (6.01E-03) | | 1.42E-02 + (4.26E-03) | | 2.16E-03 + (4.56E-04) | | 1.18E-02 (4.84E-03) | | 4.49E-03 + (1.54E-03) | | **8.15E-04 (8.61E-04)** | |
| CPLX9_1 | 7.07E-03 + (1.47E-03) | 0.02 + | 6.72E-03 + (1.14E-03) | -0.01 + | 3.94E-02 + (2.19E-03) | 1.81 + | 6.30E-02 + (9.31E-03) | 2.10 + | **4.07E-03 - (1.86E-03)** | -0.03 + | 5.29E-03 (5.13E-04) | **-0.10** |
| CPLX9_2 | 6.65E-03 + (2.23E-03) | | 5.24E-03 + (7.73E-04) | | 6.45E-02 + (6.16E-03) | | 5.22E-02 + (1.83E-02) | | 7.37E-03 + (5.81E-03) | | **1.91E-03 (2.82E-04)** | |
| CPLX10_1 | 1.03E-02 + (4.25E-03) | 0.33 + | 1.28E-02 + (4.33E-03) | 0.97 + | 8.56E-03 + (3.75E-03) | 0.19 + | 1.04E-02 + (4.01E-03) | 0.54 + | 5.58E-03 + (1.96E-03) | -0.32 + | **1.39E-03 (7.85E-04)** | **-1.26** |
| CPLX10_2 | 7.58E-03 + (2.83E-03) | | 1.04E-02 + (3.93E-03) | | 7.85E-03 + (1.71E-03) | | 9.08E-03 + (3.28E-03) | | 6.33E-03 + (1.84E-03) | | **2.52E-03 (8.74E-04)** | |
| ≈/+/- | 0/20/0 | 0/10/0 | 0/20/0 | 0/10/0 | 0/19/1 | 0/10/0 | 0/20/0 | 0/10/0 | 2/16/2 | 0/10/0 | / | / |
| Rank | 3.81 | | 3.75 | | 4.51 | | 5.36 | | 2.51 | | **1.06** | |

Table II gives the mean values and standard deviations of the IGD and MSS results obtained by all the compared MMTEAs in 30 independent runs for tackling the CPLX problems in CEC2019. Please note that CPLX1-CPLX10 comprise a variety of MOPs with complicated PS shapes, and there is little degree of intersection in PSs between different tasks except for CPLX2, CPLX4, and CPLX10, which is a major challenge for MMTEAs. From the IGD results in Table II, the overall performance of MMTEA-DTS is significantly better than that of the others. MMTEA-DTS outperforms MO-MFEA-II, MO-MFEA-AKT, MO-MFEA-SADE, and EMT/ET on nearly all 20 tasks. Additionally, MMTEA-DTS performs remarkably better than MTEA/D-DN. As there is little degree of intersection between the tasks, the subspace alignment method used in MO-MFEA-SADE is not effective, which leads to its poor performance. Decomposition-based MMTEAs transform all tasks into a number of subproblems, allowing for the exploration of useful search experiences among these subproblems. This is the primary reason for the superior performance of MTEA/D-DN and MMTEA-DTS over other Pareto-dominance-based MMTEAs. When addressing MMTOPs with little intersection in this test suite, the decomposition-based transfer selection and hybrid transfer evolutionary mechanism used in MMTEA-DTS provide two advantages, which are the main reasons for its superior performance over MTEA/D-DN in 16 out of the total 20 tasks. The first benefit is reducing negative transfer occurrence. When negative transfer occurs on a subproblem, the associated solution's transfer potential will be reduced, making it less likely to be selected for knowledge transfer afterward. The second benefit is diversifying the knowledge transfer and exploring the synergy among different tasks in the evolutionary process. The solution with higher transfer potential will undergo our hybrid transfer evolution, which increases the likelihood of transferring diverse and useful search experiences among the tasks, even when there is little degree of intersection among them. The performance ranks based on the IGD results, as shown in the last row of Table II, further confirm the advantages of MMTEA-DTS for solving this test suite, as its rank (1.06) is the smallest.

Moreover, by observing another comprehensive indicator (MSS) in Table II, the advantages of MMTEA-DTS for solving this test suite are evident, as it outperforms other competitors on all 10 MMTOPs.

*3) Performance Comparisons on ETMOF1-ETMOF8*

Table III gives the mean values and standard deviations of the IGD and MSS results obtained by all the compared MMTEAs in 30 independent runs for tackling ETMOF1-ETMOF8 in CEC2021. As one of the latest test suites in the field of MTO, ETMOF1-ETMOF8 include various MOPs with complicated PSs and multiple PF shapes, which are modified from LZ09 [10], DTLZ [53], and LSMOP [54]. As observed from the IGD results in Table III, MMTEA-DTS performs significantly better than

**TABLE III**

THE MEAN VALUES AND STANDARD DEVIATIONS OF IGD AND MSS RESULTS OBTAINED BY THE COMPARED MMTEAS IN CEC2021

| ETMO2021 | MO-MFEA-II | | MO-MFEA-AKT | | MO-MFEA-SADE | | EMT/ET | | MTEA/D-DN | | MMTEA-DTS | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | IGD | MSS | IGD | MSS | IGD | MSS | IGD | MSS | IGD | MSS | IGD | MSS |
| ETMOF1_1 | 9.42E-04 + (7.19E-05) | 0.39 + | 8.86E-04 + (1.02E-04) | -0.13 + | 8.73E-04 + (4.07E-05) | -0.06 + | 1.06E-03 + (6.12E-05) | 1.14 + | 8.60E-04 + (9.17E-05) | 0.03 + | **6.52E-04 (5.95E-05)** | **-1.66** |
| ETMOF1_2 | 2.14E-03 + (1.70E-04) | | 1.91E-03 + (2.20E-04) | | 1.99E-03 + (1.38E-04) | | 2.38E-03 + (2.39E-04) | | 2.09E-03 + (2.47E-04) | | **1.41E-03 (1.21E-04)** | |
| ETMOF2_1 | 2.67E-03 + (1.65E-03) | 0.55 + | 2.51E-03 + (1.02E-03) | 0.22 + | 1.35E-03 + (2.60E-04) | 0.15 + | 3.94E-03 + (1.51E-03) | 1.47 + | 5.67E-04 + (2.43E-04) | **-0.94 -** | **3.79E-04 (3.56E-05)** | -0.77 |
| ETMOF2_2 | 9.35E-03 + (5.99E-04) | | 8.43E-03 + (3.52E-04) | | 9.40E-03 + (2.81E-04) | | 1.11E-02 + (1.58E-03) | | **6.60E-03 - (4.74E-04)** | | 7.34E-03 (3.14E-04) | |
| ETMOF3_1 | 4.33E-03 + (3.48E-03) | 0.51 + | 4.16E-03 + (2.40E-03) | 0.51 + | 3.81E-03 + (8.01E-04) | 0.13 + | 5.15E-03 + (3.59E-03) | 1.19 + | 3.53E-03 + (3.59E-03) | -0.06 + | **1.95E-03 (4.05E-04)** | **-0.37** |
| ETMOF3_2 | 4.13E-03 + (8.45E-04) | | 4.22E-03 + (6.97E-04) | | 3.34E-03 + (1.34E-03) | | 5.63E-03 + (1.28E-03) | | 2.93E-03 + (1.19E-03) | | **2.84E-03 (5.18E-04)** | |
| ETMOF4_1 | 8.89E+04 + (9.94E+04) | 1.70 + | 2.60E+04 + (1.57E+04) | 0.59 + | **4.25E-01 - (6.89E-02)** | **-0.004 -** | 2.56E+02 + (1.42E+02) | 0.004 + | 8.04E+00 + (3.39E+00) | 0.0013 + | 2.78E+00 (2.65E+00) | 0.0011 |
| ETMOF4_2 | 1.26E+02 + (1.31E+02) | | 5.10E+01 + (5.52E+01) | | **1.88E-01 - (7.54E-03)** | | 4.12E-01 + (5.80E-02) | | 4.46E-01 + (5.30E-02) | | 3.47E-01 (5.89E-02) | |
| ETMOF5_1 | 1.39E+02 + (6.25E+01) | 1.99 + | 1.16E+02 + (4.49E+01) | 1.90 + | 2.67E-01 ≈ (1.51E-03) | -0.06 + | 2.62E-01 ≈ (5.16E-03) | -0.06 + | 2.71E-01 + (2.94E-03) | 0.17 + | **2.66E-01 (2.68E-03)** | **-0.07** |
| ETMOF5_2 | 1.33E+00 + (1.60E-01) | | 1.43E+00 + (1.68E-01) | | 2.77E-01 + (2.80E-02) | | 2.38E-01 ≈ (2.36E-02) | | 4.84E-01 + (1.58E-01) | | **2.34E-01 (2.58E-02)** | |
| ETMOF6_1 | 2.80E+03 + (8.31E+03) | 0.97 + | 3.36E+03 + (5.49E+03) | 0.89 + | 1.04E+05 + (5.31E+05) | 0.24 + | 1.13E+01 + (1.41E+01) | 0.003 + | 6.91E-01 ≈ (2.92E-01) | 0.001 + | **6.65E-01 (2.01E-01)** | **-0.001** |
| ETMOF6_2 | 1.78E+03 + (7.04E+02) | | 1.62E+03 + (8.62E+02) | | **4.17E-02 ≈ (6.10E-03)** | | 1.64E-01 + (8.11E-02) | | 4.87E-02 + (3.50E-02) | | 4.27E-02 (3.41E-02) | |
| ETMOF7_1 | 5.41E-03 + (3.92E-03) | 0.79 + | 6.08E-03 + (3.67E-03) | 1.48 + | 7.01E-04 - (2.16E-04) | -0.16 + | 1.56E-03 ≈ (2.79E-04) | 0.03 + | **4.75E-04 - (2.16E-04)** | **-0.21** - | 1.69E-03 (1.22E-03) | 0.05 |
| ETMOF7_2 | 3.47E-02 + (2.83E-02) | | 9.96E-02 + (1.09E-01) | | 1.29E-02 - (2.22E-03) | | 1.82E-02 ≈ (2.16E-03) | | **1.13E-02 - (2.38E-03)** | | 1.87E-02 (6.44E-03) | |
| ETMOF7_3 | 4.37E-02 + (6.82E-03) | | 3.67E-02 + (5.07E-03) | | / | | / | | **1.77E-02 - (3.57E-03)** | | 2.18E-02 (3.33E-03) | |
| ETMOF8_1 | 7.35E-02 + (3.95E-02) | 0.47 + | 5.00E-02 + (2.27E-02) | 0.01 + | 1.68E-01 + (8.89E-02) | 1.17 + | 5.41E-02 + (2.43E-02) | 0.33 + | 8.83E-03 + (2.34E-03) | -0.91 + | **7.58E-03 (8.01E-04)** | **-1.07** |
| ETMOF8_2 | 2.99E-02 + (1.01E-02) | | 1.31E-02 + (3.33E-03) | | 2.18E-02 + (7.26E-03) | | 2.11E-02 + (5.32E-03) | | 2.39E-03 + (8.23E-04) | | **1.99E-03 (2.00E-04)** | |
| ETMOF8_3 | 9.49E-03 + (1.35E-03) | | 6.79E-03 + (1.15E-03) | | / | | / | | **1.24E-03 ≈ (2.07E-04)** | | 1.29E-03 (8.99E-05) | |
| ≈/+/- | 0/18/0 | 0/8/0 | 0/18/0 | 0/8/0 | 2/10/4 | 0/7/1 | 4/12/0 | 0/8/0 | 2/12/4 | 0/6/2 | / | / |
| Rank | 4.93 | | 4.47 | | 3.33 | | 4.34 | | 2.24 | | **1.70** | |

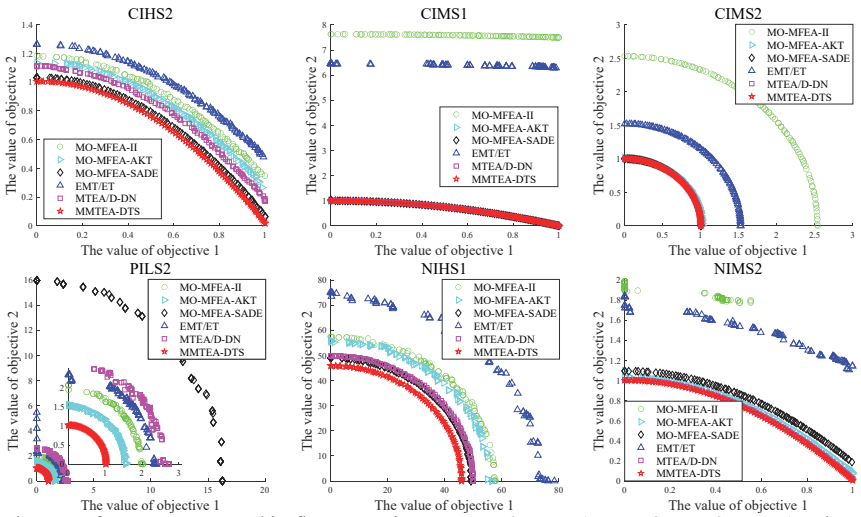


Fig. 1 Final solution sets of MMTEA-DTS and its five competitors on CIHS2, CIMS1, CIMS2, PILS2, NIHS1 and NIMS2 in CEC2017

MO-MFEA-II, MO-MFEA-AKT, MO-MFEA-SADE, EMT/ET and MTEA/D-DN on 18, 18, 10, 12 and 12 cases out of a total of 18 tasks, respectively. Since the public source codes of MO-MFEA-SADE and EMT/ET cannot solve the MMTOP with more than two tasks, only the first two tasks of ETMOF7 and ETMOF8 are solved by them. MMTEA-DTS can provide a better overall performance than other competitors, and its performance rank (1.70) in the last row of Table III is the smallest, which validates the advantages of MMTEA-DTS in handling these complex MMTOPs.

Moreover, by observing another comprehensive indicator (MSS) in Table III, when compared with MO-MFEA-II, MO-MFEA-AKT, MO-MFEA-SADE, EMT/ET, and MTEA/D-DN, MMTEA-DTS performs significantly better on 8, 8, 7, 8, and 6 cases out of a total of 9 test MMTOPs, respectively. Only MO-MFEA-SADE and MTEA/D-DN can outperform MMTEA-DTS on 1 and 2 cases, respectively. Thus, these MSS results further confirm the advantages of MMTEA-DTS on this test suite.

**TABLE IV**
THE SUMMARY OF COMPARISONS OF THE COMPARED MMTEA-DTS VARIANTS IN ALL THE TEST SUITES

| | NO-KT | | | NO-DTS | | | KT-1 | | | KT-2 | | | MMTEA-DTS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | IGD (≈/+/-) | MSS (≈/+/-) | Rank | IGD (≈/+/-) | MSS (≈/+/-) | Rank | IGD (≈/+/-) | MSS (≈/+/-) | Rank | IGD (≈/+/-) | MSS (≈/+/-) | Rank | IGD (≈/+/-) | MSS (≈/+/-) | Rank |
| CEC2017 | 1/15/2 | 0/9/0 | 4.61 | 2/14/2 | 0/7/2 | 3.80 | 11/7/0 | 0/7/2 | 2.39 | 14/2/2 | 0/5/4 | 2.16 | / | / | **2.04** |
| CEC2019 | 6/12/2 | 0/10/0 | 4.04 | 8/12/0 | 0/10/0 | 3.83 | 18/1/1 | 0/4/6 | 2.40 | 19/0/1 | 0/7/3 | 2.41 | / | / | **2.32** |
| CEC2021 | 1/16/1 | 0/8/0 | 4.43 | 3/13/2 | 0/7/1 | 3.62 | 13/5/0 | 0/7/1 | 2.50 | 18/0/0 | 0/4/4 | 2.23 | / | / | **1.70** |

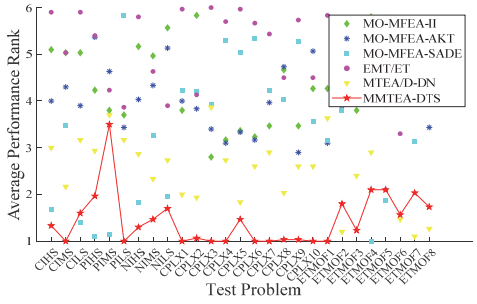


Fig. 2 Average performance ranks of the compared MMTEAs on each case

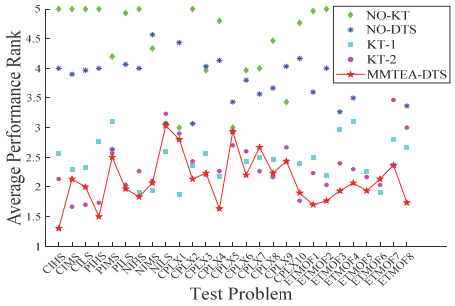


Fig. 3 Average performance ranks of the compared variants on each case

*4) Visual Comparisons*

To visually compare all the compared MMTEAs, their final solution sets with the 15th best IGD metric in 30 independent runs are plotted in Fig. 1 for solving CIHS2, CIMS1, CIMS2, PILS2, NIHS1 and NIMS2 in CEC2017, in Fig. S1 of the Supplementary Material for solving CPLX1_2 in CEC2019, and in Fig. S2 of the Supplementary Material for solving ETMOF2_1 in CEC2021 due to page limitations. From these plots, it is easy to observe that MMTEA-DTS obtains the best approximation to the PF in terms of convergence and diversity.

Moreover, in order to provide a comprehensive visual comparison of all the adopted MMTEAs, their average performance ranks on each test MMTOP are obtained by using Friedman's test [52] on the IGD results. These ranks are plotted in Fig. 2, where a lower rank indicates a better performance, and the ranks of MMTEA-DTS are connected by a red line for ease of observation. As shown in Fig. 2, MMTEA-DTS performs best on most of the test cases.

## E. More Discussions

*1) Effectiveness of the Proposed Methods*

To analyze the effectiveness of the proposed methods, the performance of MMTEA-DTS is further compared to its four variants: (1) MMTEA-DTS without using knowledge transfer by setting $Tp = 0$, denoted as NO-KT, (2) MMTEA-DTS without our decomposition-based transfer selection by setting all the transfer potentials to 1, denoted as NO-DTS, and (3) MMTEA-DTS without using our hybrid transfer evolutionary mechanism by only transferring the offspring generated by the parents in the same task into one different task or only transferring the offspring generated by the parents from different tasks into one random task, denoted as KT-1 and KT-2, respectively. Each algorithm is run 30 times independently, and the parameters settings of all the variants are set the same as that of MMTEA-DTS for a fair comparison. Due to page limitations, Tables B.I-B. III of the Supplementary Material provide the IGD and MSS results obtained by all the compared MMTEA-DTS variants for tackling three test suites of MMTOPs from the MTO computation in CEC2017, CEC2019, and CEC2021, respectively. Table IV collects the summary of comparisons of these compared MMTEA-DTS variants in all the test suites. From the statistical results (≈/+/-) of IGD and MSS, it can be observed that MMTEA-DTS performs better than the other variants in most cases. According to the ranks in Table IV, MMTEA-DTS performs best in each of the three test suites, as its ranks are 2.04, 2.32, and 1.70 for the three test suites, respectively. Without knowledge transfer across different tasks, NO-KT performs worst, as its ranks are respectively 4.61, 4.04, and 4.43 for the three test suites, which validates that knowledge transfer is the main contributor to the performance improvement of MMTEA-DTS. In NO-DTS, due to the absence of decomposition-based transfer selection, it also shows relatively poor performance with ranks of 3.80, 3.83, and 3.62 respectively for the three test suites, indicating that decomposition-based transfer selection is the secondary contributor to the performance improvement of MMTEA-DTS. KT-1 and KT-2 perform relatively better, as their ranks are (2.39, 2.40, and 2.50) and (2.16, 2.41, and 2.23) for the three test suites, respectively. However, MMTEA-DTS still shows some advantages over KT-1 and KT-2 by combining their advantages. Thus, the hybrid transfer evolutionary mechanism is the third contributor to our performance improvement.

To visually show the comprehensive comparison of all the adopted MMTEA-DTS variants, their average performance ranks on each test MMTOP are obtained by using Friedman's test [52] on the IGD results, which are plotted in Fig. 3. Here, the ranks of MMTEA-DTS are connected by a red line for ease of observation. As observed in Fig. 3, MMTEA-DTS obtains the first rank in most cases. Moreover, the convergence curves obtained by all the adopted MMTEA-DTS variants are plotted in Fig. 4 for solving the problems with three degrees of intersection (CI, PI and NI) in CEC2017. For the tasks with

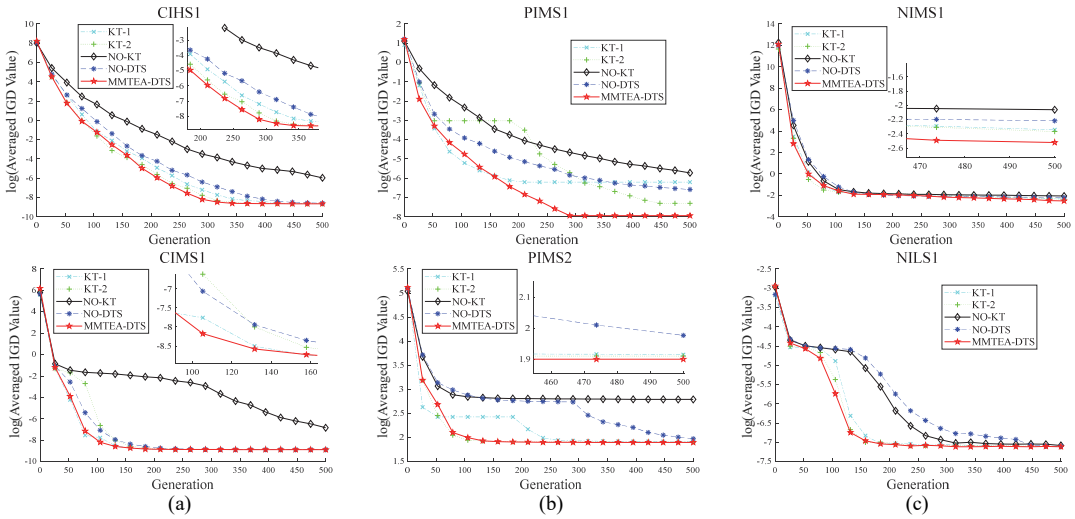


Fig. 4 The convergence curves obtained by the compared MMTEA-DTS variants on (a) CI, (b) PI and (c) NI problems

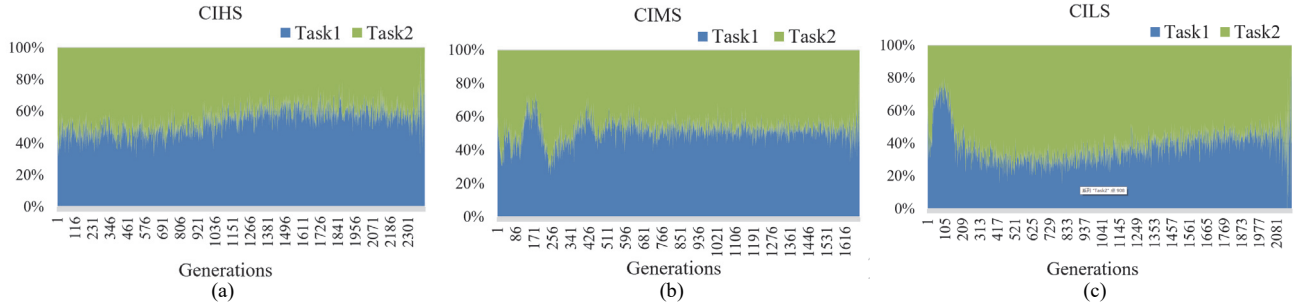


Fig. 5 Transfer ratios of MMTEA-DTS on (a) CIHS, (b) CIMS and (c) CILS

complete intersection (CIHS1 and CIMS1), Fig. 4(a) indicates that the adoption of knowledge transfer in MMTEA-DTS can always significantly speed up convergence. This is because the high-potential transfer solutions selected by the decomposition-based method can greatly promote the transfer effect. For the tasks with partial intersection (PIMS1 and PIMS2) in Fig. 4(b), although the convergence speed of MMTEA-DTS is not the fastest at the beginning, it could still reach the best convergence at the end by combining the advantages of the decomposition-based transfer selection and the hybrid transfer evolutionary mechanism. For the tasks with no intersection (NIMS1 and NILS1) in Fig. 4(c), the potential synergy of tasks can still speed up convergence to some extent, and all the variants could finally achieve the same degree of convergence. Based on these plots in Fig. 4, it is reasonable to conclude that transferring valuable knowledge across tasks using the proposed decomposition-based transfer selection and hybrid transfer evolution can speed up convergence for solving various tasks with different degrees of intersection.

*2) Analysis of Transfer Ratios and Directions*

In MMTEA-DTS, an adaptive transfer ratio can be realized through the use of decomposition-based transfer selection. Due to page limitations, CIHS, CIMS, and CILS are taken as examples, and their transfer ratios in the evolutionary process are visually plotted in Fig. 5, which are averaged from 30 independent runs of MMTEA-DTS. As observed from Fig. 5, the transfer ratios evidently fluctuate at each generation of the evolutionary process, as the demands for knowledge transfer between two tasks are not always the same. In MMTEA-DTS, decomposition-based transfer selection can estimate the transfer potential of each solution based on the performance improvement ratio to its associated subproblem, which could be used to dynamically adjust the transfer ratios among all tasks. As observed in Fig. 5(a), more solutions in task2 are transferred to task1 in the early stages of the evolutionary process, while at later stages, this situation reverses. However, in Fig. 5(b), the transfer ratios of task1 significantly fluctuate in the early stages of the evolutionary process and become relatively stable at a later stage. In Fig. 5(c), a large number of solutions in task1 are transferred to task2 during a short period of the early stages of the evolutionary process, and more solutions in task2 are transferred to task1 afterwards. Thus, based on the above observations in Fig. 5, MMTEA-DTS can dynamically adjust the transfer ratios and directions among different tasks for solving various MMTOPs.

*3) Sensitivity Analysis*

In this experiment, the sensitivity of parameter $a$ used in MMTEA-DTS is studied, which controls the size of the potential transfer solution set. Here, the values of this parameter are set as $a$ = 1, 3, 5, 7, 9, while other parameters in MMTEA-DTS are set the same as in the above experiments. All three test suites mentioned above are used, and Friedman's test [52] is run on the IGD results for statistical analysis. Due to page limitations, Tables B.IV-B. VI of the Supplementary Material provide the IGD and MSS results obtained by MMTEA-DTS with different $a$ values for tackling three test

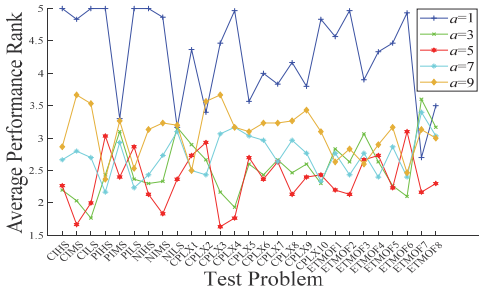


Fig. 6 Average performance rankings obtained by MMTEA-DTS with different $a$ values based on the IGD results

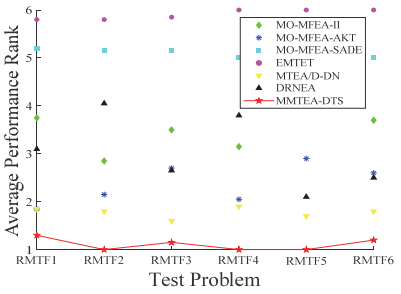


Fig. 7 Average performance ranks of the compared MMTEAs on each real-world problem

suites of MMTOPs from the MTO computation in CEC2017, CEC2019, and CEC2021, respectively. Fig. 6 plots their average performance ranks for each test problem based on the IGD results. As observed from Fig. 6, the average performance ranks of MMTEA-DTS gradually improve from $a = 1$ to $a = 3$ and then are steady from $a = 3$ to $a = 7$, but slightly deteriorate from $a = 7$ to $a = 9$. That is, when $a$ is set too small (i.e., $a = 1$), the solutions are more likely to be treated equally, so the effect of our decomposition-based transfer selection mechanism cannot be displayed; however, when $a$ is set relatively large (i.e., $a = 9$), only a few solutions are selected to be evolved and for knowledge transfer, which may result in a lack of diversity in the evolutionary process. Thus, $a$ is recommended to be set between 3 and 7, as the IGD results of MMTEA-DTS with these corresponding $a$ values are statistically similar in most test cases.

*4) Real-world Application*

To evaluate the real-world applicability of MMTEA-DTS, six real-world MMTOPs (RTMF1- RTMF6) in RTMF [42] were adopted to simulate the training of deep neural networks (DNNs) on multiple different bi-classification tasks, where each task endeavors to optimize the weights (i.e., the variables) of the involved DNN. As defined in [55], two objectives (complexity and classification error of DNNs) are concurrently minimized. For instance, in RMTF2, task 1 is to distinguish between two different varieties of raisins (Kecimen or Besni) [56], while task 2 is to distinguish between two different varieties of rice (Cammeo or Osmancik) [57]. Due to page limitations, the detailed definitions of these RMTF test problems are given in Table A.VIII.

In this study, MMTEA-DTS is compared to MO-MFEA-II, MO-MFEA-AKT, MO-MFEA-SADE, EMT/ET, MTEA/D-DN and DRNEA (a baseline algorithm in [42]) for solving these RMTF test problems. Each task has a population size of 50, and each MMETA runs for 200 iterations. The parameters settings for DRNEA are the same as suggested in [42], and the parameters settings for other MMETAs are the same as in Section IV.B. Since the PFs of practical RMTF problems (RMTFs) are unknown, the hypervolume (HV) [58] with a reference point (1, 1) is used as the metric to assess the performance of each solver for RMTFs. Generally, a larger value of HV indicates a better performance of MMTEA. All the compared MMTEAs are run 20 times independently, and the comparison results of HV values are provided in Table B. VII of the Supplementary Material due to page limitations. Fig. 7 plots their average performance ranks for each RMTF problem by using Friedman's test [52] based on the average HV value of the tasks in each RMTF problem, where the ranks of MMTEA-DTS are connected by a red line for easy observation. MMTEA-DTS performs best on all the RMTF problems, as observed from Fig. 7. MMTEA-DTS also obtains the best rank on 14 out of a total of 15 cases, as shown in Table B.VII. This is because our algorithm is able to consistently select the high-potential solutions for knowledge transfer, which are further diversified by our hybrid transfer evolutionary mechanism, resulting in enhanced performance. Thus, this experiment provides empirical evidence for the effectiveness of MMTEA-DTS in real-world applications, specifically in training DNNs for bi-classification problems.

## V. Conclusions and Future Work

This paper has proposed a new multiobjective multitasking evolutionary algorithm with decomposition-based transfer selection, called MMTEA-DTS. Decomposition-based transfer selection can properly quantify the transfer potential of each solution for knowledge transfer based on the feedback of the improvement ratio to its associated subproblem, which helps to alleviate the negative transfer effect. Moreover, to diversify and explore more useful search experiences, a hybrid transfer evolutionary mechanism is proposed in MMTEA-DTS. The experiments have validated that both operations can be cooperatively used to fully exploit the potential complementarity among the tasks and that the proposed decomposition-based transfer selection has a significant contribution to our performance improvement. The experiments have also validated that MMTEA-DTS outperforms four Pareto-dominance-based MMTEAs (MO-MFEA-II [32], MO-MFEA-AKT [30], MO-MFEA-SADE [34], and EMT/ET [39]) and one decomposition-based MMTEA (MTEA/D-DN [47]) when solving most cases of three well-known test suites and one real-world problem suite.

Although our algorithm has shown very promising performance, there are still some areas that warrant further investigation in our future work. For instance, the performance of distinct decomposition methods for various tasks should be further studied to facilitate knowledge transfer. Additionally, the relevance of different subproblems among the tasks should be further studied to improve the efficiency of mating selection and evolution during knowledge transfer. Finally, the application of this algorithm in more real-world MMTOPs [59]

should be further studied in future work.

## REFERENCES


[1] C.-Y. Lin, "Fuzzy AHP-based Prioritization of the Optimal Alternative of External Equity Financing for Start-ups of Lending Company in Uncertain Environment," *Romanian Journal of Information Science and Technology*, vol. 25, pp. 133–149, 2022.

[2] I. A. Zamfirache, R.-E. Precup, R.-C. Roman, and E. M. Petriu, "Policy Iteration Reinforcement Learning-based control using a Grey Wolf Optimizer algorithm," *Information Sciences*, vol. 585, pp. 162–175, Mar. 2022.

[3] C. Pozna, R. -E. Precup, E. Horváth and E. M. Petriu, "Hybrid Particle Filter–Particle Swarm Optimization Algorithm and Application to Fuzzy Controlled Servo Systems," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 10, pp. 4286–4297, Oct. 2022.

[4] J. Han, C. Yang, C.-C. Lim, X. Zhou, and P. Shi, "Stackelberg–Nash Game Approach for Constrained Robust Optimization With Fuzzy Variables," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 11, pp. 3519–3531, Nov. 2021.

[5] R.-E. Precup, R.-C. David, R.-C. Roman, E. M. Petriu, and A.-I. Szedlak-Stinean, "Slime Mould Algorithm-Based Tuning of Cost-Effective Fuzzy Controllers for Servo Systems," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, pp. 1042–1052, Mar. 2021.

[6] J. Han, C. Yang, C.-C. Lim, X. Zhou, P. Shi, and W. Gui, "Power scheduling optimization under single-valued neutrosophic uncertainty," *Neurocomputing*, vol. 382, pp. 12–20, Mar. 2020.

[7] A. Zhou, B.Y. Qu, H. Li, S.Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective Evolutionary Algorithms: A Survey of the State of The Art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, Mar. 2011.

[8] F. Zhao, S. Di, and L. Wang, "A Hyperheuristic With Q-Learning for the Multiobjective Energy-Efficient Distributed Blocking Flow Shop Scheduling Problem," *IEEE Transactions on Cybernetics*, in press, 2022, doi: 10.1109/TCYB.2022.3192112.

[9] P. Zhou, X. Wang, and T. Chai, "Multiobjective Operation Optimization of Wastewater Treatment Process Based on Reinforcement Self-Learning and Knowledge Guidance," *IEEE Transactions on Cybernetics*, in press, 2022, doi: 10.1109/TCYB.2022.3164476.

[10] H. Li and Q.F. Zhang, "Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, Apr. 2009.

[11] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[12] Q. Lin *et al.*, "A Clustering-based Evolutionary Algorithm for Many-objective Optimization Problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 391–405, Jun. 2019.

[13] Y.-S. Ong and A. Gupta, "Evolutionary Multitasking: A Computer Science View of Cognitive Multitasking," *Cognitive Computation*, vol. 8, no. 2, pp. 125–142, Apr. 2016.

[14] X. Ji, Y. Zhang, D. Gong, X. Sun, and Y. Guo, "Multisurrogate-Assisted Multitasking Particle Swarm Optimization for Expensive Multimodal Problems," *IEEE Transactions on Cybernetics*, in press, 2021, doi: 10.1109/TCYB.2021.3123625.

[15] L. Bai, W. Lin, A. Gupta, and Y.-S. Ong, "From Multitask Gradient Descent to Gradient-Free Evolutionary Multitasking: A Proof of Faster Convergence," *IEEE Transactions on Cybernetics*, vol. 52, no. 8, pp. 8561–8573, Aug. 2022.

[16] Z. Chen, Y. Zhou, X. He, and J. Zhang, "Learning Task Relationships in Evolutionary Multitasking for Multiobjective Continuous Optimization," *IEEE Transactions on Cybernetics*, vol. 52, no. 6, pp. 5278–5289, Jun. 2022.

[17] K. Chen, B. Xue, M. Zhang, and F. Zhou, "An Evolutionary Multitasking-Based Feature Selection Method for High-Dimensional Classification," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 7172–7186, Jul. 2022.

[18] L. Feng *et al.*, "Solving Generalized Vehicle Routing Problem With Occasional Drivers via Evolutionary Multitasking," *IEEE Transactions on Cybernetics*, vol. 51, no. 6, pp. 3171–3184, Jun. 2021.

[19] A. Gupta, Y.-S. Ong, and L. Feng, "Insights on Transfer Optimization: Because Experience is the Best Teacher," *IEEE Transactions on Emerging Topics in Computational Intelligen*ce, vol. 2, no. 1, pp. 51–64, Feb. 2018.

[20] A. Gupta, Y.-S. Ong, and L. Feng, "Multifactorial Evolution: Toward Evolutionary Multitasking," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 343–357, Jun. 2016.

[21] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[22] M. Gong, Z. Tang, H. Li, and J. Zhang, "Evolutionary Multitasking With Dynamic Resource Allocating Strategy," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 858–869, Oct. 2019.

[23] G. Li, Q. Lin, and W. Gao, "Multifactorial Optimization Via Explicit Multipopulation Evolutionary Framework," *Information Science*, vol. 512, pp. 1555–1570, Feb. 2020.

[24] X. Ma *et al.*, "Improving Evolutionary Multitasking Optimization by Leveraging Inter-Task Gene Similarity and Mirror Transformation," *IEEE Computational Intelligence Magazine*, vol. 16, no. 4, pp. 38–53, Nov. 2021.

[25] X. Xue *et al.*, "Affine Transformation-Enhanced Multifactorial Optimization for Heterogeneous Problems," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 6217–6231, Jul. 2022.

[26] Y.R. Zhou, Z.F. Chen, Z.X. Huang, and Y. Xiang, "A Multiobjective Evolutionary Algorithm Based on Objective-Space Localization Selection," *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 3888–3901, May 2022.

[27] S. Mittal, D. K. Saxena, K. Deb, and E. D. Goodman, "Enhanced Innovized Progress Operator for Evolutionary Multiand Many-objective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 961–975, Oct 2022.

[28] H.K. Chen, G.H. Wu, W. Pedrycz, P.N. Suganthan, L.N. Xing, and X.M. Zhou, "An Adaptive Resources Allocation Strategy for Objective Space Partition-based Multiobjective Optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 3, pp. 1507–1522, March 2021.

[29] A. Gupta, Y.-S. Ong, L. Feng, and K. C. Tan, "Multiobjective Multifactorial Optimization in Evolutionary Multitasking," *IEEE Transactions on Cybernetics*, vol. 47, no. 7, pp. 1652–1665, Jul. 2017.

[30] L. Zhou *et al.*, "Toward Adaptive Knowledge Transfer in Multifactorial Evolutionary Computation," *IEEE Transactions on Cybernetics*, vol. 51, no. 5, pp. 2563–2576, May 2021.

[31] L. Feng *et al.*, "Evolutionary Multitasking via Explicit Autoencoding," *IEEE Transactions on Cybernetics*, vol. 49, no. 9, pp. 3457–3470, Sep. 2019.

[32] K. K. Bali, A. Gupta, Y.-S. Ong, and P. S. Tan, "Cognizant Multitasking in Multiobjective Multifactorial Evolution: MO-MFEA-II," *IEEE Transactions on Cybernetics*, vol. 51, no. 4, pp. 1784–1796, Apr. 2021.

[33] Z. Xu, X. Liu, K. Zhang, and J. He, "Cultural Transmission Based Multiobjective Evolution Strategy for Evolutionary Multitasking," *Information Sciences*, vol. 582, pp. 215–242, Jan. 2022.

[34] Z. Liang, H. Dong, C. Liu, W. Liang, and Z. Zhu, "Evolutionary Multitasking for Multiobjective Optimization With Subspace Alignment and Adaptive Differential Evolution," *IEEE Transactions on Cybernetics*, vol. 52, no. 4, pp. 2096–2109, Apr. 2022.

[35] W. Gao, J. Cheng, M. Gong, H. Li, and J. Xie, "Multiobjective Multitasking Optimization With Subspace Distribution Alignment and Decision Variable Transfer," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 4, pp. 818–827, Aug. 2022.

[36] S. Yao, Z. Dong, X. Wang, and L. Ren, "A Multiobjective Multifactorial Optimization Algorithm based on Decomposition and Dynamic Resource Allocation Strategy," *Information Sciences*, vol. 511, pp. 18–35, Feb. 2020.

[37] Q. Zhang, W. Liu, and H. Li, "The Performance of A New Version of MOEA/D on CEC09 Unconstrained MOP Test Instances," in *2009 IEEE Congress on Evolutionary Computation*, pp. 203–208, May 2009.

[38] T. Wei and J. Zhong, "Towards Generalized Resource Allocation on Evolutionary Multitasking for Multi-Objective Optimization," *IEEE Computational Intelligence Magazine*, vol. 16, no. 4, pp. 20–37, Nov. 2021.

[39] J. Lin, H.L. Liu, K. C. Tan, and F. Gu, "An Effective Knowledge Transfer Approach for Multiobjective Multitasking Optimization," *IEEE Transactions on Cybernetics*, vol. 51, no. 6, pp. 3238–3248, Jun. 2021.

[40] J. Lin, H.L. Liu, B. Xue, M. Zhang, and F. Gu, "Multiobjective Multitasking Optimization Based on Incremental Learning," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 824–838, Oct. 2020.

[41] H. Chen, H.L. Liu, F. Gu, and K. C. Tan, "A Multiobjective Multitask Optimization Algorithm Using Transfer Rank," *IEEE Transactions on Evolutionary Computation*, in press, 2022, doi: 10.1109/TEVC.2022.3147568.

[42] S. Liu, Q. Lin, L. Feng, K.-C. Wong, and K. C. Tan, "Evolutionary Multitasking for Large-Scale Multiobjective Optimization," *IEEE Transactions on Evolutionary Computation*, in press, 2022, doi: 10.1109/TEVC.2022.3166482.
[43] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[44] Q.Z. Lin *et al.*, "A Diversity-Enhanced Resource Allocation Strategy for Decomposition-Based Multiobjective Evolutionary Algorithm," *IEEE Transactions on Cybernetics,* vol. 48, no. 8, pp. 2388–2401, Aug. 2018.
[45] S. Liu, Q. Lin, K. C. Tan, M. Gong and C. A. Coello Coello, "A Fuzzy Decomposition-Based Multi/Many-Objective Evolutionary Algorithm," *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 3495–3509, May 2022.
[46] Y. Su, Q. Lin, Z. Ming and K. C. Tan, "Adapting Decomposed Directions for Evolutionary Multiobjective Optimization," *IEEE Transactions on Cybernetics*, in press, 2022, doi: 10.1109/TCYB.2022.3165557.
[47] X. Wang, Z. Dong, L. Tang, and Q. Zhang, "Multiobjective Multitask Optimization-Neighborhood as a Bridge for Knowledge Transfer," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 1, pp. 155-169, Feb. 2023.
[48] Y. Yuan *et al.*, "Evolutionary Multitasking for Multiobjective Continuous Optimization: Benchmark Problems, Performance Metrics and Baseline Results," arXiv, Jun. 2017. https://doi.org/10.48550/arXiv.1706.02766.
[49] S. Liu, Q. Lin, K. C. Tan, and Q. Li, "Benchmark Problems for CEC2021 Competition on Evolutionary Transfer Multiobjectve Optimization," arXiv, Oct. 2021. https://doi.org/10.48550/arXiv.2110.08033.
[50] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance Assessment of Multiobjective Optimizers: An Analysis and Review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, Apr. 2003.
[51] P. A. N. Bosman and D. Thierens, "The Balance between Proximity and Diversity in Multiobjective Evolutionary Algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, Apr. 2003.
[52] J. Alcalá-Fdez *et al.*, "KEEL: A Software Tool to Assess Evolutionary Algorithms for Data Mining Problems," *Soft Computing*, vol. 13, no. 3, pp. 307–318, Feb. 2009.
[53] K. Deb, "Multiobjective Genetic Algorithms: Problem Difficulties and Construction of Test Problems," *Evolutionary Computation*, vol. 7, no. 3, pp. 205–230, Sep. 1999.
[54] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "Test Problems for Large-Scale Multiobjective and Many-Objective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 47, no. 12, pp. 4108–4121, Dec. 2017.
[55] Y. Jin, T. Okabe, and B. Sendhoff, "Neural network regularization and ensembling using multiobjective evolutionary algorithms," *Proceedings of the 2004 Congress on Evolutionary Computation*, vol. 1, pp. 1–8, Jun. 2004.
[56] L. While P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 29–38, Feb. 2006.
[57] I. Cinar, M. Koklu and S. Tasdemir, "Classification of Raisin Grains Using Machine Vision and Artificial Intelligence Methods", *Gazi Journal of Engineering Sciences*, vol. 6, no. 3, pp. 200–209, Dec. 2020.
[58] I. Cinar and M. Koklu, "Classification of Rice Varieties Using Artificial Intelligence Methods." *International Journal of Intelligent Systems and Applications in Engineering*, vol. 7, no. 3, pp. 188–194, Sep. 2019.
[59] A. Gupta, L. Zhou, Y.-S. Ong, Z. Chen, and Y. Hou, "Half a Dozen Real-World Applications of Evolutionary Multitasking, and More," *IEEE Computational Intelligence Magazine*, vol. 17, no. 2, pp. 49–66, May 2022.