Individual-Based Transfer Learning for Dynamic Multiobjective Optimization

Min Jiang, *Senior Member, IEEE*, Zhenzhong Wang, *Graduate Student Member, IEEE*, Shihui Guo, *Member, IEEE*, Xing Gao, *Member, IEEE*, and Kay Chen Tan, *Fellow, IEEE*

***Abstract*—Dynamic multiobjective optimization problems (DMOPs) are characterized by optimization functions that change over time in varying environments. The DMOP is challenging because it requires the varying Pareto-optimal sets (POSs) to be tracked quickly and accurately during the optimization process. In recent years, transfer learning has been proven to be one of the effective means to solve dynamic multiobjective optimization. However, the negative transfer will lead the search of finding the POS to a wrong direction, which greatly reduces the efficiency of solving optimization problems. Minimizing the occurrence of negative transfer is thus critical for the use of transfer learning in solving DMOPs. In this article, we propose a new individual-based transfer learning method, called an individual transfer-based dynamic multiobjective evolutionary algorithm (IT-DMOEA), for solving DMOPs. Unlike existing approaches, it uses a presearch strategy to filter out some high-quality individuals with better diversity so that it can avoid negative transfer caused by individual aggregation. On this basis, an individual-based transfer learning technique is applied to accelerate the construction of an initial population. The merit of the IT-DMOEA method is that it combines different strategies in maintaining the advantages of transfer learning methods as well as avoiding the occurrence of negative transfer; thereby greatly improving the quality of solutions and convergence speed. The experimental results show that the proposed IT-DMOEA approach can considerably improve the quality of solutions and convergence speed compared to several state-of-the-art algorithms based on different benchmark problems.**

***Index Terms*—Dynamic multiobjective optimization, evolutionary algorithm, prediction, transfer learning.**

Manuscript received September 15, 2019; revised December 25, 2019, March 15, 2020, and May 28, 2020; accepted August 7, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61673328 and Grant 61876162; in part by the Shenzhen Scientific Research and Development Funding Program under Grant JCYJ20180307123637294; and in part by the Research Grants Council of the Hong Kong under Grant CityU11202418 and Grant CityU11209219. This article was recommended by Associate Editor Y. S. Ong. *(Corresponding authors: Zhenzhong Wang; Xing Gao.)*

Min Jiang, Zhenzhong Wang, Shihui Guo, and Xing Gao are with the School of Informatics, Xiamen University, Xiamen 361005, China (e-mail: zhenzhongwang0616@gmail.com; gaoxing@xmu.edu.cn).

Kay Chen Tan is with the Department of Computer Science, City University of Hong Kong, Hong Kong, and also with the City University of Hong Kong Shenzhen Research Institute, Shenzhen 518057, China (e-mail: kaytan@cityu.edu.hk).

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

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.2020.3017049

I. Introduction

It is known that many optimization problems in science and engineering involved multiple conflicting objective functions that need to be optimized simultaneously [1], [2]. These are called dynamic multiobjective optimization problems (DMOPs) if the objective functions to be optimized change over time and/or in varying environments [3]. For example, in the design of robot motion [4], [5], hundreds of decision variables, such as joint angles, robot attitude, and weight, are involved. It is often desired to optimize these decision variables simultaneously such that the corresponding objective functions representing the requirements of energy consumption, speed, and stability of the robot are satisfied. The challenge is that these optimization functions often change with the environment of the robot and, thus, fast and efficient dynamic multiobjective optimization is needed in order to rapidly enable any additional rational movements of the robot in practice.

Many DMOP algorithms have been proposed, particularly prediction-based strategies have gained increased attention in recent years. Muruganantham *et al.* [6] proposed a predictive dynamic multiobjective evolutionary algorithm called MOEA/D-KF, which is based on the Kalman filter. When an environmental change is detected, the Kalman filter is used to guide the search for a new Pareto-optimal set (POS). Koo *et al.* [7] introduced a predictive gradient strategy that can predict the direction and magnitude of change, and this strategy can help the algorithm improve its convergence.

The essence of solving the multiobjective optimization problem is to find the POS, but in the context of dynamic multiobjective optimization, the distribution of POS under different environments or at different times may vary, although a link probably exists between these distributions [8]. Based on this hypothesis, Jiang *et al.* [8] proposed an algorithmic framework based on transfer learning [9], called the Tr-DMOEA, to predict an initial population. The Tr-DMOEA studies DMOPs from different perspectives, especially when the distribution of training and testing samples does not satisfy the independently identically distributed (IID) hypothesis, the prediction approaches based on this framework exhibit excellent performance.

However, existing dynamic multiobjective optimization approaches based on transfer learning have room for further improvement in different aspects. First, the negative transfer may arise when the training and testing samples are extremely

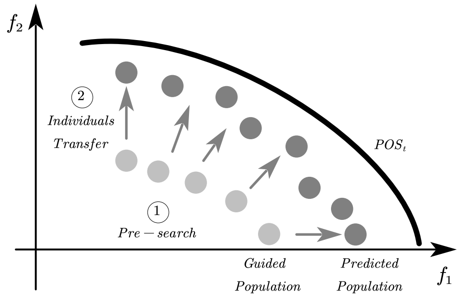


Fig. 1. Two-stage prediction process is as follows. In stage one, the guided population obtained by the presearch strategy can reduce the possibility of negative transfer. In stage two, individuals transfer reuses the information from a past environment and the guided population to produce the predicted population, and this population can be used by any population-based SMOA to obtain POS of the next moment.

dissimilar or the transfer method is not appropriate for the transfer task [10], [11]. Once negative transfer occurs, the prediction models may direct the search to the wrong direction, thus resulting in a sharp decrease in performance. Second, existing dynamic multiobjective optimization algorithms based on transfer learning often realize knowledge reuse by searching for a latent space, and this often means more parameter settings and consumes more computing resources.

To address these issues, a two-stage individual transfer-based dynamic multiobjective evolutionary algorithm (IT-DMOEA) is proposed in this article. The first stage of IT-DMOEA is an efficient presearch strategy, and the main purpose of the strategy is to reduce the possibility of negative transfer. The output of this step is a set of individuals, and we called the set as the guided population. In the second stage, a transfer learning method, transfer Adaboost (TrAdaboost) [12], is adapted to construct a predictive model. The model can produce a high-quality initial population, which is called the predicted population. Therefore, any population-based multiobjective optimization algorithm can be used to solve the DMOPs without any modification and can obtain great performance improvement. Fig. 1 illustrates the changes of a generated population due to the proposed method for a two-objective optimization problem (maximization).

The contributions of this work are as follows. First, the proposed method reduces the possibility of negative transfer by introducing a presearch strategy, and the addition of this process can make the subsequent transfer learning more effective. Second, compared with the existing methods, the proposed prediction model is essentially a sample-based classifier, which can avoid the need to set more parameters and consume many computing resources.

The remainder of this article is organized as follows. In Section II, we will first introduce the background knowledge of dynamic multiobjective optimization and then discuss the existing works in this field. In Section III, we will propose IT-DMOEA. In Section IV, we will present comparisons with five different algorithms under different indicators and all of the algorithms were tested on 14 well-known benchmark functions. In Section V, we will draw a summary of this article and outline the future research directions.

II. Preliminary Studies and Related Research

A. Dynamic Multiobjective Optimization

The mathematical form of DMOPs is as follows:

$$\textbf{Minimize} \;\; F(x,t) = \langle f_1(x,t), f_2(x,t), \ldots, f_M(x,t) \rangle$$
$$\text{s.t.} \;\; x \in \Omega$$

where $x = < x_1, x_2, \ldots, x_n >$ is the decision vector and t is the time or environment variable. $f_i(x,t) : \Omega \to \mathbb{R}(i = 1, \ldots, M)$. $\Omega = [L_1, U_1] \times [L_2, U_2] \times \cdots \times [L_n, U_n]$. $L_i, U_i \in \mathbb{R}$ are the lower and upper bounds of the ith decision variable, respectively. The aim of solving DMOPs is to find the set of solutions at different times so that all the objectives are as small as possible.

Definition 1 (Dynamic Decision Vector Domination): At time t, a decision vector x_1 Pareto dominates another vector x_2 denoted by $x_1 \succ_t x_2$, if and only if

$$\begin{cases} f_i(x_1,t) & \le f_i(x_2,t) \quad \forall i = 1, \ldots, m \\ f_i(x_1,t) & < f_i(x_2,t) \quad \exists i = 1, \ldots, m. \end{cases} \tag{1}$$

Definition 2 [Dynamic Pareto-Optimal Set (DPOS)]: If a decision vector x^* at time t satisfies

$$\text{DPOS} = \left\{x^* | \nexists x, x \succ_t x^*\right\} \tag{2}$$

then all x^* are called dynamic Pareto-optimal solutions, and the set of dynamic Pareto-optimal solutions is called the DPOS.

Definition 3 [Dynamic Pareto-Optimal Front (DPOF)]: DPOF is the corresponding objective vectors of the DPOS at time t, and

$$\text{DPOF} = \left\{y^* | y^* = F(x^*, t), x^* \in \text{DPOS}\right\}.$$

B. Related Work

In recent years, considerable progress has been made in the DMOP field, and most existing algorithms can be classified into the following categories: maintaining-diversity-based, prediction-based, and memory-based methods.

Diversity is one of the effective means to deal with dynamic change. Li *et al.* [13] introduced a novel strategy using special points (SPPSs) that can adaptively maintain the population diversity and resolve the problems at different degrees of difficulty. Wu *et al.* [14] and Ruan *et al.* [15] presented algorithms based on maintaining population diversity by using the position and direction of the center point of the optimal solution at the previous moment. They predicted the positions of some solutions at the next moment. These algorithms focused on reusing and predicting special points rather than many useless solutions, so they can track POF more accurately and consume less computing resources. Gong *et al.* [16] proposed a framework for decomposing decision variables through which all the decision variables are divided into two subpopulations according to the interval similarity between each decision variable and interval parameters. They adopted a strategy on the basis of change intensity and a random mutation strategy to track the POF, which is the accurate response of the environmental change. However, this strategy is computationally intensive to search for optimal solutions of a group that is weakly impacted by the changing interval parameter.

In addition to the aforementioned works, Azzouz *et al.* [17], Liu and Ding [18], and Jin *et al.* [19] used reference vectors to accelerate the convergence and maintain diversity. They exhibited that this strategy produces superior performance in dealing with DMOPs with nonlinear correlations between decision variables and severe environmental changes.

The memory-based mechanism enables evolutionary algorithms to record past information, and memory-based methods for DMOPs use additional storage to implicitly or explicitly store useful information from previous generations to guide future searches. Chen *et al.* [20] implemented a dynamic two-archive evolutionary algorithm that simultaneously maintains two co-evolving populations. The two populations are complementary to each other, for example, one deals with convergence while the other focuses on diversity. Xu *et al.* [21] proposed a memory-enhanced dynamic multiobjective evolutionary algorithm based on decomposition. The proposed algorithm simultaneously decomposed a DMOP into several dynamic scalar optimization subproblems and facilitated the co-evolvement. Then, a subproblem-based bunchy memory scheme, which stores good solutions from old environments and reuses them as necessary, is designed to respond to any environmental change.

Wang and Li [22] proposed a multistrategy dynamic multiobjective evolutionary algorithm, which uses a progeny generation mechanism based on adaptive genetics and differential operators to accelerate the convergence. When the environment changes, Gaussian mutation operators and similar memory strategies are used to reinitialize the population. Azzouz *et al.* [23] presented an adaptive hybrid population management strategy using memory, local search, and random strategies to effectively handle the dynamic environment in DMOPs. The special feature of this algorithm is that it can adjust the memory size and random solutions to be used according to the severity of change.

Peng *et al.* [24] employed an optimal solution set preservation mechanism to propose exploration and exploitation operators for predicting the new optimal solutions. Branke *et al.* [25] proposed a memory scheme, where the best individuals in the population are saved in an archive, and the saved individuals can be retrieved and returned to the population when the algorithm detects a change.

In general, a memory-based mechanism is suitable for periodic optima problems [26]. In the memory-based mechanism, some individuals may fall in the vicinity of the new optimal solution, but they are out of the memory, so memory-based approaches become inappropriate when there is noncyclic stochastic dynamism.

A prediction model can be constructed by reusing past information. Cao *et al.* [27] focused on dealing nonlinear correlations between solutions obtained in sequential time periods. The proposed MOEA/D-SVR maps the solutions into a high-dimensional space and a support vector regression (SVR) was employed to implement this core idea to construct a predictor. However, the performance of the predictor is depended heavily on the quality of past solutions. Rong *et al.* [28] presented a multidirectional prediction (MDP) strategy to enhance the performance of evolutionary algorithms in DMOPs. The population is clustered into several representative groups by a proposed classification strategy, where the number of clusters is adapted according to the intensity of the environmental change. The clusters are used to accurately predict the moving location of POS. However, MDP only considered the situations when two consecutive changes are similar and linear in the decision space. Muruganantham *et al.* [6] proposed a predictive model based on the Kalman filter to minimize the influence of measurement noise. When a change in the environment is detected, the Kalman filter is used to guide the search for a new POS to generate a new initial population, and then the optimal population at this moment is determined based on a decomposition-based differential evolution algorithm. However, the proposed MOEA/D-KF lacks strategies for diversity maintenance and direction-guided search.

Zhou *et al.* [29] presented an algorithm called population prediction strategy (PPS) to predict an entire population instead of predicting some isolated points. When change is detected, the algorithm uses a sequence of center points obtained from the search progress to predict the next center point, and the previous manifolds are used to estimate the next manifold. The autoregressive model used in PPS assumes that the linkage of solutions between environments is linear. Similarly, Ma *et al.* [30] and Zhou *et al.* [31] utilized a simple linear model to generate the population in the next environment. Therefore, these linear models might perform bad if the linkage is nonlinear.

In addition, Woldesenbet and Yen [26], Wang *et al.* [32], and Liu *et al.* [33] used variable relocation to adapt individuals to the changing environment. The relocation occurs during the evolutionary process, and the algorithm reuses information of evolutionary history, but these proposed algorithms often exert considerable computation. Xu *et al.* [34], Nguyen *et al.* [35], and Helbig and Engelbrecht [36] adopted the approach of coevolution. However, grouping decision variables for optimization may divide related variables into different groups, which results in a degrading performance.

It is worth noting that the distribution of the solutions of DMOPs usually varies at different times or environments, which means that the solutions at different times are non-independently identically distributed (non-IID). A framework based on transfer learning, Tr-DMOEA [8], was proposed to predict an effective initial population for solving DMOPs. The Tr-DMOEA algorithm can be divided into the following major steps. First, an initial population is randomly generated at the very beginning, and the optimization problem can be solved by a static multiobjective optimization algorithm (SMOA).[1] When an environment change is detected, Tr-DMOEA finds a latent space via the transfer learning method, and the distribution of solutions at different times will be as similar as possible in this latent space. Next, Tr-DMOEA maps the previously obtained POF into the latent space and then uses the mapped solutions to construct a population to search for the POF at a new moment.

[1] In this article, SMOA refers to any population-based multiobjective optimization algorithm.

At the same time, the concept of utilizing acquired knowledge has been proved to be promising [37]. Min *et al.* [38] demonstrated a transfer stacking of Gaussian process surrogate models by incorporating knowledge transfer. The adaptive knowledge gained from expensive problem-solving experiences is introduced to augment the optimization process of a target task. Therefore, the search in computationally expensive settings is accelerated. Bali *et al.* [39] proposed an online transfer parameter estimation scheme for the multitasking evolutionary algorithm. The proposal can capture the similarities between tasks in a pure data-driven way so that the extent of knowledge transfer can be adapted based on the optimal mixture of probabilistic models. Da *et al.* [40] designed a principled realization of black-box transfer optimization. The scheme exploits the similarities between the source and the target and reveals latent synergies during optimization. When faced with multiple sources, the positive and negative transfer are automatically distinguished, and the negative transfer is curbed, thereby the excellent performance is kept. In [41], to maintain the diversity during population transfer, a domain adaptation-based transfer population method incorporated with nonparametric density estimation and importance sampling is designed in solving DMOPs. However, these knowledge transfer-based dynamic multiobjective optimization algorithms do not consider the distribution characteristics of the previous and next solutions, which may lead to negative transfer, local optimum, or poor diversity.

C. TrAdaboost

In this article, we adapted a method called TrAdaboost [12] to accommodate the needs of DMOPs. The basic idea of TrAdaboost is to filter out dissimilar data in the source domain to those in the target domain via the boosting technique.

TrAdaboost is an ensemble method, and each weak classifier maps a sample X to a label $Y \in \{0, 1\}$, where $X = X_{\text{source}} \cup X_{\text{target}}$. The outputs of these weak classifiers are combined into a strong classifier. During the training process, the TrAdaboost increases the weights of the misclassified instances from the target domain and decreases the weights of the misclassified instances from the source domain. In this way, the TrAdaboost makes the full use of source instances that are most similar to the target data and ignores those that are dissimilar. Through continuous learning, the TrAdaboost algorithm can obtain a more accurate classifier for the target domain samples.

III. Proposed Algorithm

The details of the proposed algorithm, IT-DMOEA, are presented in this section. Briefly, the IT-DMOEA consists of two subroutines: 1) presearch and 2) individuals transfer. The main purpose of the presearch procedure is to reduce the possibility of negative transfer, and the output of this procedure is a set of solutions and we called the set as a guided population. In the individuals' transfer stage, the guided population and other solutions are fed into a transfer learning module, TrAdaboost, to obtain a classifier, which comprises several weak classifiers, and this classifier can be used to generate an initial population.

A. Presearch

The output of the presearch stage is the guided population. In the proposed method, the main purpose of the guided population is to reduce the possibility of negative transfer. Therefore, to a certain extent, the quality of a population directly affects the subsequent steps, especially the quality of the initial population.

Diversity plays a key factor in population quality. The reason is that we believe that with the increase of the diversity of solutions, the aggregation of inferior solutions can be alleviated, so possibility of negative transfer can be effectively reduced. In the presearch procedure, the uniform reference vectors [42] are used to maintain the diversity of the guided population, and the definition of uniform reference vectors R_i $(i = 1, \ldots, H)$ is as follows:

$$\begin{cases} R_i = \left(r_i^1, r_i^2, \ldots, r_i^m\right) \\ r_i^j \in \left\{\frac{0}{p}, \frac{1}{p}, \ldots, \frac{p}{p}\right\}, \sum_{j=1}^m r_i^j = 1 \end{cases} \quad (3)$$

and the number of reference vectors H can be calculated by $H = \binom{m+p-1}{p}$. p is a predefined positive integer.

For each generated uniform reference vector R_i, we can use the penalty boundary intersection (PBI) approach [43] to assign a value $g(x|R_i)$ to each individual x_i. In essence, the PBI method is to decompose a multiobjective optimization problem, $F(x) = (f_1(x), \ldots, f_m(x))$, into a vector-based single-objective optimization problem, and its definition is as follows:

$$\begin{cases} \textbf{minimize } g(x|R_i) = d_1(x|R_i) + \theta d_2(x|R_i) \\ \text{s.t. } x \in \Omega \end{cases} \quad (4)$$

where

$$d_1(x|R_i) = \frac{\left\|(F(x) - z^*)^T R_i\right\|}{\|R_i\|} \quad (5)$$

$$d_2(x|R_i) = \left\|F(x) - \left(z^* + d_1(x|R_i) \times \frac{R_i}{\|R_i\|}\right)\right\| \quad (6)$$

where $z^* = (\textbf{Min}f_1(\cdot), \ldots, \textbf{Min}f_m(\cdot))$ is the ideal point. Fig. 2 depicts how to select two individuals for every reference vector according to the PBI method.

The details of the presearch are shown in Procedure Presearch. For each given reference vector R_i, we can sort all individuals according to the $g(x|R_i)$ value and select two individuals with the minimum value. After that, all these selected individuals constitute the population P. Then, we crossover the four individuals of two adjacent vectors successively to obtain a population Q. Next, a Gauss mutation is performed on all individuals in population P to generate population P'. Finally, for every reference vector R_i, two solutions are picked out from $P' \cup Q$ to generate a population. These processes are repeated until the stop condition is satisfied. The stop condition is determined by *Coverage* metric, and the condition is defined as follows:

$$\text{Coverage}(P_i, P_{i-1}) = \frac{|\{a \in P_i | \nexists b \in P_{i-1}, b \succ a\}|}{|P_i|} < \sigma \quad (7)$$

where P_i and P_{i-1} are populations of the ith generation and the $i-1$th generation during the presearch, respectively. In this research, we set $\sigma = 0.9$. We call the population as the guided population.

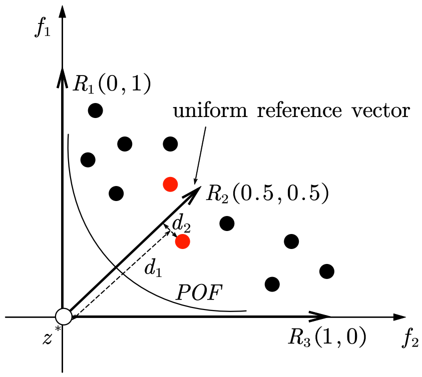


Fig. 2. In the case that $p = 2$ and $H = 3$, we have three uniform reference vectors, and for each uniform reference vector, the presearch procedure will select two individuals by using the PBI method. The two red points are the selected individuals for the reference vector in the middle.

B. Individuals Transfer

Once the guided population is generated, the individuals' transfer stage will start. In this stage, the basic idea of the TrAdaboost method is to train a strong classifier h_f by using the guided population and solutions obtained from the past one time. Once the training is complete, the classifier h_f can identify solutions that are randomly generated in the current environment, and we can pick out individuals identified as "good" by the classifier to form an initial population. Fig. 3 describes the individuals transfer procedure of IT-DMOEA.

During the transfer procedure, the guided population and the solutions obtained in a past environment are regarded as the target domain D_{ta} and the source domain D_{so}, respectively. These solutions are then labeled by $c(x) : x \in D_{ta} \cup D_{so} \to y$, $y \in \{+1, -1\}$. For each domain, the nondominated individuals are labeled as $+1$, and the dominated individuals are labeled as -1.

A base learning algorithm L is used iteratively to adjust the parameters and weights of a set of weak classifiers h_p, and then a strong classifier h_f will be obtained by combining classifiers h_p with different weights. During this process, if a training individual from the source domain is misclassified by h_p, and this individual may be quite different from the target-domain solutions. Therefore, its training weight should be reduced. On the other hand, if a training individual from the target domain is misclassified by h_p, the weight of this individual should be increased so that it can play a greater role in the future training process.

In the jth iteration ($j = 1, \ldots, N$), the error rate of the weak classifier h_p^j on D_{ta} is calculated as follows:

$$\epsilon_j = \sum_{x \in D_{ta}} p_j(x) \cdot \ell_j(x) \tag{8}$$

where

$$p_j(x) = \frac{w_j(x)}{\sum_{x \in D_{ta}} w_j(x)} \tag{9}$$

and $\ell_j(x)$ is the loss function

$$\ell_j(x) = \left| h_p^j(x) - c(x) \right| \tag{10}$$

Procedure(PRESEARCH)

Input: A MOP $F_t(\cdot)$
Output: The guided population GP
Initialize the random population P;
Reference vectors R are generated by Equation (3);
for R_i *in* R **do**
  for x *in* P **do**
    Compute $g(x|R_i)$ for x according to R_i;
  end
  The two individuals $\{x\}$ with the minimum $g(x|R_i)$ are assigned to R_i;
  $P = P - \{x\}$;
end
$P = \{x | x \textit{ is assigned to } R_i, R_i \in R\}$;
while *Coverage* $\geq \sigma$ **do**
  Mate individuals on neighboring reference vector R_i to obtain Q;
  Mutate individuals to produce P';
  $P = P' \cup Q$;
  for R_i *in* R **do**
    for x *in* P **do**
      Compute $g(x|R_i)$ for x according to R_i;
    end
    The two individuals $\{x\}$ with the minimum $g(x|R_i)$ are assigned to R_i;
    $P = P - \{x\}$;
  end
  $P = \{x | x \textit{ is assigned to } R_i, R_i \in R\}$;
end
$GP = P$;
return GP

and the coefficient α_j of the weak classifier h_p is calculated as

$$\alpha_j = \frac{1}{2} \ln \frac{1 - \epsilon_j}{\epsilon_j}. \tag{11}$$

When ϵ_j is small, α_j becomes large, and the weak classifier h_p^j with a large α_j will have a substantial influence on the changing weights of the individuals, and the weights are updated as follows:

$$w_{j+1}(x) = \begin{cases} w_j(x) \cdot e^{\alpha \cdot \left| h_p^j(x) - c(x) \right|}, & x \in D_{so} \\ w_j(x) \cdot e^{\alpha_j \cdot \left| h_p^j(x) - c(x) \right|}, & x \in D_{ta} \end{cases} \tag{12}$$

where

$$\alpha = \frac{1}{2} \ln\left(\frac{1}{1 + \sqrt{2 \ln N}} \right). \tag{13}$$

w_{j+1} will be inputted into the next weak classifier h_p^{j+1}. In this way, the weights of individuals that are more similar to the solution of the target domain will gradually increase, and vice versa, and this also means that these weak classifiers, $h_p = h_p^1, h_p^2, \ldots,$, will gradually become more accurate in recognizing the solution of the target domain.

When the stop condition is satisfied, we can use these weak classifiers to construct a strong classifier in different ways. In

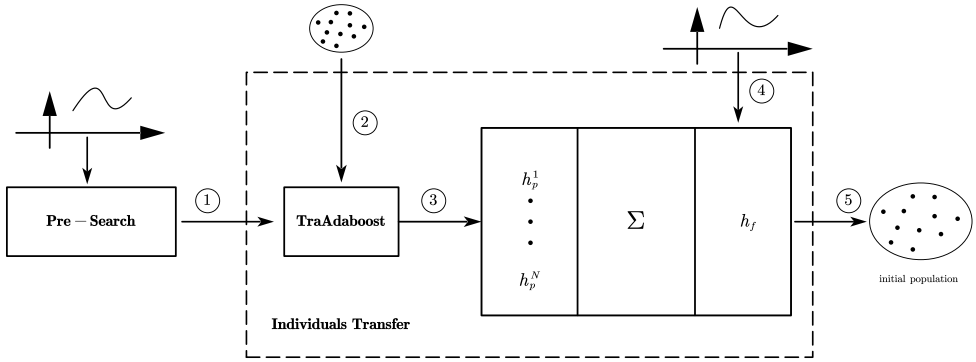


Fig. 3. Schematic of procedure individuals transfer. Steps 1 and 2: the output of the presearch procedure and the solutions obtained in the past one time (not only nondominated solutions) are fed into the TrAdaboost algorithm, respectively. Step 3: the TrAdaboost obtains some weak classifiers h_p and then obtains a strong classifier h_f. Step 4: a number of individuals of the current time will be generated and be input them into h_f. Step 5: those individuals identified as "good" by the classifier h_f will form the initial population.

Procedure(INDIVIDUALS TRANSFER)

Input: The two labeled data sets D_{so} and D_{ta};
Output: The initial population *initPop*

Set a based classifier L and the number of iterations N;
Initialize the weight vector $w_1(x) = \begin{cases} \frac{1}{|D_{so}|}, x \in X_{so}. \\ \frac{1}{|D_{ta}|}, x \in X_{ta}. \end{cases}$
for $j = 1$ *to* N **do**
  Call the classifier L with $D_{ta} \cup D_{so}$ and p_j to get h_p^j;
  Calculate ϵ_j of h_p^j according to (8);
  Calculate α_j according to (11);
  Update the weight vector according to (12);
end
Get h_f according to (14);
Sample solutions x_{test} at the current environment;
return $initPop = \{x | h_f(x) = +1, x \in x_{\text{test}}\}$

Algorithm 1: IT-DMOEA

Input: The Dynamic Optimization problem $F_t(x)$, a SMOA **SMOA** ;
Output: the POS of the $F_t(x)$ at the different moments.;

Initialization;
$POS_t = \textbf{SMOA}(F_t(x))$;
Generate randomly dominated solutions $P_t^{'}$;
while *the environment has changed* **do**
  $t = t + 1$;
  $D_{so}^{t-1} = POS_{t-1} \cup P_{t-1}^{'}$;
  $D_{ta}^{t} = \textbf{PreSearch}(F_t(x))$;
  $iPop = \textbf{Individuals Transfer}(D_{so}^{t-1} \cup D_{ta}^{t})$;
  $POS_t = \textbf{SMOA}(iPop, F_t(x))$;
  Generate randomly dominated solutions $P_t^{'}$;
  return POS_t
end

our research, we used the following equation to construct a strong classifier h_f:

$$h_f(x) = \textbf{sign}\left(\sum_{j=1}^{N} \alpha_j h_p^j(x)\right). \tag{14}$$

After obtaining the strong classifier h_f, the proposed algorithm IT-DMOEA generates a large number of solutions x_{test} at the current moment. These generated solutions are inputted into the classifier, and those individuals identified as good by the classifier will be selected to form the initial population. In this work, we sample $100 \cdot S$ solutions as test solutions x_{test}, where S is the population size. Procedure INDIVIDUALS TRANSFER describes the individuals' transfer stage.

Algorithm 1 describes the main program of the proposed IT-DMOEA method.

C. Complexity Analysis

In the static presearch stage, assigning individuals to reference vectors requires $O(H \cdot S)$, where S is the population size and H is the number of reference vectors. Mating and mutating operations spend $O(H)$ and $O(S)$, respectively. Suppose there are I_p iterations in the presearch stage, the complexity of the presearch stage of the IT-DMOEA algorithm is $O(I_p \cdot H \cdot S) + O(I_p \cdot H) + O(I_p \cdot S) = O(I_p \cdot H \cdot S)$.

Suppose we use the support vector machine (SVM) [44] as the base classifier, the nondominated sorting algorithm is used to label the individuals, then the proposed method needs $O(m \cdot S^2)$ time to obtain a labeled training set, where m is the number of objectives. The individuals' transfer module spends $O(N \cdot S^2 \cdot d)$ steps to obtain the strong classifier h_f by using the SVM classifier, where d is the number of decision variables and N is the number of iterations for individuals transfer. We sample $100 \cdot S$ individuals from decision space as test population, and the complexity of this stage is $O(S \cdot d)$. The proposed algorithm requires $O(S^2 \cdot d)$ steps to determine whether an individual in the test population can join the initial population by using h_f. Hence, the individuals transfer part requires $O(m \cdot S^2) + O(N \cdot S^2 \cdot d) + O(S \cdot d) + O(S^2 \cdot d) = O(N \cdot S^2 \cdot d)$.

IV. EXPERIMENTS

A. Test Problems and Performance Indicators

All compared algorithms are evaluated based on 20 DMOPs selected from two benchmark suites: 1) DF functions[45], which are from CEC 2018 DMO benchmarks and 2) F functions [29]. The DF benchmark suite comprises of 14 problems (DF1–DF14), and the F benchmark suite contains six problems (F5–F10). The dynamics of a DMOP is governed by $t = (1/n_t)\lfloor(\tau/\tau_t)\rfloor$, where τ, n_t, and τ_t refer to the maximum generation counter, the severity of change, and the frequency of change, respectively, so there will be $\lfloor\tau/\tau_t\rfloor$ environment changes. A low τ_t value implies a rapidly changing environment while a low n_t value denotes severely changing environment.

In this study, the following metrics are utilized to evaluate the performance of various algorithms.

1) Inverted Generational Distance (IGD): The IGD is a metric for quantifying the convergence of the solutions obtained by a multiobjective optimization algorithm. When the IGD value is small, the convergence of the solution is improved. IGD is defined as

$$\mathrm{IGD}\big(\mathrm{POF}^*, \mathrm{POF}\big) = \frac{1}{|\mathrm{POF}^*|}\sum_{p^*\in\mathrm{POF}^*}\min_{p\in\mathrm{POF}}\left\|p^* - p\right\|^2 \quad (15)$$

where POF^* is the true POF of a multiobjective optimization problem, and POF is an approximation set of POF obtained by a multiobjective optimization algorithm.

The MIGD metric is a variant of IGD and is defined as the average of IGD values in some time steps over a run

$$\mathrm{MIGD}\big(\mathrm{POF}_t^*, \mathrm{POF}_t\big) = \frac{1}{|T|}\sum_{t\in T}\mathrm{IGD}\big(\mathrm{POF}_t^*, \mathrm{POF}_t\big) \quad (16)$$

where T is a set of discrete-time points in a run and $|T|$ is the cardinality of T.

2) Maximum Spread (MS): Measuring the extent of obtained solutions covers the true POF. A large MS value facilitates additional coverage for the real POF by the solution obtained by the algorithm. MS is calculated as follows:

$$MS = \sqrt{\frac{1}{M}\sum_{k=1}^{M}\left[\frac{\min\left[F_k^{\max}, f_k^{\max}\right] - \max\left[F_k^{\min}, f_k^{\min}\right]}{F_k^{\max} - F_k^{\min}}\right]^2} \quad (17)$$

where $F_k^{\max}$ and $F_k^{\min}$ represent the maximum and minimum of the kth objective in true POF, respectively; and $f_k^{\max}$ and $f_k^{\min}$ represent the maximum and minimum of the kth objective in the obtained POF, respectively. This metric is similarly modified to the IGD to act as a performance metric for evaluating dynamic MOEAs.

B. Compared Algorithms and Parameter Settings

The proposed IT-DMOEA algorithm is compared against several popular dynamic MOEAs, including MDP [28], MOEA/D-SVR [27], Tr-DMOEA [8], MOEA/D-KF [6], SGEA [46], and PPS [29]. For a fair comparison, most of the parameters of these algorithms are set according to the original references. Other parameters are set as follows.

1) *Population Size and Number of Decision Variables:* In the comparison, the population size is set to 100 and 200 for biobjective and triobjective problems, respectively. Each benchmark problem has ten variables.
2) *Dynamic Test Settings:* We run each test function ten times independently. In the following experiments, the severity of change is set as $n_t = 10, 5$, and the frequency of change is set as $\tau_t = 5, 10$. Therefore, these values will comprise three pairs of change characteristics, ($n_t = 10, \tau_t = 10$), ($n_t = 10, \tau_t = 5$), and ($n_t = 5, \tau_t = 10$), for each problem. To minimize the effect of static optimization, we gave 30 generations for each algorithm before the first change occurs. The total number of generations τ is fixed to be $30+50*\tau_t$, which ensures there are 50 changes in each run.
3) *Parameters in IT-DMOEA:* For generating reference vectors, the parameter p in (3) is set to 24 and 9 for bi- and tri-objective problems, respectively. The penalty parameter θ in (4) is set to $\frac{1}{6}$ and the parameter σ of the stop condition is set to 0.9. In the individuals' transfer stage, most of the parameters in the SVM are set by default [44]. The number of iterations N for individuals transfer is set to 5. We choose RM-MEDA [47] as the SMOA optimizer for IT-DMOEA.

C. Negative Transfer

One of our motivations is to minimize the influence of negative transfer when using transfer learning. In the field of machine learning, the negative transfer usually means "transferring knowledge from the source can have a negative impact on the target learner" [10], [11]. We extend this idea to the field of evolutionary optimization, which means that a transfer learning method prevents the optimization algorithm from using previous experience to obtain better solutions. At the same time, we believe that individuals with high quality will reduce the occurrence of negative transfer.

To validate our idea, we compared two algorithms, IT-RM-MEDA$_{np}$ (IT-NSGA-II$_{np}$) and IT-RM-MEDA (IT-NSGA-II), where the former is the algorithm without using the pre-search strategy and transfers the population of the last one environment directly; and the latter is the one with the pre-search strategy. The statistical results of MIGD are shown in Table I and Table I (supplementary material), respectively. In the tables, (+), ((=), and (−)) indicate that compared algorithms are statistically significantly better (insignificant and significantly worse) than RM-MEDA (NSGA-II)[2] using the Wilcoxon rank-sum test [48], and all of them are at the 5% significance level.

Table I shows that IT-RM-MEDA achieves 32 out of 40 best results. Table I (supplementary material) shows that IT-NSGA-II achieves 26 out of 40 best results. At the same time, we also want to point out that among 40 test cases, IT-RM-MEDA$_{np}$ performed worse than the RM-MEDA algorithm, an algorithm without transfer learning, in 15 instances, and this means that at least about 37% of the cases of the transfer learning-based algorithm does not have the positive effect as we expected.

Based on our understanding of negative transfer, we propose a novel measure, negative IGD (NIGD), to evaluate the degree

[2]The initial population used by these algorithms is randomly generated.

TABLE I
MEAN AND STANDARD DEVIATION VALUES OF MIGD METRIC OBTAINED BY RM-MEDA, TR-RM-MEDA, IT-RM-MEDA$_{np}$, AND IT-RM-MEDA

Problems	τ_t, n_t	RM-MEDA	Tr-RM-MEDA	IT-RM-MEDA$_{np}$	IT-RM-MEDA
DF1	10,10	0.0930±6.53E-3	0.1811±1.59E-2(-)	0.2937±2.75E-5(-)	**0.0246±7.33E-5(+)**
	10,1	0.2388±1.04E-2	0.1806±3.06E-2(+)	0.3133±1.96E-2(-)	**0.0977±7.89E-3(+)**
DF2	10,10	0.0608±2.57E-3	0.1208±8.19E-3(-)	0.0071±1.94E-5(+)	**0.0062±6.16E-5(+)**
	10,1	0.1550±8.21E-2	0.1236±5.51E-3(+)	0.1622±1.71E-3(-)	**0.0063±2.40E-3(+)**
DF3	10,10	0.3449±5.95E-2	0.2702±2.28E-2(+)	0.0221±2.31E-2(+)	**0.0179±3.09E-4(+)**
	10,1	0.3860±1.99E-1	**0.2743±7.53E-2(+)**	0.7816±1.75E-1(-)	0.6750±2.00E-2(-)
DF4	10,10	1.1713±2.80E-1	1.1675±3.54E-1(=)	0.8274±5.01E-2(+)	**0.7791±3.15E-1(+)**
	10,1	1.2229±6.57E-2	**0.8382±8.22E-2(+)**	1.5191±2.70E-2(-)	1.5258±2.46E-1(-)
DF5	10,10	1.9117±8.32E+0	2.8491±1.96E+1(-)	0.0137±4.97E-3(+)	**0.0067±4.77E-5(+)**
	10,1	2.2794±1.38E+1	2.2900±1.40E+1(=)	0.0511±7.89E-3(+)	**0.0198±1.95E-3(+)**
DF6	10,10	3.9898±3.44E+0	5.9398±1.63E+1(-)	0.7257±4.52E-1(+)	**0.6382±5.06E-2(+)**
	10,1	30.7860±3.33E+2	29.0123±2.88E+2(=)	1.2758±1.82E-1(+)	**1.2379±1.80E-1(+)**
DF7	10,10	4.0805±3.46E+0	3.7439±1.39E+0(+)	0.7259±4.41E-1(+)	**0.6589±6.37E-3(+)**
	10,1	15.9909±3.75E+2	7.0123±3.08E+0(+)	2.0883±1.42E+0(+)	**1.4991±1.93E-1(+)**
DF8	10,10	0.7109±1.66E-1	0.8227±2.32E-1(-)	0.0833±9.62E-3(+)	**0.0559±2.22E-2(+)**
	10,1	0.4592±1.60E-1	0.6123±2.45E-1(-)	0.0299±1.22E-3(+)	**0.0270±1.95E-3(+)**
DF9	10,10	**2.1170±5.59E+0**	2.5579±5.273E+0(-)	2.5678±7.28E+0(-)	3.1016±2.68E+0(-)
	10,1	3.1198±6.21E+0	2.0288±4.89E+0(+)	3.7871±2.01E+0(-)	**1.4630±1.52E+0(+)**
DF10	10,10	0.1308±4.13E-3	0.1093±2.56E-3(+)	0.0444±5.66E-3(+)	**0.0313±1.96E-3(+)**
	10,1	0.4461±6.99E-2	**0.1108±2.30E-3(+)**	0.5796±7.05E-3(-)	0.2033±8.70E-3(+)
DF11	10,10	0.1513±9.00E-3	0.2866±2.43E-2(-)	0.0856±7.39E-3(+)	**0.0506±5.12E-3(+)**
	10,1	0.8306±9.86E-2	**0.3122±1.43E-2(+)**	1.0945±2.01E+0(-)	1.0909±5.81E+0(-)
DF12	10,10	**0.6053±1.74E-2**	1.1933±1.64E-5(-)	1.0300±1.81E+0(-)	1.1903±2.09E+0(-)
	10,1	**0.6266±1.61E-2**	1.1962±1.18E-5(-)	0.7153±1.25E-1(-)	0.9568±1.50E-1(-)
DF13	10,10	2.0295±8.67E+0	2.1710±9.05E+0(-)	0.0690±3.15E-2(+)	**0.0681±1.86E-3(+)**
	10,1	2.2440±9.13E+0	2.2730±1.26E+1(-)	**0.1536±3.15E-3(+)**	0.1653±1.25E-3(+)
DF14	10,10	1.3740±4.11E+0	1.6629±4.98E+0(-)	0.0210±1.20E-3(+)	**0.0152±2.44E-3(+)**
	10,1	1.6315±5.28E+0	1.4387±5.56E+0(+)	0.0895±4.97E-3(+)	**0.0385±1.71E-3(+)**
F5	10,10	2.7930±2.48E+0	2.6642±1.76E+0(=)	0.5717±3.15E-1(+)	**0.1006±7.89E-2(+)**
	10,1	0.9602±4.98E-2	1.0652±4.80E+0(-)	1.2107±2.88E+0(+)	**0.0941±1.96E-2(+)**
F6	10,10	1.1618±4.74E-1	1.2206±5.83E-1(-)	1.4515±4.51E-1(-)	**0.2801±5.82E-2(+)**
	10,1	1.8515±1.46E+0	1.9353±2.03E-1(-)	0.2529±5.03-2(+)	**0.1088±9.21E-2(+)**
F7	10,10	1.2632±4.45E-1	1.4381±5.34E-1(-)	0.5212±1.13E-2(+)	**0.2455±7.15E-2(+)**
	10,1	1.6198±3.39E-1	1.7884±4.33E-1(-)	2.0712±5.12E-2(-)	**0.3288±1.55E-1(+)**
F8	10,10	0.8285±4.05E-2	0.7482±8.32E-2(+)	0.4582±3.15E-1(+)	**0.3236±7.82E-2(+)**
	10,1	0.8523±5.68E-2	0.7102±1.84E-2(+)	0.9047±4.51E-1(-)	**0.2969±5.01E-2(+)**
F9	10,10	1.5000±5.57E-1	1.4281±5.27E-1(+)	0.3738±1.54E-2(+)	**0.2056±1.20E-3(+)**
	10,1	0.6717±1.67E-2	0.7522±2.62E-2(-)	0.8421±2.83E-2(-)	**0.0985±3.75E-3(+)**
F10	10,10	3.9045±1.28E+1	2.5320±2.56E+0(+)	0.2166±1.35E-2(+)	**0.1204±1.96E-3(+)**
	10,1	0.7190±1.94E-2	0.7583±1.97E-2(-)	0.2759±2.88E-2(+)	**0.1098±1.20E-3(+)**
+/=/-			16/4/20	25/0/15	34/0/6

of negative transfer

$$
\begin{aligned}
\mathrm{NIGD} &= \sum_{t \in T} \left(\mathrm{IGD}_t^{tr} - \mathrm{IGD}_t^{rnd} \right) \\
\text{s.t.} \quad & \mathrm{IGD}_t^{tr} > \mathrm{IGD}_t^{rnd}
\end{aligned} \tag{18}
$$

where

$$\mathrm{IGD}_t^{rnd} = \mathrm{IGD}\left(\mathrm{POF}_t^{rnd}, \mathrm{POF}_t^{*}\right) \tag{19}$$

$$\mathrm{IGD}_t^{tr} = \mathrm{IGD}\left(\mathrm{POF}_t^{tr}, \mathrm{POF}_t^{*}\right). \tag{20}$$

POF_t^{rnd}, POF_t^{tr}, and POF_t^{*} are the POFs obtained by the random population, the transfer population, and the true POF at time t.

The idea behind this definition is that if the quality of the solutions obtained by using the transfer learning method is worse than that obtained by using random population, then we consider that transfer learning has a negative effect on the optimization process. The greater the value of NIGD, the greater the degree of negative transfer. As can be observed from Fig. 4, in most cases, the more drastic the change in the function (the smaller the n_t value), the greater the value of NIGD, which means that the greater the degree of negative transfer. At the same time, it is not hard to see that IT-RM-MEDA tends to have the smallest NIGD value, suggesting that it is more capable of preventing negative transfer.

These results confirm our conjecture that the more drastic the dynamic optimization function changes, the more likely it is that negative transfer will occur. An intuitive explanation is that the distribution of solutions varies greatly in the dramatic changing environment, which makes the source domain and the target domain extremely dissimilar.

In addition, the plots of average IGD values of the algorithms are shown in Fig. 5 and Fig. 1 (supplementary material), respectively. We can find that IT-RM-MEDA (IT-NSGA-II) not only can produce smaller IGD values but also have faster convergence speed than other algorithms on most of the benchmark functions.

D. Performance on DF and F Problems

The statistical results of the MIGD and MS values over ten runs can be found in Tables II and III, respectively. In the tables, (+), ((=), and (-)) indicate that IT-RM-MEDA is statistically significantly better (insignificant and significantly worse) than compared algorithms using the Wilcoxon rank-sum test, and all of them are at the 5% significance level. As can be seen from Table II, IT-RM-MEDA achieves 29 out of 60 best results, MOEA/D-SVR has 13 best results, MDP has 9 best results, and SGEA and MOEA/D-KF achieve 5 and 3 best results for IGD values, respectively. PPS only achieves 1 best result. To be specific, IT-RM-MEDA performs better on DF2, DF3, DF5, DF10, DF13, and DF14 under all dynamic test settings. MOEA/D-SVR achieves better convergence over POF on DF1, DF4, DF6, DF7, DF9, and DF11. For SGEA and MOEA/D-KF, they can obtain well-converged solutions for DF8 and F8 problems, respectively. In other cases, IT-RM-MEDA is a little worse than the corresponding best-performing algorithms.

Table III shows the MS values of the seven compared algorithms. It is obvious that the proposed IT-RM-MEDA exhibits a better performance than the other six compared algorithms for uniformity. IT-RM-MEDA performs the best on 35 out of 60 test instances, which is followed by MDP gaining 13 best results, Tr-RM-MEDA and SGEA gaining 6 and 5 best results, respectively, and MOEA/D-SVR gaining 1 best result. Specifically, IT-RM-MEDA performs better on DF1, DF2, DF3, DF10, DF13, F5, F9, and F10 under all dynamic test settings. For the majority of the other testing instances, IT-RM-MEDA slightly falls behind the corresponding best-performing algorithms.

The above experimental results suggest that IT-DMOEA can obtain a set of solutions with good convergence and diversity for most of the test problems. However, for some benchmark functions, such as DF4, DF6, DF8, DF9, and DF12, IT-DMOEA leaves some room to be desired, and the reason may be due to the POS of these problems changes dramatically, so the presearch stage is very difficult to find good candidates.

What we want to point out is that IT-RM-MEDA can achieve good performance on the most cases of the F benchmark suites, except on F8, F9, and F10. The characteristics of the three benchmark functions are that the POS jumps from one area to another area and the geometric shapes of

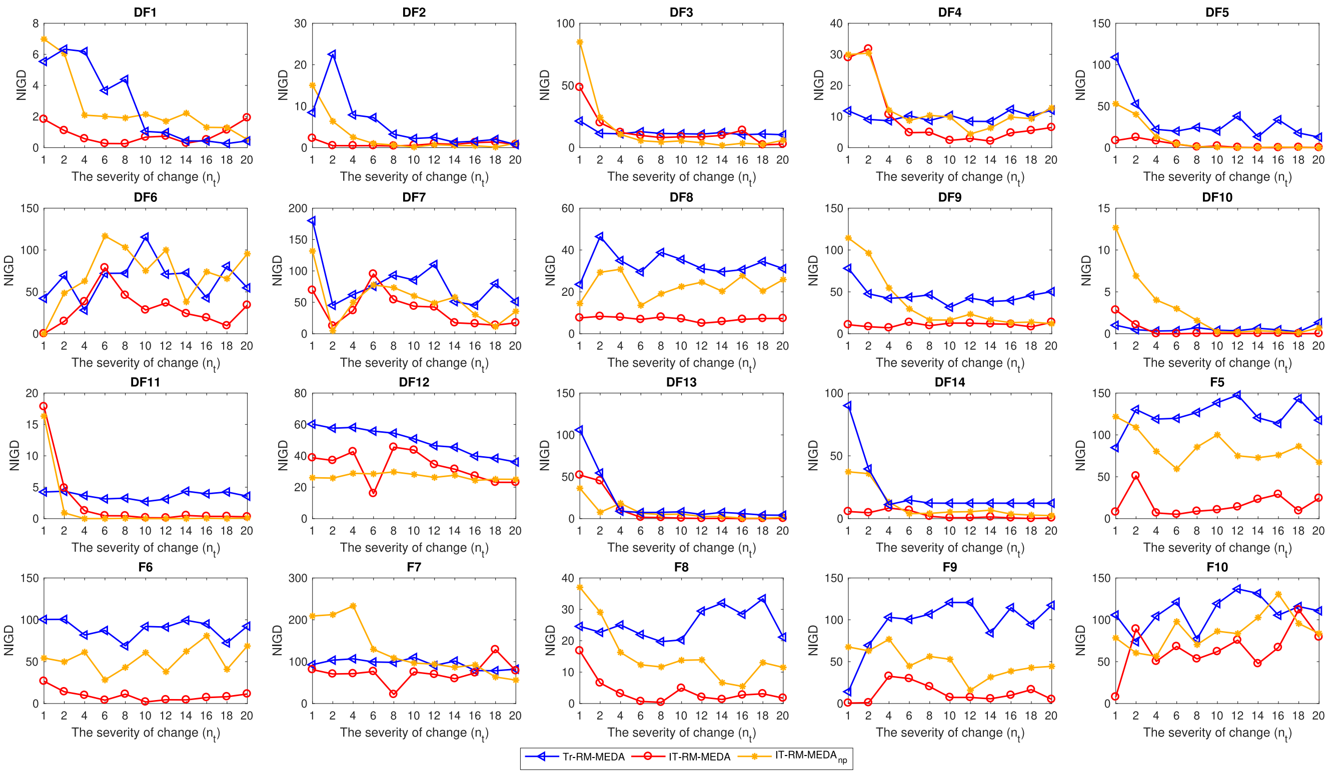


Fig. 4. Relationship between n_t and NIGD of initial populations obtained by Tr-RM-MEDA, IT-RM-MEDA$_{np}$ and IT-RM-MEDA, respectively, when the frequency $\tau_t = 10$. The smaller the value of the severity of change, the more dramatic the change in the optimization function; the larger the value of NIGD, the greater the degree of negative transfer.

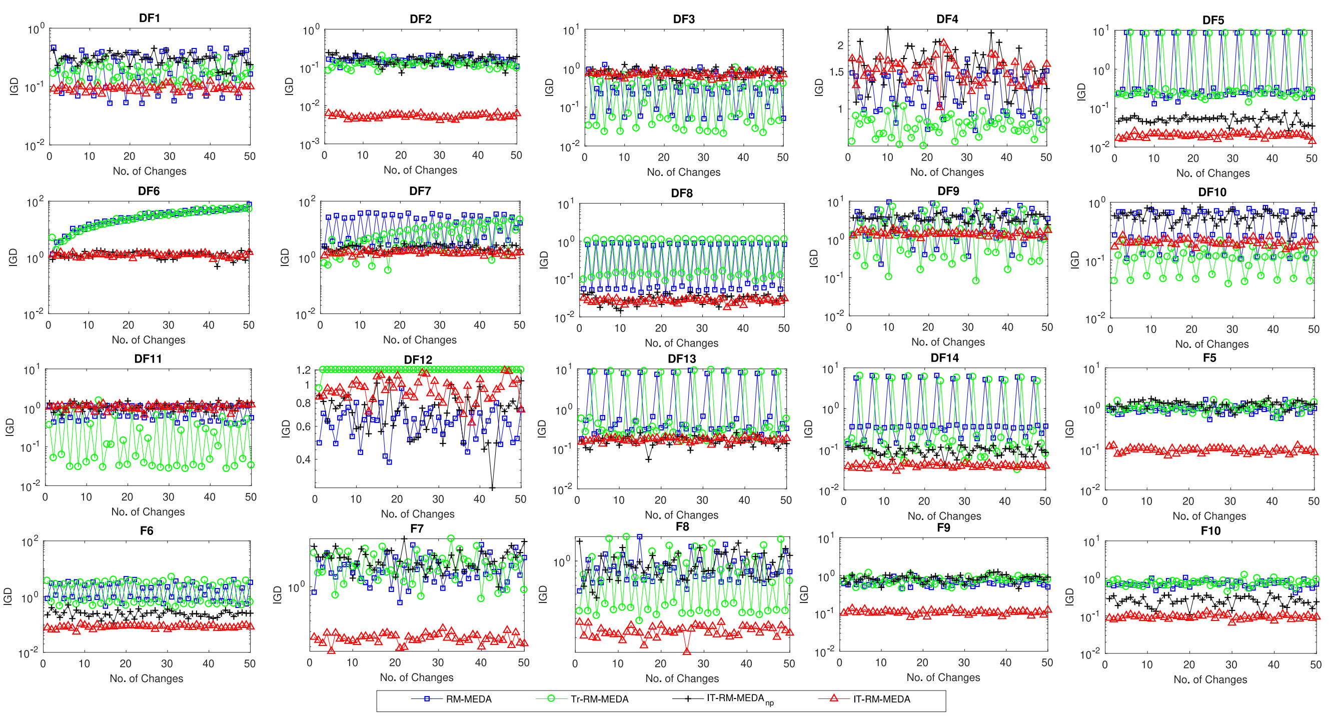


Fig. 5. IGD values of four algorithms at the configuration ($\tau_t = 10$ and $n_t = 1$).

two consecutive POFs are totally different from each other. This complication leads to the possibility of negative transfer increases.

E. Runtime Performance

The capability of speeding up the runtime is one of the most important characteristics of the proposed IT-DMOEA

TABLE II
MEAN AND STANDARD DEVIATION VALUES OF MIGD METRIC OBTAINED BY COMPARED ALGORITHMS FOR DIFFERENT DYNAMIC TEST SETTINGS

Problems	τ_t, n_t	PPS	MDP	MOEA/D-SVR	Tr-DMOEA	MOEA/D-KF	SGEA	IT-RM-MEDA
DF1	10,10	0.0129±5.14E-4(-)	0.0690±2.51E-3(+)	**0.0072±7.45E-5(-)**	0.1794±1.42E-2(+)	0.0190±1.41E-3(-)	0.0235±1.60E-3(+)	0.0205±6.82E-5
	10,5	0.0260±2.53E-4(+)	0.0573±1.07E-3(+)	**0.0088±8.39E-5(-)**	0.1676±1.31E-2(+)	0.0199±5.36E-3(+)	0.0143±1.79E-3(-)	0.0169±7.91E-5
	5,10	0.1164±1.49E-2(+)	0.1188±3.65E-3(+)	0.0151±3.44E-4(+)	0.2263±1.89E-2(+)	0.0711±2.35E-2(+)	0.0340±4.22E-3(+)	**0.0135±1.12E-4**
DF2	10,10	0.0245±1.79E-3(+)	0.0424±3.29E-3(+)	0.0080±8.52E-5(+)	0.1315±8.70E-3(+)	0.0154±6.42E-4(+)	0.1611±2.01E-2(+)	**0.0064±6.13E-5**
	10,5	0.0460±3.77E-3(+)	0.0447±8.99E-4(+)	0.0090±7.36E-5(+)	0.1088±4.80E-3(+)	0.0176±2.59E-4(+)	0.1634±2.18E-2(+)	**0.0068±5.59E-5**
	5,10	0.2488±5.69E-3(+)	0.0789±6.19E-3(+)	0.0361±2.16E-3(+)	0.1808±1.35E-2(+)	0.0535±5.49E-3(+)	0.2345±2.63E-2(+)	**0.0139±1.68E-4**
DF3	10,10	0.2535±2.35E-2(+)	0.1500±1.14E-2(+)	0.2472±1.92E-2(+)	0.3406±2.82E-2(+)	0.2544±1.61E-2(+)	0.3791±4.60E-2(+)	**0.0203±4.55E-4**
	10,5	0.3077±4.77E-2(+)	0.1530±1.79E-2(+)	0.2490±4.30E-2(+)	0.3150±1.91E-2(+)	0.2557±3.47E-2(+)	0.3788±4.64E-2(+)	**0.0414±3.02E-3**
	5,10	0.4812±3.34E-2(+)	0.1534±1.08E-2(+)	0.2919±1.35E-2(+)	0.4117±3.27E-2(+)	0.3799±6.91E-3(+)	0.3864±4.83E-2(+)	**0.0499±3.57E-3**
DF4	10,10	0.0422±3.96E-3(-)	0.8778±2.99E-1(+)	**0.0399±2.69E-3(-)**	1.5869±4.47E-1(+)	0.0641±4.22E-2(-)	0.0403±7.26E-3(-)	0.8042±2.98E-1
	10,5	**0.0348±4.55E-3(-)**	0.8768±2.98E-1(-)	0.0386±2.30E-3(-)	1.5689±5.36E-1(+)	0.0659±3.64E-2(-)	0.0627±9.39E-3(-)	1.0458±1.31E-1
	5,10	0.0595±4.35E-3(-)	0.8917±3.01E-1(-)	**0.0549±3.61E-3(-)**	1.8344±6.48E-1(=)	0.2181±6.40E-2(-)	0.0793±9.94E-3(-)	1.7878±1.47E-1
DF5	10,10	0.0096±8.72E-5(+)	1.2343±4.65E+0(+)	0.0072±5.14E-5(=)	2.4713±1.15E+1(+)	0.0119±1.15E-2(+)	0.0288±3.36E-3(+)	**0.0067±4.82E-5**
	10,5	0.0117±5.82E-4(+)	1.2333±4.22E+0(+)	0.0074±6.66E-5(=)	2.3146±1.21E+1(+)	0.0129±1.84E-2(+)	0.0289±3.67E-3(+)	**0.0073±6.69E-5**
	5,10	0.0210±4.18E-4(+)	1.2407±4.18E+0(+)	0.0158±3.19E-4(=)	2.6205±1.29E+1(+)	0.0804±1.57E-3(+)	0.0492±6.20E-3(+)	**0.0152±2.87E-4**
DF6	10,10	3.5152±3.59E-1(+)	0.7629±1.69E+0(+)	**0.6784±3.48E-1(-)**	6.7379±4.95E+1(+)	2.3345±2.59E-1(+)	0.7980±9.86E-2(+)	0.6925±4.83E-2
	10,5	4.1941±8.82E-1(+)	**0.5104±8.33E-1(-)**	1.0817±2.39E-1(+)	5.5982±2.10E+1(+)	1.1002±1.21E-1(+)	0.8122±9.56E-2(+)	0.6588±5.12E-2
	5,10	3.6975±4.56E-2(+)	3.1601±1.93E+1(=)	**0.8938±1.71E-1(-)**	7.2692±6.09E+1(+)	2.4866±7.70E-2(-)	1.6423±2.02E-1(-)	3.2139±5.27E-1
DF7	10,10	0.7541±8.64E-2(+)	1.1039±2.75E+0(+)	**0.2458±3.72E-2(-)**	4.0323±1.91E+0(+)	0.7046±2.55E-2(+)	1.2949±1.62E-1(+)	0.6382±6.16E-3
	10,5	1.2623±7.72E-2(+)	1.0499±1.70E+0(+)	1.2651±1.17E-1(+)	8.8403±2.32E+1(+)	1.2040±3.72E-2(+)	1.3089±1.71E-1(+)	**0.8382±6.20E-3**
	5,10	0.8746±4.57E-2(-)	2.9739±2.72E+1(+)	**0.6096±1.89E-1(-)**	4.2387±2.76E+0(+)	2.2017±1.17E-1(=)	1.3225±1.65E-1(-)	2.4008±2.59E-1
DF8	10,10	0.0220±5.25E-3(-)	0.7141±1.92E-1(+)	0.0158±2.25E-3(-)	0.8026±1.88E-1(+)	0.0300±3.67E-3(-)	**0.0148±2.51E-3(-)**	0.0662±2.77E-2
	10,5	0.0413±2.85E-3(-)	0.5978±1.86E-1(+)	0.0167±8.56E-4(-)	0.7780±1.76E-1(+)	0.0323±4.95E-3(-)	**0.0158±2.61E-3(-)**	0.1976±4.81E-2
	5,10	0.0869±3.00E-3(-)	0.7149±1.85E-1(+)	0.0239±8.49E-4(-)	0.7993±1.71E-1(+)	0.0759±4.32E-3(-)	**0.0230±3.07E-3(-)**	0.1666±2.35E-2
DF9	10,10	0.4677±3.64E-2(-)	2.1200±3.34E+0(+)	**0.1045±1.20E-2(-)**	2.3714±4.46E+0(-)	0.2252±4.48E-2(-)	0.4885±5.60E-2(-)	3.1829±2.99E-1
	10,5	0.5399±1.59E-2(-)	1.9943±3.03E+0(+)	**0.1483±2.73E-2(-)**	2.5958±5.11E+0(-)	0.2314±3.79E-2(-)	0.4519±5.65E-2(-)	3.6233±3.38E-1
	5,10	0.7226±1.63E-2(-)	1.3349±2.43E+0(+)	**0.1303±3.41E-2(-)**	2.7079±6.96E+0(-)	0.3134±1.39E-2(-)	0.7543±9.53E-2(-)	3.6645±3.64E-1
DF10	10,10	0.2878±5.55E-2(+)	0.6533±2.01E-1(+)	0.0862±3.11E-3(+)	0.1194±2.26E-3(+)	0.1181±1.44E-3(+)	0.1714±2.14E-2(+)	**0.0332±2.05E-3**
	10,5	0.3651±6.27E-2(+)	0.1722±3.25E-2(+)	0.0918±5.27E-3(+)	0.1487±4.60E-3(+)	0.1494±3.72E-3(+)	0.1738±2.24E-2(+)	**0.0352±2.18E-3**
	5,10	0.3344±8.71E-2(+)	0.2046±2.43E-2(+)	0.1178±9.66E-4(+)	0.1493±3.08E-3(+)	0.1796±4.76E-3(+)	0.1766±2.20E-2(+)	**0.0397±2.34E-3**
DF11	10,10	0.0711±5.68E-3(+)	0.0773±6.79E-4(+)	0.0469±5.23E-3(=)	0.3331±2.19E-2(+)	0.0483±3.58E-3(+)	0.0886±1.08E-2(+)	**0.0461±4.68E-3**
	10,5	0.0733±1.90E-3(+)	0.0787±7.61E-4(+)	**0.0451±3.27E-3(-)**	0.3797±5.13E-2(+)	0.0511±2.94E-3(=)	0.0977±1.45E-2(+)	0.0506±3.09E-3
	5,10	0.0749±9.11E-3(+)	0.0849±1.01E-3(+)	**0.0453±3.66E-3(-)**	0.4178±2.94E-2(+)	0.0604±2.91E-2(=)	0.1192±1.37E-2(+)	0.0495±3.58E-3
DF12	10,10	0.8513±8.37E-2(-)	0.6769±1.23E-2(-)	0.5851±5.26E-2(-)	1.1900±1.64E-5(=)	0.6468±5.90E-2(-)	**0.2480±2.98E-2(-)**	1.1915±2.66E-1
	10,5	0.8193±9.67E-2(-)	0.6766±1.06E-2(-)	0.5444±5.19E-2(-)	1.1933±1.29E-5(=)	0.5809±2.49E-2(-)	**0.2895±3.06E-2(-)**	1.1974±2.75E-1
	5,10	0.8618±6.61E-2(-)	**0.3552±1.05E-2(-)**	0.5927±3.45E-2(-)	1.1923±1.65E-5(=)	0.6866±4.74E-2(-)	0.3953±4.32E-2(-)	1.2037±2.72E-1
DF13	10,10	0.2414±2.49E-3(+)	1.3416±3.41E+0(+)	0.2345±2.87E-3(+)	2.7312±1.09E+1(+)	0.2364±5.20E-3(+)	0.1587±1.91E-2(+)	**0.0668±1.40E-3**
	10,5	0.2660±2.30E-3(+)	1.3526±3.64E+0(+)	0.2351±2.57E-3(+)	2.6262±1.08E+1(+)	0.2478±3.42E-3(+)	0.1900±1.49E-2(+)	**0.0702±1.25E-3**
	5,10	0.2338±2.52E-3(+)	1.3573±3.50E+0(+)	0.2454±3.96E-3(+)	2.8032±1.14E+1(+)	0.2414±2.13E-3(+)	0.2735±3.40E-2(+)	**0.0693±1.13E-3**
DF14	10,10	0.0329±6.76E-3(+)	0.9623±2.05E+0(+)	0.0332±6.65E-3(+)	1.8257±6.69E+0(+)	0.0358±5.93E-3(+)	0.0555±6.89E-3(+)	**0.0150±2.00E-3**
	10,5	0.0339±2.95E-3(+)	0.9556±1.98E+0(+)	0.0368±4.41E-3(+)	1.6727±5.57E+0(+)	0.0351±5.95E-3(+)	0.0781±1.01E-2(+)	**0.0198±2.35E-3**
	5,10	0.0400±6.41E-3(+)	0.9730±2.18E+0(+)	0.0361±9.98E-4(+)	1.8333±6.06E+0(+)	0.0466±3.19E-3(+)	0.0838±1.01E-2(+)	**0.0276±3.36E-3**
F5	10,10	2.3272±8.15E+0(+)	**0.0768±1.99E-2(-)**	2.1466±1.80E+0(+)	2.6592±1.98E+0(+)	2.7516±4.27E+0(+)	2.5827±1.88E+0(+)	0.1092±7.19E-2
	10,5	2.4299±1.39E+1(+)	**0.1252±1.82E-2(-)**	2.3276±1.31E+0(+)	2.8026±3.26E+0(+)	2.4003±2.52E+0(+)	2.4366±1.85E+0(+)	0.1888±3.34E-2
	5,10	3.8499±4.96E+0(+)	0.6815±3.37E-1(+)	2.6935±1.35E+0(+)	3.6919±1.63E+0(+)	3.0477±2.21E+0(+)	4.5352±8.15E+0(+)	**0.0884±8.25E-3**
F6	10,10	1.4723±8.97E-1(+)	**0.0509±1.06E-3(-)**	1.8716±9.12E-1(+)	1.3095±8.35E-1(+)	1.3247±3.86E-1(+)	1.2826±6.51E-1(+)	0.2804±5.10E-2
	10,5	1.5399±2.44E+0(+)	0.4274±1.04E-1(+)	1.4738±4.62E+0(+)	1.2349±5.27E-1(+)	1.9116±2.30E+0(+)	1.2529±3.99E-1(+)	**0.2251±8.25E-2**
	5,10	1.4189±8.39E-1(+)	1.2831±5.07E-1(+)	2.0589±2.38E+0(+)	2.4094±2.71E+0(+)	2.2788±1.67E+0(+)	2.3230±8.53E-1(+)	**0.3039±3.04E-2**
F7	10,10	1.2089±2.03E+0(+)	**0.0734±8.87E-3(-)**	1.3380±7.74E-1(+)	1.3270±5.77E-1(+)	1.6307±1.05E+0(+)	1.1449±1.98E-1(+)	0.2436±7.19E-2
	10,5	1.4417±5.84E+0(+)	0.2224±4.75E-2(+)	1.4501±2.21E+0(+)	1.4295±3.32E-1(+)	1.5664±1.77E+0(+)	1.2566±3.59E-1(+)	**0.0766±2.46E-3**
	5,10	1.7897±2.02E+0(+)	1.2676±4.51E-1(+)	1.7304±1.55E+0(+)	3.1593±8.77E-1(+)	1.9415±8.07E-1(+)	2.6158±1.19E+0(+)	**0.0941±2.06E-3**
F8	10,10	0.3216±6.11E-2(+)	0.4479±2.47E-2(+)	0.5578±1.01E-1(+)	0.7875±8.01E-2(+)	**0.2171±5.94E-3(-)**	0.8432±6.46E-2(+)	0.3157±7.93E-2
	10,5	0.4917±1.21E-2(+)	0.4334±1.93E-2(+)	0.7866±3.72E-1(+)	0.7331±8.94E-2(+)	**0.2393±2.18E-2(-)**	0.8125±2.51E-2(+)	0.3076±3.30E-2
	5,10	0.5349±2.52E-2(+)	0.5831±7.23E-2(+)	0.6813±1.05E-1(+)	1.0615±4.13E-1(+)	**0.3266±1.16E-2(-)**	1.3390±1.69E-1(+)	0.3536±1.38E-3
F9	10,10	0.9658±6.26E-1(+)	**0.1478±3.61E-2(-)**	1.5411±8.39E-1(+)	1.4721±5.94E-1(+)	0.8209±5.96E-1(+)	1.3766±6.30E-1(+)	0.2221±1.27E-3
	10,5	0.9949±2.62E+0(+)	**0.1769±2.62E-2(-)**	1.6188±5.61E-1(+)	1.4594±4.75E-1(+)	0.7225±2.34E-1(+)	1.3579±3.91E-1(+)	0.2079±3.30E-3
	5,10	1.7503±1.29E+0(+)	0.6782±3.47E-1(+)	2.7898±4.17E+0(+)	2.6079±1.14E+0(+)	1.7066±6.89E-1(+)	3.0071±2.05E+0(+)	**0.1797±3.34E-3**
F10	10,10	3.5221±7.01E+0(+)	1.1779±1.96E+0(+)	2.1921±1.87E+0(+)	2.7327±2.60E+0(+)	2.9650±5.78E+0(+)	4.1148±1.84E+1(+)	**0.1252±1.56E-3**
	10,5	1.8889±2.72E+0(+)	**0.1690±2.10E-2(-)**	1.9292±9.45E-1(+)	1.4407±6.08E-1(+)	0.6933±1.57E-1(-)	1.6451±9.44E-1(+)	0.9813±2.10E-3
	5,10	3.5543±7.53E+0(+)	0.6153±3.01E-1(+)	4.5335±2.21E+1(+)	2.7845±3.20E+0(+)	1.9263±1.46E+0(+)	3.1591±7.60E+0(+)	**0.0785±4.45E-3**
+/=/-		46/0/14	46/1/13	36/4/20	53/4/3	40/2/18	45/0/15	

algorithm. In this section, we compare the runtime of different algorithms and present the results in Table IV.[3] Table IV shows that the runtime of IT-DMOEA is the fastest in most of the benchmark functions. This shows that the proposed presearch and individuals transfer modules are very efficient. The runtime of Tr-DMOEA is also much longer than that of IT-DMOEA. The important difference is that the Tr-DMOEA needs a considerable amount of computation in order to obtain the latent space of objective space. In the Tr-DOMEA [8], finding solutions in the latent space requires $O(S^3L)$, where L is the total number of bits of the input. Meanwhile, IT-DMOEA is based on individual transfer in the decision space with low time complexity. The simulation results show that the proposed IT-DMOEA method is not only fast but also greatly improves the quality of solutions.

F. Discussions on IT-DMOEA

To further investigate the role of the guided population, a modified version, called IT-RM-MEDA$_p$, is designed. The IT-RM-MEDA$_p$ algorithm directly uses the results of the presearch, the guided population, as the initial population.[4] Table II

[3]The implementation environment is as follows: 1.6-GHz Intel Core i5, 8-GB 1600-MHz DDR3.

[4]In order to maintain the population size as S, in some cases, random individuals are added.

TABLE III
MEAN AND STANDARD DEVIATION VALUES OF MS METRIC OBTAINED BY COMPARED ALGORITHMS FOR DIFFERENT DYNAMIC TEST SETTINGS

Problems	τ_t, n_t	PPS	MDP	MOEA/D-SVR	Tr-DMOEA	MOEA/D-KF	SGEA	IT-RM-MEDA
DF1	10,10	0.5304±3.13E-2(+)	0.9444±4.10E-3(+)	0.7498±1.04E-2(+)	0.8898±1.28E-2(+)	0.8143±1.22E-2(+)	0.8519±1.25E-2(+)	**0.9874±1.25E-4**
	10,5	0.5057±3.61E-2(+)	0.9635±1.90E-3(+)	0.7903±4.72E-2(+)	0.9203±6.54E-3(+)	0.7995±9.61E-3(+)	0.8397±7.84E-3(+)	**0.9775±2.57E-4**
	5,10	0.4799±5.94E-2(+)	0.9307±4.85E-3(+)	0.7816±1.29E-2(+)	0.8416±3.63E-3(+)	0.7807±1.44E-2(+)	0.7536±4.35E-3(+)	**0.9947±8.09E-5**
DF2	10,10	0.6245±2.22E-2(+)	0.9646±8.99E-3(+)	0.9194±4.83E-3(+)	0.9394±1.42E-3(+)	0.8330±9.00E-3(+)	0.9442±1.61E-3(+)	**0.9874±1.58E-4**
	10,5	0.6160±2.97E-2(+)	0.9651±4.32E-3(+)	0.9173±5.67E-3(+)	0.9473±7.17E-4(+)	0.8202±1.95E-2(+)	0.9266±3.02E-3(+)	**0.9984±4.52E-5**
	5,10	0.6225±2.56E-2(+)	0.9320±1.90E-2(+)	0.7776±2.62E-2(+)	0.9166±1.76E-3(+)	0.8029±8.99E-3(+)	0.8686±5.62E-3(+)	**0.9956±2.54E-5**
DF3	10,10	0.3295±7.46E-2(+)	0.7067±9.11E-2(+)	0.2528±2.53E-2(+)	0.6198±9.82E-3(+)	0.2195±3.17E-2(+)	0.3832±2.30E-2(+)	**0.8602±1.13E-3**
	10,5	0.3633±5.54E-2(+)	0.7189±9.07E-2(+)	0.2713±4.45E-2(+)	0.4701±4.55E-2(+)	0.2300±3.81E-2(+)	0.3093±2.62E-2(+)	**0.7652±2.03E-3**
	5,10	0.3427±8.43E-2(+)	0.7205±8.50E-2(+)	0.3896±2.06E-2(+)	0.4496±1.60E-2(+)	0.2828±3.87E-2(+)	0.4366±3.20E-2(+)	**0.8545±4.52E-3**
DF4	10,10	0.2042±2.16E-2(+)	0.3213±3.76E-2(+)	0.2786±2.51E-2(+)	0.3176±1.59E-2(+)	0.3080±1.92E-2(+)	0.2018±1.51E-2(+)	**0.4544±2.82E-2**
	10,5	0.1828±1.45E-2(+)	0.3220±3.87E-2(+)	0.2885±1.49E-2(+)	**0.4383±1.09E-1(-)**	0.2705±3.35E-2(+)	0.1603±7.76E-3(+)	0.3989±2.99E-2
	5,10	0.1796±1.22E-2(+)	0.3600±3.68E-2(+)	0.3999±4.17E-3(+)	0.2442±2.82E-2(+)	0.2985±1.71E-2(+)	0.2967±2.30E-2(+)	**0.4542±2.93E-2**
DF5	10,10	0.9321±2.29E-2(+)	1.0000±2.26E-33(=)	0.9431±3.79E-3(+)	0.9990±4.69E-6(+)	0.9303±1.33E-2(+)	0.9978±4.84E-6(+)	**1.0000±0**
	10,5	0.9418±8.86E-3(+)	**1.0000±2.50E-33(-)**	0.9194±7.41E-3(+)	1.0000±7.81E-17(-)	0.9426±8.31E-3(+)	0.9985±1.31E-6(+)	0.9997±6.16E-6
	5,10	0.9388±8.17E-3(+)	**1.0000±2.26E-33(-)**	0.9890±1.87E-4(+)	1.0000±1.82E-13(-)	0.9560±2.89E-3(+)	0.9930±3.78E-5(+)	0.9990±8.04E-6
DF6	10,10	0.5472±9.38E-2(+)	**1.0000±1.96E-15(-)**	0.8157±4.89E-2(+)	0.6156±5.86E-2(+)	0.7500±7.75E-2(+)	0.6543±1.25E-1(+)	0.9965±3.14E-6
	10,5	0.4365±1.00E-1(+)	1.0000±4.65E-15(=)	0.8000±2.14E-2(+)	1.0000±0(=)	0.7099±4.23E-2(+)	0.8341±6.64E-2(+)	**1.0000±0**
	5,10	0.6398±6.31E-2(+)	**1.0000±1.04E-9(-)**	0.7386±7.02E-2(+)	0.6383±4.50E-2(+)	0.8682±2.34E-2(+)	0.7009±8.50E-2(+)	0.9949±3.14E-5
DF7	10,10	0.6196±1.08E-1(+)	**0.9478±2.37E-2(-)**	0.8034±5.81E-2(-)	0.6662±4.63E-2(+)	0.7515±6.74E-2(+)	0.6449±1.31E-1(+)	0.7899±5.81E-2
	10,5	0.5993±1.10E-1(+)	0.9333±1.20E-2(-)	0.7900±2.31E-2(-)	**1.0000±0(-)**	0.7155±4.20E-2(-)	0.8145±6.14E-2(-)	0.3415±2.00E-3
	5,10	0.5980±9.90E-2(+)	**0.9981±1.78E-4(-)**	0.6383±4.53E-2(-)	0.6915±4.90E-2(-)	0.8441±3.47E-2(-)	0.6958±8.49E-2(-)	0.3787±5.01E-2
DF8	10,10	0.4426±9.73E-2(+)	0.4704±6.89E-2(+)	0.3501±1.33E-1(+)	0.6571±4.20E-2(+)	0.4329±7.62E-2(+)	0.5989±6.31E-2(+)	**0.8714±5.31E-3**
	10,5	0.4058±8.25E-2(+)	**0.6154±5.91E-2(-)**	0.5101±1.91E-1(-)	0.4501±9.90E-2(+)	0.3004±3.56E-2(+)	0.4763±5.09E-2(+)	0.4934±5.10E-2
	5,10	0.3993±8.65E-2(+)	0.5636±7.01E-2(+)	**0.8833±3.26E-2(-)**	0.7103±5.83E-2(+)	0.6078±5.55E-2(+)	0.6437±8.69E-2(+)	0.8749±3.15E-3
DF9	10,10	0.8396±4.83E-2(-)	0.7121±8.49E-2(-)	0.7451±9.95E-2(-)	0.6461±4.75E-2(-)	0.7012±9.57E-2(-)	**0.9092±1.08E-2(-)**	0.3237±3.67E-2
	10,5	0.8033±8.22E-2(-)	0.7750±6.93E-2(-)	0.6968±1.21E-1(-)	0.8068±1.06E-1(-)	0.6775±9.75E-2(-)	**0.8958±1.80E-2(-)**	0.2613±2.88E-2
	5,10	0.8280±4.86E-2(-)	0.9197±1.25E-2(-)	0.6670±1.45E-2(-)	0.7410±4.31E-2(-)	0.7588±3.87E-2(-)	**0.9350±2.80E-3(-)**	0.4497±1.15E-2
DF10	10,10	0.9942±4.56E-4(+)	0.4144±2.46E-1(+)	0.8876±4.32E-3(+)	0.9996±2.26E-7(+)	0.9125±2.06E-3(+)	1.0000±3.08E-17(=)	**1.0000±0**
	10,5	0.9983±1.45E-4(+)	0.9714±2.05E-2(+)	0.9184±5.58E-3(+)	0.9998±1.28E-7(+)	0.9502±5.24E-3(+)	0.9895±7.31E-4(+)	**1.0000±0**
	5,10	0.9993±2.84E-5(+)	0.9553±8.16E-3(+)	0.9256±7.58-3(+)	0.9993±8.71E-7(+)	0.9927±2.25E-4(+)	1.0000±1.43E-14(=)	**1.0000±0**
DF11	10,10	0.8667±4.99E-3(+)	0.9587±2.30E-3(+)	0.9382±5.66E-3(+)	**0.9989±3.24E-6(-)**	0.9826±5.24E-4(+)	0.9562±1.70E-4(+)	0.9982±1.52E-5
	10,5	0.8888±6.35E-3(+)	0.9589±2.33E-3(+)	0.9139±5.78E-3(+)	0.9709±4.25E-3(-)	0.9008±5.82E-3(+)	0.9498±1.40E-4(+)	**0.9955±8.04E-5**
	5,10	0.8879±3.88E-3(+)	0.9540±2.66E-3(+)	0.9912±1.98E-4(+)	0.9762±1.01E-3(-)	0.9891±2.92E-4(+)	0.8901±1.65E-3(+)	**0.9984±2.01E-5**
DF12	10,10	0.0011±5.16E-5(-)	0.6655±1.99E-2(-)	0.0708±2.49E-2(-)	0.0056±8.25E-5(-)	0.1009±3.58E-2(-)	**0.6982±8.15E-3(-)**	0.0005±1.19E-5
	10,5	0.0003±4.90E-6(+)	0.6827±2.01E-2(-)	0.2199±4.62E-2(-)	0.0045±6.75E-5(-)	0.3006±8.51E-2(-)	**0.6845±5.66E-3(-)**	0.0005±2.99E-5
	5,10	0.0138±6.79E-3(-)	**0.9000±8.04E-3(-)**	0.2019±4.95E-2(-)	0.0008±9.22E-6(-)	0.1060±2.71E-2(-)	0.7045±1.67E-2(-)	0.0004±9.39E-6
DF13	10,10	0.9635±8.87E-4(+)	0.9970±1.43E-4(+)	0.9359±2.53E-3(+)	0.9961±1.55E-5(+)	0.9361±3.61E-3(+)	0.9974±2.43E-6(+)	**0.9998±6.16E-7**
	10,5	0.9562±1.53E-3(+)	0.9961±1.78E-4(+)	0.9331±2.26E-3(+)	0.9950±5.26E-5(+)	0.9087±3.39E-3(+)	0.9960±2.32E-6(+)	**0.9996±3.14E-7**
	5,10	0.9631±1.47E-3(+)	0.9985±3.88E-4(+)	0.9526±2.45E-3(+)	0.9972±1.70E-5(+)	0.9479±1.58E-3(+)	0.9945±5.17E-6(+)	**0.9991±5.03E-7**
DF14	10,10	0.9629±9.25E-4(+)	**1.0000±0(-)**	0.8290±7.23E-3(+)	0.9162±2.92E-3(+)	0.7720±1.67E-3(+)	0.9864±3.88E-4(+)	0.9958±1.25E-7
	10,5	0.9641±1.37E-3(+)	**1.0000±0(-)**	0.7009±3.48E-2(+)	0.9270±4.72E-3(+)	0.7855±2.70E-2(+)	0.9900±1.18E-4(+)	0.9952±2.57E-7
	5,10	0.9511±1.41E-3(+)	**1.0000±0(-)**	0.9068±3.21E-3(+)	0.9039±1.78E-3(+)	0.8260±2.94E-3(+)	0.9757±7.40E-4(+)	0.9950±1.08E-7
F5	10,10	0.4531±1.18E-1(+)	0.9426±2.37E-3(+)	0.6822±5.59E-2(+)	0.5998±2.29E-2(+)	0.6696±8.83E-2(+)	0.5697±2.56E-2(+)	**0.9434±3.16E-3**
	10,5	0.4847±7.91E-2(+)	0.9285±2.62E-3(+)	0.6332±1.52E-1(+)	0.6768±3.79E-2(+)	0.6133±7.22E-2(+)	0.5418±5.11E-2(+)	**0.9441±1.18E-3**
	5,10	0.4882±7.14E-2(+)	0.7222±2.45E-2(+)	0.9057±6.86E-3(+)	0.6702±3.11E-2(+)	0.7623±3.10E-2(+)	0.7105±3.43E-2(+)	**0.9338±1.14E-3**
F6	10,10	0.4450±8.38E-2(+)	**0.9660±5.66E-4(-)**	0.6608±7.21E-2(+)	0.5969±2.54E-2(+)	0.4971±5.00E-2(+)	0.6497±1.79E-2(+)	0.8537±3.46E-3
	10,5	0.4551±7.85E-2(+)	0.8477±1.42E-2(+)	0.3979±5.91E-2(+)	0.6917±2.66E-2(+)	0.5548±5.68E-2(+)	0.6368±2.42E-2(+)	**0.9378±5.21E-3**
	5,10	0.3917±5.07E-2(+)	0.6308±2.41E-2(+)	0.6195±7.74E-2(+)	0.6755±3.79E-2(+)	0.6745±6.71E-2(+)	0.6288±3.78E-2(+)	**0.9256±5.23E-3**
F7	10,10	0.3886±6.36E-2(+)	**0.9576±1.64E-3(-)**	0.4863±6.52E-2(+)	0.5934±2.50E-2(+)	0.5414±6.18E-2(+)	0.6695±1.99E-2(+)	0.9382±2.58E-3
	10,5	0.3998±7.03E-2(+)	0.9136±6.04E-3(+)	0.5989±4.34E-2(+)	0.6482±1.75E-2(+)	0.5175±7.25E-2(+)	0.6131±2.86E-2(+)	**0.9628±5.78E-3**
	5,10	0.3465±7.10E-2(+)	0.6556±1.90E-2(+)	0.6392±4.24E-2(+)	0.6489±6.25E-2(+)	0.7871±3.57E-2(+)	0.6185±3.31E-2(+)	**0.9510±1.14E-3**
F8	10,10	0.9836±1.96E-3(+)	1.0000±3.56E-33(=)	0.9685±4.32E-3(+)	**1.0000±0(-)**	0.9835±2.27E-3(+)	0.9990±5.16E-7(+)	0.9996±1.19E-31
	10,5	0.9929±1.19E-3(+)	1.0000±1.70E-33(=)	0.9729±5.39E-3(+)	**1.0000±0(-)**	0.9871±2.64E-3(+)	0.9993±2.83E-7(=)	0.9997±1.18E-31
	5,10	0.9813±4.36E-3(+)	1.0000±1.08E-32(=)	0.9654±4.23E-3(+)	**1.0000±0(-)**	0.9980±6.27E-5(+)	0.9968±1.13E-5(+)	0.9997±1.24E-31
F9	10,10	0.3631±5.78E-2(+)	0.9410±3.38E-3(+)	0.5896±7.33E-2(+)	0.6495±2.85E-2(+)	0.5602±2.11E-2(+)	0.6651±2.03E-2(+)	**0.9627±1.19E-3**
	10,5	0.3699±5.96E-2(+)	0.9377±2.68E-3(+)	0.6558±5.60E-2(+)	0.6831±9.40E-3(+)	0.5528±2.17E-2(+)	0.6663±1.95E-2(+)	**0.9629±1.38E-3**
	5,10	0.3211±4.74E-2(+)	0.7497±3.00E-2(+)	0.6675±2.38E-2(+)	0.5888±4.40E-2(+)	0.6172±4.646-2(+)	0.5521±6.36E-2(+)	**0.9739±1.15E-3**
F10	10,10	0.4109±7.33E-2(+)	0.9262±9.01E-3(+)	0.7451±8.73E-2(+)	0.7709±3.57E-2(+)	0.6982±3.67E-2(+)	0.7811±4.71E-2(+)	**0.9695±4.54E-3**
	10,5	0.4366±7.32E-2(+)	0.9266±5.95E-3(+)	0.5647±2.94E-2(+)	0.6699±2.22E-2(+)	0.5614±2.37E-2(+)	0.6719±1.93E-2(+)	**0.9594±3.90E-3**
	5,10	0.3551±8.38E-2(+)	0.8383±4.50E-2(+)	0.6871±8.68E-2(+)	0.6946±1.06E-1(+)	0.7753±4.90E-2(+)	0.7602±5.36E-2(+)	**0.9617±1.74E-3**
+/=/-		55/0/5	36/5/19	49/0/11	42/1/17	52/0/8	49/3/8	

(supplementary material) records the MIGD and MS values of IT-RM-MEDA$_{np}$, IT-RM-MEDA$_p$, and IT-RM-MEDA. By observing these results, we have the following findings: first, it is not hard to find that the result of IT-RM-MEDA$_p$ is obviously better than that of IT-RM-MEDA$_{np}$, and this suggests that the presearch strategy does work and the guided population has good quality. Second, the comparisons between IT-RM-MEDA$_p$ and IT-RM-MEDA show that IT-RM-MEDA achieves a significant improvement in 19 out of 40 cases in terms of the MIGD metric and achieves a significant improvement in 24 out of 40 cases in terms of the MS metric. The experimental results indicate that the transfer learning method can enhance the performance of related algorithms, and the coverage and convergence can be further improved by incorporating the guided population with transfer learning.

The individual transfer-based population prediction is effective in DMOPs with dramatic changes due to the inclusion of presearch, which considers the distributions of solutions under different environments. In a dynamic environment with drastic changes, the distributions of solutions under different environments may be radically different. Therefore, when using transfer learning to predict the initial population for solving DMOPs, negative transfer may lead to poor performance with local optimum or poor diversity. In the presearch stage, the solution distribution under a new environment can be explored. The prediction model will be guided to the correct direction,

TABLE IV
MEAN VALUES OF RUNTIME OBTAINED BY COMPARED ALGORITHMS AT A CONFIGURATION OF τ_t=10 AND n_t=10 (IN SECONDS)

Problems	IT-DMOEA	MDP	MOEA/D-SVR	Tr-DMOEA	MOEA/D-KF	SGEA	PPS
DF1	0.7999	0.9580	3.4562	46.1883	3.0723	**0.5219**	3.0559
DF2	0.8076	**0.4117**	3.3490	36.3649	3.2793	0.4849	2.9185
DF3	0.7375	0.5536	2.9751	45.9701	2.7094	**0.5013**	1.8899
DF4	0.6477	**0.4030**	2.3864	16.5985	2.3663	0.4174	1.7276
DF5	0.7324	0.4159	3.4126	39.9392	3.2197	**0.3346**	3.0275
DF6	**0.7135**	0.7529	2.5676	52.3478	2.1687	0.7245	2.4351
DF7	0.8049	**0.7355**	2.6245	37.8134	2.3021	0.8557	2.5906
DF8	0.8297	0.5778	3.0093	42.5193	2.7812	**0.2978**	2.8804
DF9	**0.5523**	0.5732	2.6720	48.4561	2.5312	0.6145	2.0859
DF10	1.8495	2.4438	10.4172	86.6931	9.7104	**0.9557**	8.9799
DF11	**0.9427**	1.2538	8.9900	89.1901	9.1663	0.9535	8.1743
DF12	**0.8792**	6.2164	5.3619	97.0358	4.7006	3.5782	4.9663
DF13	1.2273	1.1792	10.5268	109.3813	9.8379	**0.9079**	8.5382
DF14	**1.2543**	1.6758	10.0918	90.8033	9.6494	2.4510	8.5602
F5	**1.3396**	10.6630	7.2160	51.3163	5.9335	4.9360	4.8181
F6	**1.3208**	10.5513	6.4391	54.5093	6.0874	4.4336	5.0109
F7	**1.3236**	9.7266	6.4625	57.1244	6.1244	4.3350	5.9100
F8	1.3874	3.2069	7.2878	126.9918	8.6568	**1.1961**	7.6505
F9	**1.1304**	9.8454	6.4259	97.0314	6.3404	4.6409	5.2085
F10	**1.1389**	10.4060	7.0571	59.7421	5.2330	4.3505	4.9030

and the possibility of negative transfer can be reduced by introducing the presearch strategy; thereby, facilitating efficient transfer learning.

The proposed IT-DMOEA achieves improved performance with less time because the prediction model is a sample-based classifier that can efficiently classify a large number of excellent individuals in the current environment. Therefore, this approach is more efficient than other prediction methods based on transfer learning. Furthermore, the proposed prediction model is a solution variable feature-based classification model, which can predict excellent solutions for solving linear and nonlinear problems generally.

V. CONCLUSION

In recent years, transfer learning has been proved to be one of the effective means for solving dynamic multiobjective optimization. However, when the problem changes dramatically, the knowledge learned in the source domain will have a negative effect on the learning in the target domain, which is also called negative transfer. Negative transfer leads to a variety of problems, one of which is to take the search for solutions in the wrong direction so that a noteworthy amount of the computing resources is wasted.

In this article, a transfer learning-based dynamic multiobjective optimization algorithm, IT-DMOEA, has been proposed. The IT-DMOEA applied a presearch strategy to reduce the possibility of negative transfer and an individual-based transfer learning technique to accelerate the construction of an initial population. Consequently, the population generated by the proposed method can be used for improving the performance of other kinds of population-based multiobjective evolutionary algorithms in a drastically changing environment.

The experimental results show that the proposed algorithm considerably improves dynamic optimization performance. Particularly, the proposed algorithm has improved by tens or even hundreds of times in speed of finding the POS compared with that of existing transfer learning-based algorithms.

There are several possible directions for future work. Weak classifiers with low time complexity and probability density estimation methods will be applied to accelerate the optimization process. In the presearch stage, more precise and effective methods will be used to obtain better converged and diversified individuals. A rich body of transfer learning techniques can also be applied here to inspire further innovations in solving real-world applications with various degrees of complexities and uncertainties.

REFERENCES

[1] Y. Yang, Y. Sun, and Z. Zhu, "Multi-objective memetic algorithm based on request prediction for dynamic pickup-and-delivery problems," in *Proc. Congr. Evol. Comput.*, 2017, pp. 1728–1733.

[2] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: Test cases, approximations, and applications," *IEEE Trans. Evol. Comput.*, vol. 8, no. 5, pp. 425–442, Oct. 2004.

[3] C. Cruz, J. R. González, and D. A. Pelta, "Optimization in dynamic environments: A survey on problems methods and measures," *Soft Comput.*, vol. 15, no. 7, pp. 1427–1448, Dec. 2010.

[4] S. Boonphoapichart, S. Komada, and T. Hori, "Robot's motion decision-making system in unknown environment and its application to a mobile robot," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, vol. 1, Dec. 2002, pp. 18–23.

[5] P. Kulvanit, N. Chaiyaratana, and D. Laowattana, "Biped fast walking gait shaping via evolutionary multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, 2007, pp. 4019–4026.

[6] A. Muruganantham, K. Tan, and P. Vadakkepat, "Evolutionary dynamic multiobjective optimization via Kalman filter prediction," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2862–2873, Dec. 2016.

[7] W. T. Koo, K. G. Chi, and K. C. Tan, "A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment," *Memetic Comput.*, vol. 2, no. 2, pp. 87–110, 2010.

[8] M. Jiang, Z. Huang, L. Qiu, W. Huang, and G. Yen, "Transfer learning based dynamic multiobjective optimization algorithms," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 501–514, Aug. 2018.

[9] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: A survey," *Knowl. Based Syst.*, vol. 80, pp. 14–23, May 2015.

[10] 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.

[11] M. T. Rosenstein, Z. Marx, L. P. Kaelbling, and T. G. Dietterich, "To transfer or not to transfer," in *Proc. NIPS Workshop Transfer Learn.*, vol. 898, 2005, p. 3.

[12] W. Dai, Q. Yang, G. R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 193–200.

[13] Q. Li, J. Zou, S. Yang, J. Zheng, and R. Gan, "A predictive strategy based on special points for evolutionary dynamic multi-objective optimization," *Soft Comput.*, vol. 23, no. 1, pp. 1–17, 2018.

[14] Y. Wu, Y. Jin, and X. Liu, "A directed search strategy for evolutionary dynamic multiobjective optimization," *Soft Comput.*, vol. 19, no. 11, pp. 3221–3235, Nov. 2015.

[15] G. Ruan, G. Yu, J. Zheng, J. Zou, and S. Yang, "The effect of diversity maintenance on prediction in dynamic multi-objective optimization," *Appl. Soft Comput.*, vol. 58, pp. 631–647, Sep. 2017.

[16] D. Gong, B. Xu, Y. Zhang, Y. Guo, and S. Yang, "A similarity-based cooperative co-evolutionary algorithm for dynamic interval multi-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 142–156, Feb. 2020.

[17] R. Azzouz, S. Bechikh, and L. Ben Said, "A multiple reference point-based evolutionary algorithm for dynamic multi-objective optimization with undetectable changes," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 3168–3175.

[18] Q. Liu and J. Ding, "Reference vector based multidirectional prediction for evolutionary dynamic multiobjective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2019, pp. 1081–1087.

[19] Y. Jin, C. Yang, J. Ding, and T. Chai, "Reference point based prediction for evolutionary dynamic multiobjective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 3769–3776.

[20] R. Chen, K. Li, and X. Yao, "Dynamic multiobjectives optimization with a changing number of objectives," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 157–171, Feb. 2018.
[21] X. Xu, Y. Tan, W. Zheng, and S. Li, "Memory-enhanced dynamic multi-objective evolutionary algorithm based on LP decomposition," *Appl. Sci.*, vol. 8, no. 9, p. 1673, 2018.
[22] Y. Wang and B. Li, "Multi-strategy ensemble evolutionary algorithm for dynamic multi-objective optimization," *Memetic Comput.*, vol. 2, no. 1, pp. 3–24, 2010.
[23] R. Azzouz, S. Bechikh, and L. B. Said, "A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy," *Soft Comput.*, vol. 21, no. 4, pp. 885–906, 2017.
[24] Z. Peng, J. Zheng, J. Zou, and M. Liu, "Novel prediction and memory strategies for dynamic multiobjective optimization," *Soft Comput.*, vol. 19, no. 9, pp. 2633–2653, Sep. 2015.
[25] J. Branke, "Memory enhanced evolutionary algorithms for changing optimization problems," in *Proc. Congr. Evol. Comput.*, 1999, pp. 1875–1882.
[26] Y. Woldesenbet and G. Yen, "Dynamic evolutionary algorithm with variable relocation," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 500–513, Jun. 2009.
[27] L. Cao, L. Xu, E. D. Goodman, C. Bao, and S. Zhu, "Evolutionary dynamic multiobjective optimization assisted by a support vector regression predictor," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 305–319, Apr. 2020.
[28] M. Rong, D. Gong, Y. Zhang, Y. Jin, and W. Pedrycz, "Multidirectional prediction approach for dynamic multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3362–3374, Sep. 2019.
[29] A. Zhou, Y. Jin, and Q. Zhang, "A population prediction strategy for evolutionary dynamic multiobjective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 40–53, Jan. 2014.
[30] Y. Ma, R. Liu, and R. Shang, "A hybrid dynamic multi-objective immune optimization algorithm using prediction strategy and improved differential evolution crossover operator," in *Neural Information Processing*, B.-L. Lu, L. Zhang, and J. Kwok, Eds. Heidelberg, Germany: Springer, 2011, pp. 435–444.
[31] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization," in *Proc. Int. Conf. Evol. Multi Criterion Optim.*, 2007, pp. 832–846.
[32] Z. Wang, Z. Zhan, K. Du, Z. Yu, and J. Zhang, "Orthogonal learning particle swarm optimization with variable relocation for dynamic optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 594–600.
[33] M. Liu, J. Zheng, J. Wang, Y. Liu, and L. Jiang, "An adaptive diversity introduction method for dynamic evolutionary multiobjective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 3160–3167.
[34] B. Xu, Y. Zhang, D. Gong, Y. Guo, and M. Rong, "Environment sensitivity-based cooperative co-evolutionary algorithms for dynamic multi-objective optimization," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 15, no. 6, pp. 1877–1890, Nov. 2018.
[35] S. Nguyen, M. Zhang, M. Johnston, and K. C. Tan, "Automatic design of scheduling policies for dynamic multi-objective job shop scheduling via cooperative coevolution genetic programming," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 193–208, Apr. 2014.
[36] M. Helbig and A. P. Engelbrecht, "Analyses of guide update approaches for vector evaluated particle swarm optimisation on dynamic multi-objective optimisation problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2012, pp. 1–8.
[37] A. Gupta, Y. Ong, and L. Feng, "Insights on transfer optimization: Because experience is the best teacher," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 51–64, Feb. 2018.
[38] A. T. W. Min, Y. Ong, A. Gupta, and C. Goh, "Multiproblem surrogates: Transfer evolutionary multiobjective optimization of computationally expensive problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 15–28, Feb. 2019.
[39] K. K. Bali, Y. 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.
[40] B. Da, A. Gupta, and Y. Ong, "Curbing negative influences online for seamless transfer evolutionary optimization," *IEEE Trans. Cybern.*, vol. 49, no. 12, pp. 4365–4378, Dec. 2019.
[41] M. Jiang, L. Qiu, Z. Huang, and G. G. Yen, "Dynamic multi-objective estimation of distribution algorithm based on domain adaptation and nonparametric estimation," *Inf. Sci.*, vol. 435, pp. 203–223, Apr. 2018.
[42] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 838–856, Dec. 2015.
[43] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems," *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, 1996.
[44] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, May 2011.
[45] S. Jiang, S. Yang, X. Yao, K. Tan, M. Kaiser, and N. Krasnogor, "Benchmark problems for cec2018 competition on dynamic multiobjective optimisation," in *Proc. CEC Competition*, 2018, pp. 1–8.
[46] S. Jiang and S. Yang, "A steady-state and generational evolutionary algorithm for dynamic multi-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 65–82, Feb. 2017.
[47] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 41–63, Feb. 2008.
[48] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011.

Min Jiang (Senior Member, IEEE) received the bachelor's and Ph.D. degrees in computer science from Wuhan University, Wuhan, China, in 2001 and 2007, respectively.

He was a Postdoctoral Researcher with the Department of Mathematics, Xiamen University, Xiamen, China, where he is currently a Professor with the Department of Artificial Intelligence. His main research interests are machine learning, computational intelligence, and robotics. He has a special interest in dynamic multiobjective optimization, transfer learning, software development, and in the basic theories of robotics.

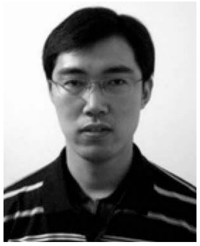
Prof. Jiang received the Outstanding Reviewer Award from the IEEE TRANSACTIONS ON CYBERNETICS in 2016. He is currently serving as an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS and the IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS. He is the Chair of IEEE CIS Xiamen Chapter.

Zhenzhong Wang (Graduate Student Member, IEEE) received the bachelor's degree in computer science and technology from Northeastern University, Shenyang, China, in 2017. He is currently pursuing the master's degree with the School of Informatics, Xiamen University, Xiamen, China.

His research interests include computational intelligence and machine learning.

Shihui Guo (Member, IEEE) received the B.S. degree in electrical engineering from Peking University, Beijing, China, in 2010, and the Ph.D. degree in computer animation from the National Centre for Computer Animation, Bournemouth University, Poole, U.K., in 2015.

He is an Associate Professor with the School of Informatics, Xiamen University, Xiamen, China. His research interests include computational intelligence and human–computer interaction.



Xing Gao (Member, IEEE) received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2009.

He is currently an Associate Professor with the School of Informatics, Xiamen University, Xiamen, China. His research interests include computing intelligence and computer graphics.

Kay Chen Tan (Fellow, IEEE) received the B.Eng. degree (First Class Hons.) in electronics and electrical engineering and the Ph.D. degree in evolutionary computation and control systems 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.

Prof. Tan is the Editor-in-Chief of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION. He was the Editor-in-Chief of the IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE from 2010 to 2013 and currently serves as the editorial board member of over ten journals. He was an Elected Member of the IEEE CIS AdCom from 2017 to 2019.