Solving Dynamic Multiobjective Optimization Problems via Feedback-Guided Transfer and Trend Manifold Prediction

Yong Wang, *Member, IEEE*, Kuichao Li, Gai-Ge Wang, *Member, IEEE*, Dunwei Gong, *Senior Member, IEEE*, and Keqin Li, *Fellow, IEEE*

***Abstract*—Solving dynamic multiobjective optimization problems (DMOPs) is very challenging due to the requirements to respond rapidly and precisely to changes in an environment. Many prediction- and memory-based algorithms have been recently proposed for meeting these requirements. However, much useful knowledge has been ignored during the historical search process, and prediction deviations could occur, thus limiting the applicability of these methods to a variety of problems. Facing these concerns, this article proposes an evolutionary algorithm named FGTTMP based on feedback-guided transfer (FGT) and trend manifold prediction (TMP) for solving DMOPs. The FGT employs an information feedback model to extract valuable knowledge using all historical environments and then identifies excellent individuals using cluster-transfer learning. This can both accelerate convergence and introduce diversity for future environments. The TMP applies the probability-based trend prediction method to estimate the mass center of the whole population and the corresponding manifold, relying on the two previous moments. Thus, the FGT and TMP strategies combine historical knowledge with prediction techniques, synergistically leading the population in promising directions. The performance of the proposed algorithm is fully investigated and compared with eight state-of-the-art algorithms. Experimental results demonstrate that the proposed FGTTMP method can achieve better convergence and diversity on 19 various benchmark problems than the state-of-the-art algorithms.**

***Index Terms*—Dynamic multiobjective optimization, feedback-guided transfer (FGT), knowledge, trend manifold prediction (TMP).**

I. Introduction

DYNAMIC multiobjective optimization problems (DMOPs) represent a class of problems whose objectives conflict with each other and change over time. The DMOPs are very common in many realistic scenarios [1], including wastewater treatment [2], image classification [3], and job-shop scheduling [4]. For instance, to increase the exploitation rate of nonrenewable resources in the raw ore allocation optimization, it is needed to maximize the equipment capability and minimize runtime under the dynamic model parameters and constraints [5]. In contrast to the static multiobjective problems (MOPs), the most significant characteristic of DMOPs is that the number of objectives varies with time, which is attributed to constant changes in time or an environment. Thus, a DMOP can be regarded as multiple consecutive MOPs addressed separately. Once a variation in an environment is detected, the initialization is imposed on the entire population. Although this approach can simplify the complexity of solving DMOPs, it is a challenging task to design dynamic response mechanisms to handle environmental changes.

In the past few years, various multiobjective evolutionary algorithms (MOEAs) [6] have been widely used for solving static MOPs, such as nondominated sorting genetic algorithm II (NSGA-II) [7] and regularity model-based multiobjective estimation of distribution algorithm (RM-MEDA) [8], achieving high efficiency and practicality. However, DMOPs require that an algorithm can trace the Pareto-optimal set (POS) and Pareto-optimal front (POF) precisely and rapidly when monitoring environmental changes. Therefore, it could be highly effective to combine MOEAs and dynamic response strategies to handle DMOPs. Accordingly, various dynamic MOEAs (DMOEAs) have been presented in recent years, including the diversity-, memory-, and prediction-based approaches. Diversity-based approaches address the diversity loss when changes occur, whereas memory-based approaches directly reuse historical special solutions or nondominated solutions to accelerate convergence. Recently, prediction-based approaches have received great attention because they require data on only several historical time steps, and some of the change patterns are similar and predictable. With the wide applicability in machine learning, autoregression [9], sampling [10], and grey prediction [11], these approaches have been used for the prediction of new populations.

The aforementioned approaches, particularly the prediction-based methods, can perform well in solving a variety of benchmark problems. However, they have certain limitations with them. First, when environmental changes are drastic and rapid, previously acquired POS may deviate from the true POS. Therefore, directly reusing the past individuals

Manuscript received 12 November 2023; revised 26 March 2024; accepted 8 August 2024. Date of publication 6 September 2024; date of current version 20 November 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62473350, and in part by the Fundamental Research Funds for the Central Universities. This article was recommended by Associate Editor J. Bi. *(Corresponding author: Gai-Ge Wang.)*

Yong Wang, Kuichao Li, and Gai-Ge Wang are with the School of Computer Science and Technology, Ocean University of China, Qingdao 266101, China (e-mail: gaigewang@gmail.com).

Dunwei Gong is with the School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266101, China.

Keqin Li is with the Department of Computer Science, State University of New York at New Paltz, New Paltz, NY 12561 USA.

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TSMC.2024.3443143.

Digital Object Identifier 10.1109/TSMC.2024.3443143

might mislead the subsequent search process. Second, most prediction-based approaches mainly employ linear prediction models that cannot accurately predict individuals in new environments if the POS of a DMOP exhibits nonlinear correlation at different moments. Third, the majority of available methods for generating new solutions are based on two or three previous moments' solutions, but in reality, employing historical data on only the adjacent times might result in overlooking valuable information at earlier moments. Finally, at the end of each iteration, the number of optimal solutions is finite compared with the entire search area; thus, it is challenging to derive all useful knowledge from limited historical individuals.

To address the above-mentioned challenges, this article proposes an evolutionary algorithm based on the feedback-guided transfer (FGT) and trend manifold prediction (TMP), called FGTTMP. The FGTTMP integrates valuable information from all previous iterations and adopts prediction techniques to generate high-quality initial populations. In the FGTTMP, the FGT strategy extracts the features of the optimal solutions using an information feedback model and then assimilates the features through cluster-based transfer learning; thus, a more effective and precise knowledge extraction model can be constructed to solve DMOPs. The TMP strategy reduces the prediction deviation of the linear model as much as possible by means of a probability-based trend prediction method and explores the most promising movement direction for the center of mass, which is of great help to the evaluation of manifolds and nondominated solutions.

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

1) A knowledge extraction model is constructed using an information feedback model and clusters-based transfer learning. In this way, valuable historical information from all previous search processes can be preserved, which speeds up convergence and enhances diversity in the future time period.
2) To decrease the prediction deviation of the linear model, this study develops a probability-based trend prediction model to relocate the center point. Therefore, the estimated center of mass and manifold can together constitute nondominated solutions in a new environment.
3) The results of comprehensive comparative experiments on 19 benchmarks show that the proposed FGTTMP performs better than the other algorithms and can effectively promote convergence and improve diversity.

The remainder of this study is structured as follows. Section II discusses the concepts related to DMOPs, the information feedback model, and available DMOEAs. Section III introduces the proposed FGTTMP method. Section IV presents and analyzes the experimental results. Finally, Section V draws conclusions and gives future work directions.

II. Preliminaries and Related Work

A. Dynamic Multiobjective Optimization Problems

Generally, a DMOP can be treated as a minimization problem to be optimized. Therefore, the mathematical expression of a DMOP is as follows:

$$\begin{aligned} \text{Minimize} \quad & F(x,t) = \langle f_1(x,t), f_2(x,t), \ldots, f_m(x,t)\rangle \\ \text{s.t.} \quad & x \in \Omega \end{aligned} \tag{1}$$

where $x = (x^1, x^2, \ldots, x^n)$ is the n-dimensional decision vector in a decision space Ω; t refers to dynamic factors, such as time or environmental variables; f_i is the ith objective function; and $F(x, t)$ represents the objective vector consisting of m objective functions.

Definition 1 (Pareto Domination [12], [13]): For two decision vectors x_1 and x_2 at a time t, x_1 is deemed to Pareto dominate x_2, which is expressed as $x_1 \succ_t x_2$, if and only if

$$\begin{cases} \forall i = 1, \ldots, m, \; f_i(x_1, t) \leq f_i(x_2, t) \\ \exists i = 1, \ldots, m, \; f_i(x_1, t) < f_i(x_2, t). \end{cases} \tag{2}$$

Definition 2 [Dynamic POS (DPOS) [12], [13]]: Assume that x^* and x are decision vectors. A decision vector x^* is considered nondominated (Pareto optimal) at a time t if and only if it is not dominated by another decision vector x. A DPOS is the set of all Pareto-optimal solutions, which is defined as follows:

$$\text{DPOS}_t = \{x^* \in \Omega | \neg\exists x \in \Omega, x \succ_t x^*\}. \tag{3}$$

Definition 3 [Dynamic POF (DPOF) [12], [13]]: At a time t, a DPOF represents the corresponding objective vector of a DPOS, which is given by

$$\text{DPOF}_t = \{\text{F}\big(x^*, t\big) | x^* \in \text{DPOS}_t\}. \tag{4}$$

The DMOEAs are committed to obtaining a POS at different moments that not only converges to the corresponding POF as much as possible but also has good diversity. The DMOPs can be roughly categorized into four classes according to the variation characteristics of POS and POF as follows.

Type I: POS varies over time, but POF is stationary.

Type II: Both POS and POF vary over time.

Type III: POS is stationary yet POF is dynamic.

Type IV: Both POS and POF are stationary.

Problems of type IV are not characterized by dynamic changes, and thus, this type is not considered in this study.

B. Information Feedback Model

In the literature, the information feedback model [14] was employed to reuse useful historical knowledge to lead the subsequent search. In recent years, the information feedback models have received great attention and achieved wide application, and their numerous variants have been applied to address different optimization problems. For instance, Han et al. [15] improved the offspring generation process by exploiting historical information; Pan et al. [16] extracted historical knowledge to weigh the total power expenditure and delay simultaneously using problem-specific information and a random heuristic. Moreover, Pan et al. [17] adopted the feedback scheme to adjust the population size dynamically. In particular, in [18], the information feedback model has been integrated into multiobjective algorithms to improve the iterative process of a many-objective, large-scale optimization.

Based on the above investigation, an information feedback model could be used to address DMOPs by the extraction and integration of all historical individuals. However, to the best of the authors' knowledge, this is the first time that an information feedback model has been applied to the DMOEA field. In general, most memory-based DMOEAs directly reuse nondominated or special individuals from only one or two previous generations. Thus, solutions from earlier generations are ignored, which causes a great loss for future evolutionary processes. Different from the original information feedback model, in the model proposed in this study, all historical POS are used to extract information and generate new solutions. On this basis, solutions with good convergence and distributivity are further filtered using cluster-based transfer learning. There are two knowledge extraction types in the information feedback model, namely, fixed and random extraction mechanisms. Aiming to accelerate the convergence of subsequent evolutions, this study selects fixed extraction.

Suppose that x_i^j is the ith solution at a moment j ($1 \leq i \leq N$, $1 \leq j \leq t-1$), where N is the population size, and t indicates the present moment. Based on the fixed extraction mechanism, the ith solution obtained through historical knowledge extraction $\text{new\_}x_i^t$ at a moment t can be expressed by

$$\text{new\_}x_i^t = \alpha_1 x_i^1 + \alpha_2 x_i^2 + \cdots + \alpha_j x_i^j \tag{5}$$

where $\alpha_1, \alpha_2, \ldots, \alpha_j$ are the fitness scale factors satisfying the condition of $\alpha_1 + \alpha_2 + \cdots + \alpha_j = 1$, and they can be calculated by

$$\alpha_1 = \frac{1}{\beta} \cdot \frac{\text{fitness\_sum}_i - \text{fitness}_i^1}{\text{fitness\_sum}_i}$$
$$\alpha_2 = \frac{1}{\beta} \cdot \frac{\text{fitness\_sum}_i - \text{fitness}_i^2}{\text{fitness\_sum}_i}$$
$$\vdots$$
$$\alpha_j = \frac{1}{\beta} \cdot \frac{\text{fitness\_sum}_i - \text{fitness}_i^j}{\text{fitness\_sum}_i} \tag{6}$$

where β is set to $(t-2)$; fitness_i^j represents the fitness value of x_i^j; fitness_sum_i is the fitness sum of the ith position in all previous times, which can be expressed by

$$\text{fitness\_sum}_i = \sum_{j=1}^{t-1} \text{fitness}_i^j. \tag{7}$$

To ensure that all generated individuals are within decision vector boundaries, this study adopts a boundary check paradigm, defined as follows:

$$y_i = \begin{cases} \text{sol}_i, l_i \leq \text{sol}_i \leq u_i \\ \text{rand}(l_i, 0.5 \times (l_i + u_i)), \text{sol}_i < l_i \\ \text{rand}(0.5 \times (l_i + u_i), u_i), \text{sol}_i > u_i \end{cases} \tag{8}$$

where rand(a, b) denotes a randomly generated number between a and b, excluding the boundaries of a and b. l_i and u_i ($i = 1, 2, \ldots, n$) are the upper and lower boundaries of an individual sol_i, respectively. The boundary correction is carried out to obtain the individual with the correct range y_i.

C. Related Work

Over the past few decades, much effort has been dedicated to developing effective and efficient DMOEAs to solve DMOPs. The existing approaches are categorized into three main types: diversity-, memory-, and prediction-based approaches [19].

The diversity-based approaches are committed to balancing diversity and convergence, avoiding populations falling into a local optimum when encountering environmental changes. In [20], an innovative novel precision controllable mutation was employed to control the mutation level of individuals. A mixture of steady-state and generative regulations was proposed in [21] to cope with environmental variations. When the variation is monitored, partially obsolete individuals with good distribution are regained to constitute the initial population. In [22], the simulated isotropic magnetic particle niching was used to obtain the solution to maintain homogeneous distribution; therefore, the well-diverse population could be maintained during the entire evolution. Chen et al. [23] proposed a response mechanism that combines partially stored solutions with randomly generated solutions to constitute the initial population. Hu et al. [24] introduced a subspace-based diversity maintenance strategy, which can identify interindividual gaps and employ a gap-filling technique to enhance population diversity.

The memory-based approaches use extra memory to store excellent historical individuals and then repurpose them for new environments. Chen et al. [25] developed an algorithm that can store two subpopulations, one focusing on convergence and the other on diversity, which evolve synergistically. In [26], a memory-based method was used to locate new positions of population members when environmental changes were analogous to changes in history. Recently, Zhao et al. [27] proposed an extraction mechanism of previous information, which can reduce the inaccuracy of the predictors and promote convergence. In general, historical data on individuals can be processed and reused when an environment changes similarly over time. However, the performance of memory-based methods decreases dramatically when environments change differently.

The prediction-based approaches use data on previous individuals to relocate the population using a predictive model when the pattern of environmental changes is predictable [28], [29]. Currently, the introduction of machine learning methods into DMOEAs to predict individuals for new environments has attracted great attention [6], [30]. Zhou et al. [9] introduced a center-of-mass-based prediction mechanism, where the autoregression model was used to assess the center of mass and the corresponding manifold for the entire population. Muruganantham et al. [31] developed the Kalman filter-based prediction strategy, which can learn from historical data, determine changing patterns, and predict new individuals. Furthermore, to evaluate the locations of individuals more precisely, multiple representative individuals were selected to locate populations from multiple directions [32]. In [33], a predictive model using a Gaussian inverse process was presented to map individuals from the objective space to the decision space. In addition, other prediction methods have been devised, such as correlation-guided layered prediction [34] and mixture-of-experts prediction [35].

III. Proposed FGTTMP

The flowchart of the proposed FGTTMP is presented in Fig. 1, and its pseudocode is shown in Algorithm 1. As shown

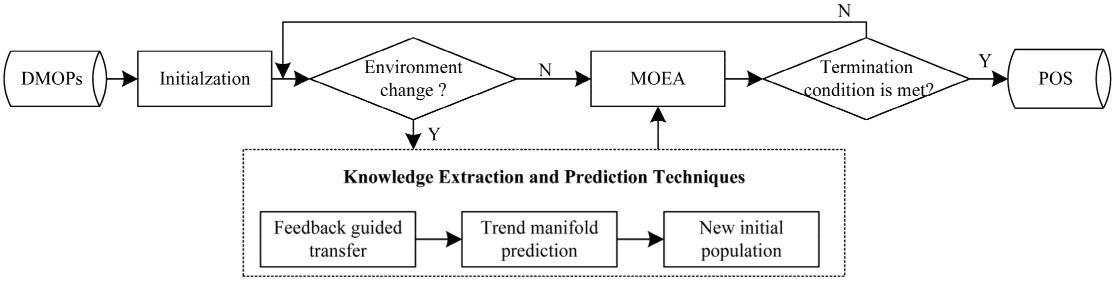

DMOPs
Initialzation
Environment change ?
N
Y
MOEA
Termination condition is met?
N
Y
POS
Knowledge Extraction and Prediction Techniques
Feedback guided transfer
Trend manifold prediction
New initial population

Fig. 1. Flowchart of the proposed FGTTMP.

Algorithm 1 FGTTMP Algorithm

```
Input: Dynamic optimization function $F(x, t)$, population size $N$, and number of clusters $K$;
Output: A set of POS and POF in different environments $PS$ and $PF$.
Initialization;
while an environment changes do
If $t == 1|| t == 2$ then
    Initialize randomly the population $initPop$;
    $[POS_t, POF_t] = \mathbf{SMOA}(F(x, t), initPop)$;
    $PS = PS \cup POS_t$;
    $PF = PF \cup POF_t$;
else
    $Pop_1 = \mathbf{FGT}(PS, PF, POS_{t-1}, K, N)$;
    $Pop_2 = \mathbf{TMP}(POS_{t-2}, POS_{t-1})$;
    $initPop = Pop_1 \cup Pop_2$;
    $[POS_t, POF_t] = \mathbf{SMOA}(F(x, t), initPop)$;
    $PS = PS \cup POS_t$;
    $PF = PF \cup POF_t$;
end if
$t = t + 1$;
end while
```

in Algorithm 1, the proposed FGTTMP is composed of the static MOEA (SMOEA), FGT process, and TMP process. First, initialization and SMOEA are imposed on the population for optimization in the first two environments. Then, if there is a change in the environment, the FGT and TMP processes are performed to generate a new population. Based on the POS and POF of all previous environments, the FGT mechanism applies the information feedback model generator to generate N new individuals. Afterward, well-converging clusters are identified among the K clusters using a transfer learning technique and denoted as the first subinitial population Pop_1. In addition, since the POS movement is affected by both the center of mass and manifold, the TMP mechanism is employed to track the POS. Using a probability-based prediction model, the movement direction and manifold change in the nondominated solutions can be estimated, and they are denoted as the second subinitial subpopulation Pop_2 to maintain diversity. Finally, Pop_1 and Pop_2 are merged into the initial population $initPop$, thus jointly guiding the search toward the true POF. The proposed FGT and TMP strategies are introduced in detail in the following.

Algorithm 2 FGT Strategy

```
Input: POS and POF for all historical environments, which are denoted by $PS$ and $PF$, respectively; a Pareto-optimal set $POS_{t-1}$ at time $(t - 1)$; number of clusters $K$, and population size $N$.
Output: Individuals generated by the feedback-guided transfer $Pop_1$.
$Pop = \varnothing, \ k = N$;
for $i = 1$ to $N$
for $j = 1$ to $t - 1$
    Calculate the fitness sum $fitness\_sum_i$ by (7);
    Calculate the coefficient $\alpha_j$ by (6);
end for
Generate new individuals new_$x_i^t$ by (5);
$Pop = Pop \cup$ new_$x_i^t$;
end for
 Correct the boundaries of $Pop$ by (8);
 Set $x_i^t \in Pop$ as a cluster $C_i$ with the center point $c_i$;
 while $k > K$
 Find the two nearest clusters $C_i$ and $C_j$ by (10);
 Merge $C_i$ and $C_j$ and remove $C_j$;
 Recalculate the center point $c_i$ by (9);
 $k - -$;
 end while
 $c = c_1 \cup c_2 \cup \cdots \cup c_K$;
 $(F_1, F_2, \ldots, F_p) = FastNondominatedSort(c)$;
 Select clusters with $c \in F_1$ as a target domain TD, $POS_{t-1}$ as the source domain SD;
 Initialize $w_1$ by (11);
 for $i = 1$ to $M$
     Call SVM with TD $\cup$ SD and $w_i$ to obtain a weak classifier $h_w^i$;
     Update $w_{i+1}$ by (15);
 end for
 Obtain a strong classifier $h_s$ by (16);
 Generate numerous random solutions $x_{rand}$;
 $Pop_1 = \{x | h_s(x) = 1, x \in x_{rand}\}$;
```

A. Feedback-Guided Transfer

As mentioned above, the purpose of the FGT is to exploit all valuable information from historical environmental data, guiding the evolution toward promising search regions. Currently, most memory-based approaches reuse nondominated solutions of historical environments or conduct similar operations, which are efficient for periodic environmental changes. However, in real environments, many changes are nonperiodic or acyclic. In view of that, this section introduces an advanced information feedback model-based clustering transfer that can effectively integrate useful and potential information from all historical environments.

The pseudocode of the FGT strategy is shown in Algorithm 2, where its inputs include POS and POF for all historical environments denoted by *PS* and *PF*, respectively, POS POS_{t-1} at a time $(t - 1)$, the number of clusters K, and a population size N. In the initial environment, set *Pop* is initialized to empty, reserving generative solutions using the information feedback model. Then, in lines 2–9, *Pop* is obtained by performing the information feedback model on *PS* and *PF*. First, the sum of fitness $\mathit{fitness\_sum}_i$ from one to $(t-1)$ times is obtained by (7). Then, the fitness scale factor α_i for each solution in the population is computed using (6). In this case, a solution at time t, denoted by $\text{new}\_x_i^t$, is obtained using (5) based on information from all historical solutions. By performing the aforementioned process, N solutions with feedback are generated. Specifically, the boundary correction operator is used to correct the boundaries of generative solutions, as given in (8).

Although the information feedback-based model can produce many promising solutions, to achieve better subsequent evolutions, the FGT strategy employs cluster-based transfer learning to identify superior solutions with better convergence and diversity. The main idea is to obtain the clusters with center points located on the first frontier. Therefore, the hierarchical clustering method [30] is used to cluster the obtained solutions. First, as given in line 11, the hierarchical clustering method initially considers each x_i^t as a cluster C_i and then iteratively merges similar clusters into one cluster. The center point of a cluster C_i can be obtained by

$$c_i = \frac{1}{N} \sum_{\text{new}\_x_i^t \in \text{Pop}} \text{new}\_x_i^t. \tag{9}$$

Furthermore, the similarity between clusters C_i and C_j is expressed as

$$\text{Similarity}_{C_i,C_j} = \sqrt{\sum_{d=1}^{n} \left(c_i^d, c_j^d\right)^2} \tag{10}$$

where c_i^d and c_j^d are the dth dimensions of c_i and c_j, respectively. Then, the two most similar clusters C_i and C_j are merged into cluster C_i, and cluster C_j is removed, as shown in line 14. Subsequently, the corresponding center point of the new cluster is updated by (9). Finally, K clusters are obtained by iteratively running lines 12–17.

To determine which clusters are available, their center points are integrated into a set c and ranked into p layers $(F_1, F_2, \ldots, F_p)$, lines 18 and 19, using the nondominated sorting [7]. Then, clusters whose center points are positioned in layer F_1 are recognized as valuable knowledge and integrated into the target domain TD; this can lead the search process to converge more rapidly and precisely. Meanwhile, POS_{t-1} is considered a source domain SD. For each domain, the nondominated solutions are marked as "1" and the remaining solutions are marked as "–1," which is expressed by $L(x) : x \in \text{SD} \cup \text{TD} \rightarrow y, y \in \{1, -1\}$. The support vector machine (SVM) is employed to train iteratively the weights and parameters of M weak classifiers h_w^i, which constitute a strong classifier h_s for discriminating good solutions. The detailed procedure of transfer learning [36] is as follows.

First, the weights of all solutions in the SD and TD are initialized as follows:

$$w_1 = \begin{cases} \frac{1}{|\text{SD}|}, x \in \text{SD} \\ \frac{1}{|\text{TD}|}, x \in \text{TD} \end{cases} \tag{11}$$

where SD and TD denote the source and target domains, respectively; |SD| and |TD| are the cardinality of SD and TD, respectively.

Next, in the ith training $(i = 1, 2, \ldots, M)$, the factor ε_i of h_w^i on TD is calculated by

$$\varepsilon_i = \sum_{x \in \text{TD}} \frac{w_i \cdot |h_w^i - L(x)|}{\sum_{x \in \text{TD}} w_i}. \tag{12}$$

The weight coefficients of SD and TD can be formulated as follows:

$$\beta = \frac{1}{2} \ln\left(\frac{1}{1 + \sqrt{2 \ln M}}\right) \tag{13}$$

and

$$\beta_i = \frac{1}{2} \ln \frac{1 - \varepsilon_i}{\varepsilon_i}. \tag{14}$$

Therefore, the weights of solutions in the SD and TD can be updated by

$$w_{i+1} = \begin{cases} w_i \cdot e^{\beta \cdot |h_w^i - L(x)|}, x \in \text{SD} \\ w_i \cdot e^{\beta_i \cdot |h_w^i - L(x)|}, x \in \text{TD}. \end{cases} \tag{15}$$

The weights of solutions approaching the TD will progressively increase, indicating that the recognition ability of the weak classifiers becomes more accurate. Afterward, M weak classifiers are used to construct a strong classifier h_s when the stopping condition is satisfied, which is expressed as follows:

$$h_s = \text{sign}\left(\sum_{i=1}^{M} \beta_i h_w^i\right). \tag{16}$$

After obtaining h_s, numerous random solutions x_{rand} are generated in line 27. Finally, x_{rand} is fed to h_s and solutions identified as "1" constitute the subinitial population Pop_1. The workflow of the FGT strategy is shown in Fig. 2.

B. Trend Manifold Prediction

According to [37], under moderate conditions, the POS of a continuous MOP with m objectives can be considered a piecewise continuous manifold with $(m - 1)$ dimensions. Therefore, the POS^t can be partitioned into two components: 1) a center point c^t and 2) a manifold $\widetilde{m}^t$ at time t, which can be expressed as follows:

$$\text{POS}^t = c^t + \widetilde{m}^t \tag{17}$$

where c^t is calculated by (9). Each solution $x^t \in \text{POS}^t$ can be defined as follows:

$$x^t = c^t + \widetilde{x}^t \tag{18}$$

where $\widetilde{m}^t = \{\widetilde{x}^t\}$. Thus, as long as the center point and manifold of POS^t can be predicted at time t, the POS^t can be obtained.

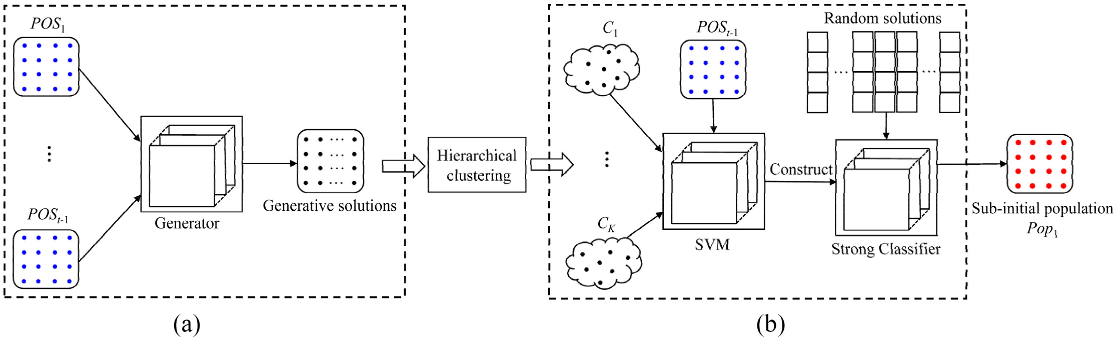


Fig. 2. Schematic of the FGT strategy: (a) POSs of all historical environments are fed into an information feedback model-based generator to generate solutions; (b) generative solutions are divided into K clusters using hierarchical clustering; then, the clusters in the first frontier are denoted as a target domain, and POS_{t-1} is set as a source domain, and they are fed to the SVM to train numerous weak classifiers, which together construct a strong classifier that can identify good solutions, which construct a subinitial population Pop_1.

In the TMP strategy, the approximate values $\widetilde{m}^{t-2}$ and $\widetilde{m}^{t-1}$ of the first two times are archived to evaluate the manifold at time t. Particularly, each point is employed to predict a new point according to

$$\widetilde{x}^t = \widetilde{x}^{t-1} + \xi(0, \sigma) \tag{19}$$

where variable σ is calculated by

$$\sigma = \frac{1}{N}\text{dis}\left(\widetilde{m}^{t-2}, \widetilde{m}^{t-1}\right) \tag{20}$$

and $\text{dis}(\widetilde{m}^{t-2}, \widetilde{m}^{t-1})$ is calculated by

$$\text{dis}\left(\widetilde{m}^{t-2}, \widetilde{m}^{t-1}\right) = \frac{1}{|\widetilde{m}^{t-2}|} \sum_{\widetilde{x}_1^{t-2} \in \widetilde{m}^{t-2}} \text{min}_{\widetilde{x}_2^{t-1} \in \widetilde{m}^{t-1}} ||\widetilde{x}_1^{t-2} - \widetilde{x}_2^{t-1}|| \tag{21}$$

where $|\widetilde{m}^{t-2}|$ is the cardinality of $\widetilde{m}^{t-2}$, and $\|.\|$ represents the Euclidean distance. Finally, the predicted manifold $\widetilde{m}_{\text{pre}} = \{\widetilde{x}^t\}$ is obtained.

The center point at the next time can be predicted by a probability density function-based method [37]. The specific steps are as follows. First, based on the movement direction of c^{t-1} and c^{t-2} at the previous two times, the n-dimensional direction vector $v = (v^1, v^2, \ldots, v^n)$ can be obtained by $v = c^{t-1} - c^{t-2}$. In the polar coordinate system, v can be expressed as $v = (\varphi^1, \varphi^2, \ldots, \varphi^{n-1}, |v|)$, where $\varphi^i (1 \leq i \leq n)$ denotes the angular coordinate, and $|v|$ is the modulus of v. Then, the ith correlative angle of v can be calculated by

$$\varphi^i = \arctan\left(\frac{\sqrt{\sum_{d=j+1}^{n} (v^d)^2}}{v^i}\right). \tag{22}$$

Second, for a given direction vector v^i, the probability density function is used to determine the deflection angle with the greatest probability, which is performed using the proposed algorithm for each angle coordinate as follows:

$$\theta^i \sim \text{TMP}\left(v^i\right) = \frac{e^{-\text{sign}(\theta^i)\cdot \frac{\theta i}{v^i}}}{\int_{-\pi}^{0} e^{\frac{\theta^i}{|v^i|}} d\theta^i + \int_{0}^{\pi} {}^{-\frac{\theta^i}{v^i}} d\theta^i} \tag{23}$$

where θ^i is a deflection angle, and the closer θ^i is to zero, the more likely the center of mass is to move.

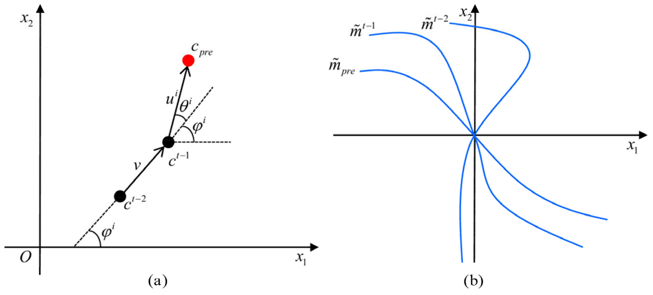


Fig. 3. Schematic of the TMP. (a) Center point is estimated at the next time using the trend prediction method. (b) Manifold is predicted based on the manifolds of the previous two times.

Once θ^i is determined, the predicted direction vector u^i can be calculated by

$$u^i = \begin{cases} |v|\cos\left(\varphi^1 + \theta^1\right), i = 1 \\ |v| \prod_{d=1}^{i-2} \sin\left(\varphi^d + \theta^d\right)\cos\left(\varphi^{i-1} + \theta^{i-1}\right), 1 < i < n \\ |v| \prod_{d=1}^{i-1} \sin\left(\varphi^d + \theta^d\right), i = n. \end{cases} \tag{24}$$

Finally, the predicted center point c_{pre} can be obtained by

$$c_{\text{pre}} = c^{t-1} + u. \tag{25}$$

By substituting the predicted center point c_{pre} and manifold $\widetilde{m}_{\text{pre}}$ into (17), the predicted POS can be obtained, which can significantly accelerate the convergence of the algorithm. The pseudocode of the TMP strategy is shown in Algorithm 3. In addition, to understand the idea of the TMP strategy better, the schematic of the TMP strategy is presented in Fig. 3.

C. Computational Complexity

The computational complexity in the proposed FGTTMP algorithm is mainly defined by the FGT and TMP strategies. Assume that N is the population size, t is the time variable, M is the iterative number for the transfer learning, and n is the number of decision variables. In the FGT module, the information feedback model is used to extract knowledge from all historical solutions, and its computational complexity is $O(Nt)$; the computational complexity of the cluster-based transfer learning is $O(MN^2n)$. In the TMP module, the probability-based prediction method is used to estimate the center point and manifold, having the computational complexity of $O(N)$. Thus, the computational complexity of the

Algorithm 3 TMP Strategy

```
Input: Pareto-optimal sets $POS_{t-2}$ and $POS_{t-1}$ corresponding to times (t – 2) and (t – 1);
Output: Individuals generated by the TMP $Pop_2$;
Obtain center points of $POS_{t-2}$ and $POS_{t-1}$ and denote them by $c^{t-2}$ and $c^{t-1}$ respectively, by (9);
Calculate manifolds $\widetilde{m}^{t-2}$ and $\widetilde{m}^{t-1}$ corresponding to times (t – 2) and (t – 1) by (17);
Estimate the manifold $\widetilde{m}_{\text{pre}}$ by (19);
for $i = 1$ to $n - 1$
    $v = c^{t-1} - c^{t-2}$;
    Calculate the angular coordinate $\varphi^i$ by (22);
    Obtain the deflection angle $\theta^i$ by (23);
end for
for $i = 1$ to $n$
    Compute the deflection vector $u^i$ by (24);
end for
Estimate the center point $c_{\text{pre}}$ by (25);
Obtain $Pop_2$ by (17);
```

proposed FGTTMP is calculated as $O(Nt) + O(MN^2n) + O(N) = O(MN^2n)$.

IV. Experimental Results

A. Benchmark Problems and Comparative Algorithms

Problems with dynamically changing characteristics can systematically determine and assess the behavior of algorithms. In this study, four benchmark test suites were used, namely, the dMOP [38], FDA [39], F [9], and DF [40], for a total of 19 benchmark problems. The dMOP and FDA are classic and the earliest test suites; the F test suite contains complex dynamic features; the DF test suite has been recently proposed to provide many new dynamic properties, such as degenerate POF, irregular POF shapes, disconnected POF segments, and location of optima.

To verify the superiority of the proposed FGTTMP, eight state-of-the-art algorithms were selected for comparative experiments, namely, PPS [9], CKPS [41], MMTL [42], IT [36], GM [11], AE [43], KT, and CPDM [44]. This study focuses on how to handle changes in dynamic environments since any MOEA can be used to handle static environments. For the purpose of a fair comparison, the FGTTMP and the comparative algorithms all used RM-MEDA [8] as SMOEA. In addition, all comparative algorithms were parameterized following the configurations of the original paper. The dynamic factor of a DMOEA was specified as $t = (1/n_t)\lfloor \tau/\tau_t \rfloor$, where τ, n_t, and τ_t denote the maximum generation, the change severity, and the change frequency, respectively. A smaller n_t implies a more drastic environmental variation, and a smaller τ_t implies a faster environmental variation.

B. Performance Metrics and Settings

Performance metrics can precisely measure and reflect the performance of algorithms, regardless of both diversity and convergence. In this study, four performance metrics were used: 1) inverted generational distance (IGD); 2) mean IGD (MIGD); 3) DMIGD; and 4) mean hypervolume (MHV).

TABLE I
Parameter Settings

Parameter	Value
Number of decision variables n	10
Population size N	200
Dynamic settings	$\tau_t = 5, 10$ $n_t = 10, 15, 20, 30$
Number of changes	50
Number of generations	$50\tau_t$
Number of clusters K	12

TABLE II
Comparison Statistics on the F, DMOP, and FDA Problems

MIGD	+/-/=	MHV	+/-/=
FGTTMP vs PPS	39/0/1	FGTTMP vs PPS	38/0/2
FGTTMP vs CKPS	40/0/0	FGTTMP vs CKPS	40/0/0
FGTTMP vs MMTL	40/0/0	FGTTMP vs MMTL	40/0/0
FGTTMP vs IT	40/0/0	FGTTMP vs IT	40/0/0
FGTTMP vs GM	34/0/6	FGTTMP vs GM	32/4/4
FGTTMP vs AE	36/3/1	FGTTMP vs AE	35/3/2
FGTTMP vs KT	39/1/0	FGTTMP vs KT	38/0/2
FGTTMP vs CPDM	39/1/0	FGTTMP vs CPDM	40/0/0

TABLE III
Comparison Statistics on the DF Problems

MIGD	+/-/=	MHV	+/-/=
FGTTMP vs PPS	29/6/1	FGTTMP vs PPS	20/9/7
FGTTMP vs CKPS	34/2/0	FGTTMP vs CKPS	30/1/5
FGTTMP vs MMTL	35/0/1	FGTTMP vs MMTL	32/0/4
FGTTMP vs IT	36/0/0	FGTTMP vs IT	36/0/0
FGTTMP vs GM	26/7/3	FGTTMP vs GM	11/9/16
FGTTMP vs AE	31/4/1	FGTTMP vs AE	20/12/4
FGTTMP vs KT	22/14/0	FGTTMP vs KT	26/6/4
FGTTMP vs CPDM	34/2/0	FGTTMP vs CPDM	28/2/6

Regarding the experimental parameters, the number of decision variables was set to $n = 10$, and the population size was set to $N = 200$ for all benchmarks regardless of the number of objectives in the problem (i.e., for both two- and tri-objective problems). In the proposed FGTTMP, the number of clusters K was set to 12, and relevant parameters of the SVM were set by default [36]. All benchmark problems were run independently ten times. Five environmental configurations were adopted: ($n_t = 10$, $\tau_t = 5$), ($n_t = 15$, $\tau_t = 5$), ($n_t = 20$, $\tau_t = 5$), ($n_t = 30$, $\tau_t = 5$), and ($n_t = 10$, $\tau_t = 10$); τ was set to $50\tau_t$, which meant that there were 50 environment changes in each run. All algorithms were implemented in MATLAB R2020b software. All the key parameters are displayed in Table I.

C. Experimental Results and Analysis

To investigate the capacity to cope with various change severities in dynamic environments, τ_t was set to five, and n_t was set to 10, 15, 20, and 30. The statistics results of the MIGD, MHV, and DMIGD metrics are presented in Tables II–IV and Tables A.I–A.IV in the Supplementary Material. Furthermore, the Wilcoxon rank-sum test at the 0.05 confidence level was implemented to confirm a significant difference. In Tables II–IV, "+," "–," and "=" indicate the

TABLE IV
DMIGD Results of Different Algorithms

Problem	PPS	CKPS	MMTL	IT	GM	AE	KT	CPDM	FGTTMP
F5	2.1545	2.3553	2.7776	1.9428	0.7175	1.8169	1.2918	2.4817	**0.5026**
F6	1.3584	1.5397	1.5586	1.4630	0.6761	1.1554	1.1348	1.8308	**0.3757**
F7	1.3148	1.6311	2.0862	1.4258	0.6262	1.1213	0.8777	1.9319	**0.3356**
F9	3.4032	2.1070	2.7007	1.7209	1.1544	1.4903	1.4085	2.3676	**0.9574**
F10	3.6185	3.6308	3.8999	3.2358	3.1646	3.5099	3.1787	3.7102	**3.0041**
dMOP1	0.1977	0.3505	0.1963	0.3388	0.0659	0.1617	0.0830	0.4593	**0.0522**
dMOP2_dec	1.4224	1.7183	1.3666	1.7171	0.6947	1.3289	0.7217	2.2078	**0.5095**
FDA1	0.2815	0.3557	0.2660	0.3339	0.0504	0.1268	0.1676	0.5224	**0.0417**
FDA2	0.0350	0.0714	0.0977	0.0875	0.0286	**0.0197**	0.0770	0.0939	0.0257
FDA3	0.1315	0.5100	0.1774	0.3222	0.0518	0.0964	0.1463	0.3730	**0.0439**
DF3	0.4143	0.4263	0.4002	0.4253	0.2243	0.3344	0.3324	0.5181	**0.2241**
DF4	1.2205	1.4801	1.2636	1.4492	1.1194	1.2121	1.2006	1.5970	**1.1081**
DF6	7.7071	6.8674	3.6463	5.5948	3.9009	6.2678	**2.1382**	5.8975	3.0043
DF7	7.2908	5.9322	3.2338	4.7867	3.8195	5.1657	**2.1630**	5.3245	2.8072
DF9	1.9227	2.0487	1.9786	2.0755	1.8050	1.8813	2.1946	1.9426	**1.8026**
DF11	0.2666	0.1566	0.4090	0.2960	0.3670	0.3492	0.4135	0.5063	**0.1764**
DF12	1.0293	0.9646	0.9908	0.9795	0.9803	0.9540	0.9076	0.9956	**0.8158**
DF13	1.1798	**1.1417**	1.8574	1.5723	1.3389	1.2487	1.2079	1.6464	1.2895
DF14	0.6524	0.7582	0.8672	1.0185	0.6751	0.6358	**0.5962**	0.9658	0.7196

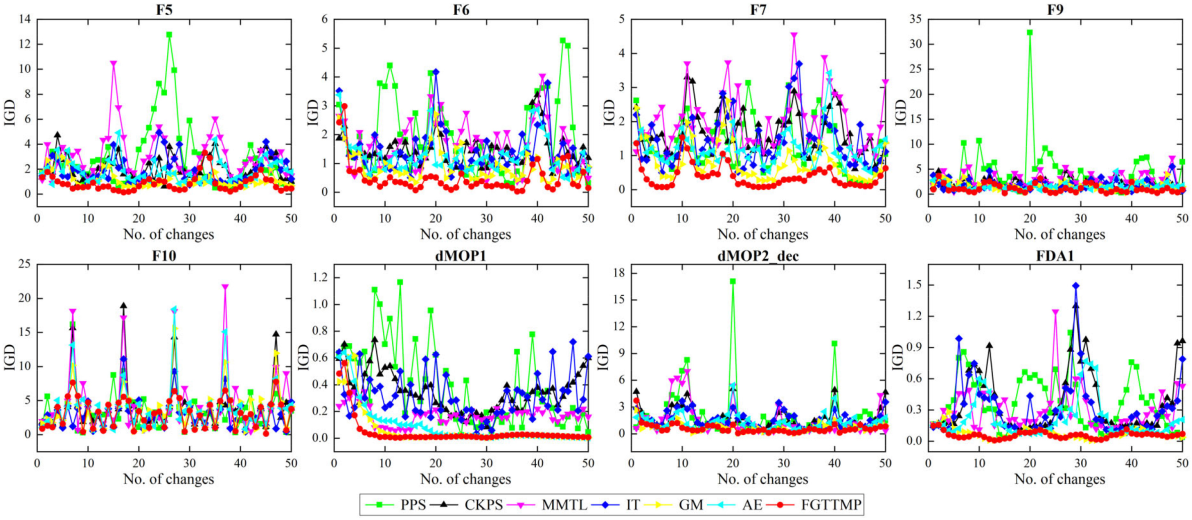


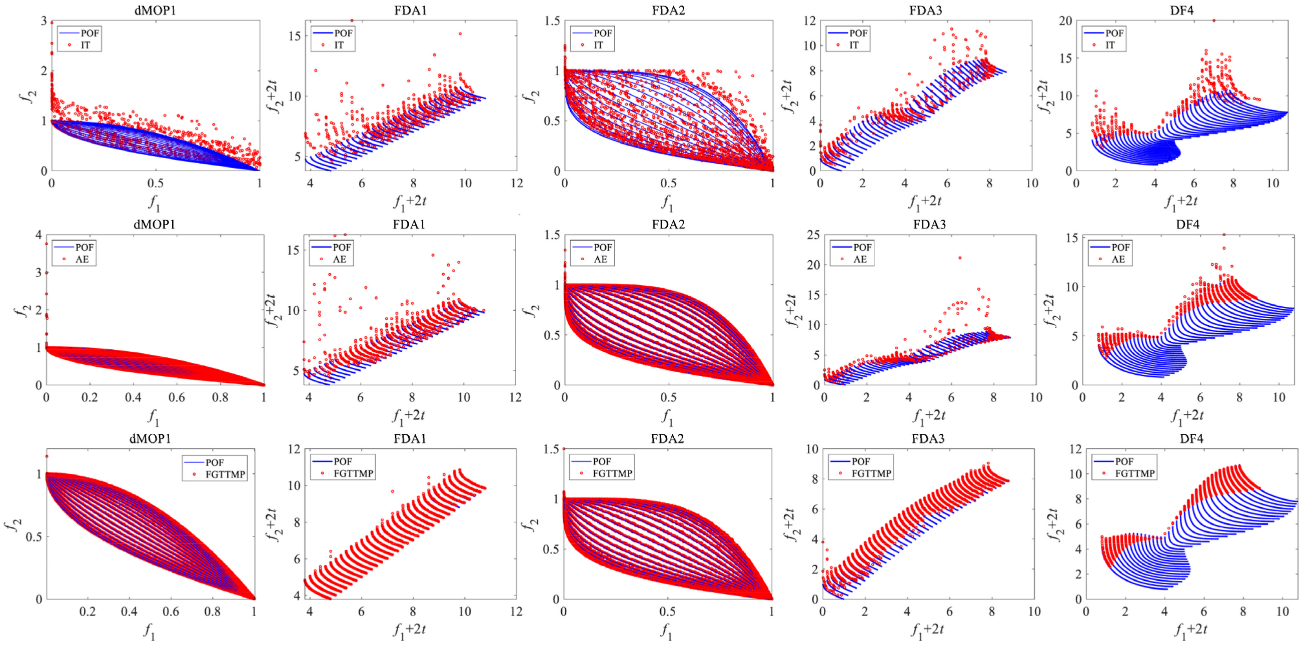
Fig. 4. IGD evolutionary curves on the F, dMOP, and FDA problems at $n_t = 10$ and $\tau_t = 5$.

FGTTMP performed significantly better, worse, and indiscriminately than competitors, respectively.

As the MIGD results displayed in Tables II and III and Tables A.I and A.II in the Supplementary Material show, the proposed FGTTMP was superior over the PPS, CKPS, MMTL, IT, GM, AE, KT, and CPDM algorithms on 68, 74, 75, 76, 60, 67, 61, and 73 cases, respectively. The statistical results confirmed that the FGTTMP could achieve better diversity and convergence performance than its competitors when dealing with dynamic environments. Furthermore, since the FGTTMP could transfer knowledge extracted from all historical individuals, it outperformed the IT and MMTL algorithms on most problems. In addition, the MHV results presented in Tables II and III and Tables A.III and A.IV in the Supplementary Material further validate the excellent performance of the proposed FGTTMP. Compared with the MIGD metric, the MHV metric can provide better quantification of convergence and diversity according to the hypervolume. In general, the proposed FGTTMP performed better than the PPS, CKPS, MMTL, IT, GM, AE, KT, and CPDM algorithms on 58, 70, 72, 76, 43, 55, 64, and 68 cases, respectively. The MHV results showed that the FGTTMP could generate a high-quality population in different environments. The excellent diversity performance of the FGTTMP contributed to finding diverse exploration for the benchmark problems.

The statistical results of the DMIGD metric are presented in Table IV, where it can be seen that the FGTTMP achieved better overall performance than the other algorithms on 14 benchmark problems with different characteristics, which was consistent with the MIGD results.

The IGD evolutionary curves for 16 different benchmarks for $n_t = 10$ and $\tau_t = 5$ are presented in Fig. 4 and Fig. A.1 in the Supplementary Material, where it can be seen that the proposed FGTTMP had the least floating IGD curve among all algorithms and achieved more stable response to environmental changes compared with the other algorithms, thus yielding better convergence performance. Moreover, the results also indicated that the IGD values of the proposed FGTTMP fluctuated over a wider range on most test problems, such as F6, F7, dMOP1, and FDA1, compared with the other algorithms. Despite this, the FGTTMP behaved more steadily than the other algorithms. To illustrate the tracking capabilities



(a) (b) (c) (d) (e)

Fig. 5. POFs obtained by the IT, AE, and FGTTMP algorithms: (a) dMOP1; (b) FDA1; (c) FDA2; (d) FDA3; and (e) DF4.

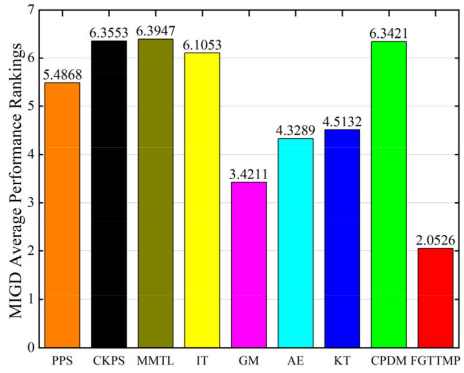


Fig. 6. MIGD Friedman rankings on all test cases.

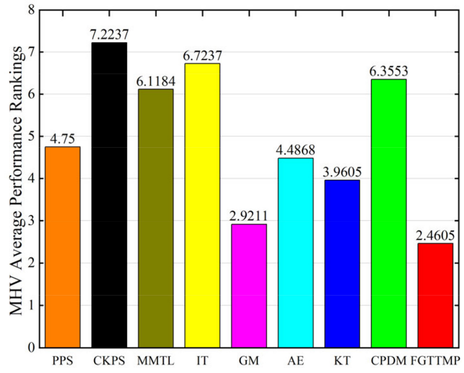


Fig. 7. MHV Friedman rankings on all test cases.

of the tested algorithms, approximate POF values of the algorithms on five benchmark problems, namely, dMOP1, FDA1, FDA2, FDA3, and DF4, for $(n_t, \tau_t) = (10, 10)$, (20, 5), (10, 10), (20, 5), and (10, 5) configurations, are plotted in Fig. 5 and Fig. A.2 in the Supplementary Material. The results in Fig. 5 and Fig. A.2 in the Supplementary Material show that the proposed FGTTMP was adept at tracking changing environments and could obtain solutions with good convergence and distributivity. However, the performance indicators might be affected due to fewer nonconverging boundary individuals in the FGTTMP.

To verify the overall performance of each algorithm regarding the MIGD and MHV metrics, the average performance rankings of the nine algorithms were obtained by the Friedman test [45], and the results are depicted in Figs. 6 and 7; the smaller ranking implied the better performance. The results showed that the proposed FGTTMP achieved the best score of 2.0526 on the MIGD metric among all algorithms, being superior to the PPS (5.4868), CKPS (6.3553), MMTL (6.3947), IT (6.1053), GM (3.4211), AE (4.3289), KT (4.5132), and CPDM (6.3421) algorithms. As expected, the FGTTMP also achieved the best score of 2.4605 on the MHV metric. Therefore, the FGTTMP had a better overall performance regarding the MIGD and MHV metrics compared with the other algorithms.

The running costs of all compared algorithms are shown in Table A.V in the Supplementary Material, and a detailed analysis can be found in the Supplementary Material.

D. Influence of Parameter K

In the FGTTMP algorithm, variable K has a key effect on the cluster determination performance of individuals generated by the information feedback model-based generators. Thus, the selection of the K value directly affects the performance of subsequent nondominated sorting and transfer. Therefore, the MIGD values of the FGTTMP on 19 benchmark problems were used to analyze the influence of the K value selection on the performance, as shown in Fig. 8 and Fig. A.3 in

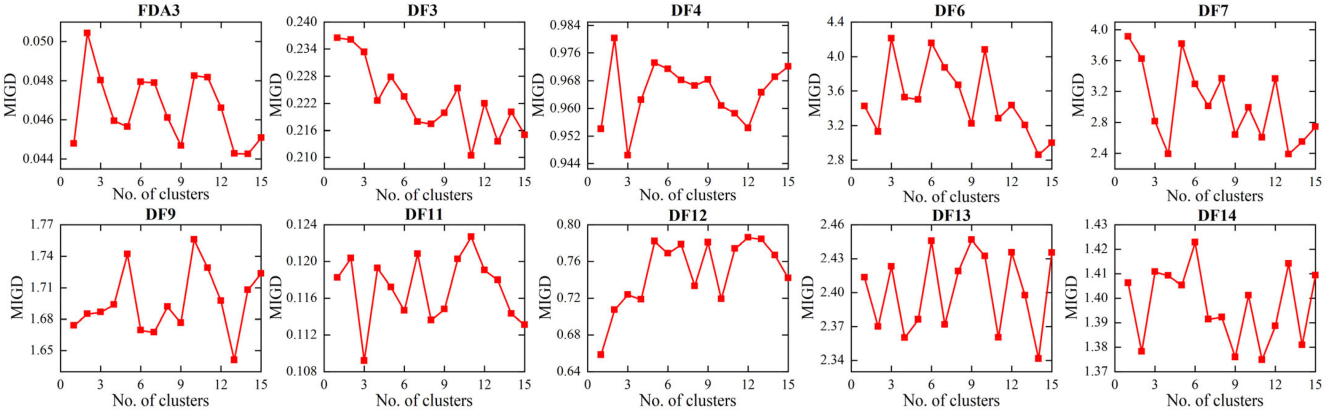


Fig. 8. MIGD values obtained by the FGTTMP for different K values.

TABLE V
COMPARISON STATISTICS OF THE FGTTMP AND ITS VARIANTS

MIGD	+/-/=	MHV	+/-/=
FGTTMP vs FGTTMP-V1	19/0/0	FGTTMP vs FGTTMP-V1	18/0/1
FGTTMP vs FGTTMP-V2	12/3/4	FGTTMP vs FGTTMP-V2	12/5/2

the Supplementary Material. The results indicated that as the number of clusters increased, the FGTTMP performed better on most problems, suggesting that more clusters could retain more high-quality individuals and thus better lead the population to converge toward the true POF. It is worth noting that for some test problems with complex DPOS or POF, such as DF13 and DF14, the fluctuations in the parameter K value had a significant impact on the MIGD values. However, the best convergence and diversity of most benchmark problems were not obtained for the same K value; therefore, a suitable K value was selected as 12.

E. Ablation Study

The main contribution of the FGTTMP is that it combines knowledge extraction with a prediction technique. Particularly, two different strategies are adopted to generate the initial population, the FGT and TMP strategies. To validate the advantageousness of the strategies, an ablation study was conducted using two variants of the FGTTMP, each of which used only one of the two mentioned strategies, the FGT or the TMP, and they were denoted by FGTTMP-V1 or FGTTMP-V2, respectively. Comparative experiments of FGTTMP and the two variants were conducted on the DF, F, dMOP, and FDA problems at $n_t = 10$ and $\tau_t = 5$. The settings of the other experimental parameters were consistent with the previous experiments. The MIGD and MHV results are presented in Tables V and Tables A.VI and A.VII in the Supplementary Material, where it can be seen that the FGTTMP performed significantly better than the FGTTMP-V1 and FGTTMP-V2 on 19 and 12 cases regarding MIGD, and on 18 and 12 cases regarding MHV, respectively. The statistics results indicated that combining the FGT and TMP strategies could provide better performance than using only one of them. A more detailed analysis is available in the Supplementary Material.

TABLE VI
COMPARISON STATISTICS ON DIFFERENT τ_t

MIGD	+/-/=	MHV	+/-/=
FGTTMP vs PPS	32/4/2	FGTTMP vs PPS	30/1/7
FGTTMP vs CKPS	31/4/3	FGTTMP vs CKPS	34/1/3
FGTTMP vs MMTL	36/1/1	FGTTMP vs MMTL	36/0/2
FGTTMP vs IT	36/2/0	FGTTMP vs IT	37/0/1
FGTTMP vs GM	30/6/2	FGTTMP vs GM	25/2/11
FGTTMP vs AE	26/8/4	FGTTMP vs AE	26/5/7
FGTTMP vs KT	27/8/3	FGTTMP vs KT	33/3/2
FGTTMP vs CPDM	36/1/1	FGTTMP vs CPDM	34/0/4

F. Different Change Frequencies

To investigate the effect of different change frequencies, additional experimental tests were conducted for n_t of 10 and τ_t of 5 and 10. The statistics results are presented in Tables VI and Tables A.VIII and A.IX in the Supplementary Material. Again, the FGTTMP performed better than its competitors on the majority of benchmark problems under different τ_t. Meanwhile, the results indicated that all algorithms were sensitive to τ_t. Despite various difficulties in the tracking capability of the algorithms posed by different τ_t values, the FGTTMP could still obtain more promising results than the other algorithms. For an intuitive comparison regarding the robustness of the algorithms, the box plots of the MIGD results on four representative benchmarks are presented in Fig. 9, where it can be seen that compared with the other algorithms, the proposed FGTTMP was much less susceptible to the effect of τ_t and performed more robustly in response to different τ_t.

V. CONCLUSION

This article presents an evolutionary algorithm based on the FGT and TMP strategies, named FGTTMP, to handle DMOPs. The proposed FGTTMP includes two main novelties. First, instead of directly exploiting the previously obtained POS to constitute an initial population, the FGTTMP accelerates convergence and maintains diversity for future evolutions by extracting valuable historical knowledge from all previous search processes using the information feedback model and cluster-based transfer learning. This strategy is not limited by memory-based algorithms and thus can provide an improved performance in various changing scenarios. Second,

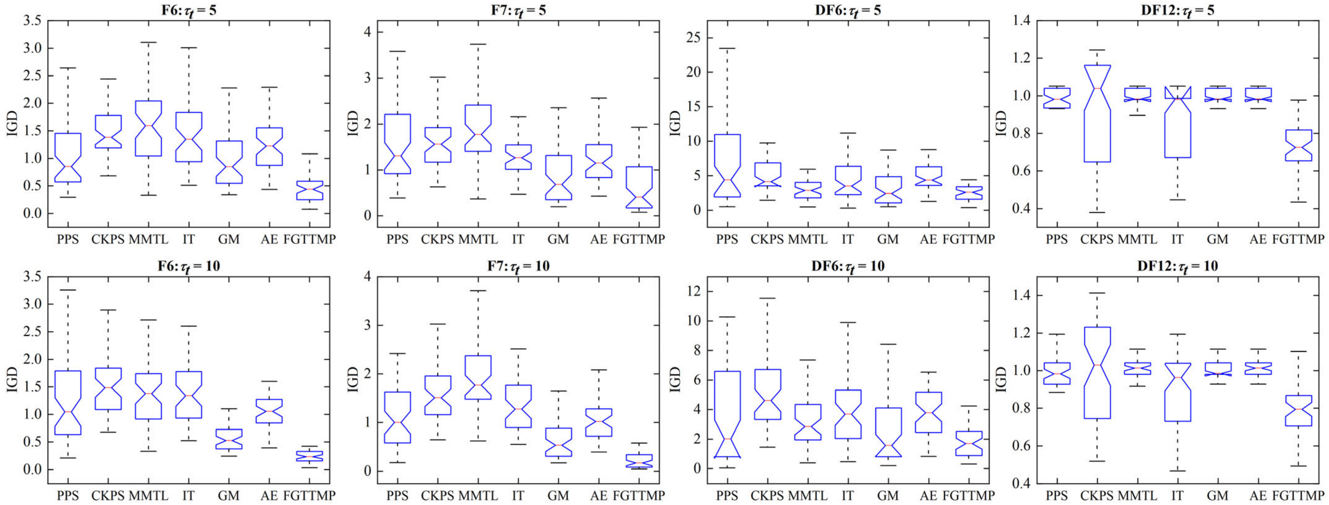


Fig. 9. Influence of τ_t on F6, F7, DF6, and DF12; n_t was set to 10, and τ_t was set to 5 and 10, in turn; each column represents the change in one benchmark problem.

a probability-based trend prediction model is introduced to track the POS manifold, which can lead to subsequent evolutions and effectively reduce the prediction deviation of the linear model.

The proposed FGTTMP is comprehensively verified by comparison experiments with eight state-of-the-art algorithms on 19 various benchmarks. The statistics results clearly illustrate the superiority of the proposed FGTTMP over the other algorithms in convergence, diversity, and robustness. This indicates that the FGTTMP can respond rapidly and efficiently to changes in an environment. In addition, the effect of each of the two strategies used in the FGTTMP is analyzed, and the analysis results suggest that the combination of the two strategies provides better results than using each of the strategies alone.

Although the FGTTMP can achieve excellent tracking capability and competitive performance on a wide range of benchmark problems, it has certain limitations. Therefore, future work could develop other knowledge extraction methods that could better learn from the historical search process. In addition, the FGTTMP could be improved to address more complex DMOPs with more objectives and larger dimensions. Finally, how to apply the FGTTMP to address practical issues, such as job scheduling and raw ore allocation, could be further investigated.

References

[1] L. Feng et al., "Solving generalized vehicle routing problem with occasional drivers via evolutionary multitasking," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3171–3184, Jun. 2021.

[2] P. Zhou, X. Wang, and T. Chai, "Multiobjective operation optimization of wastewater treatment process based on reinforcement self-learning and knowledge guidance," *IEEE Trans. Cybern.*, vol. 53, no. 11, pp. 6896–6906, Nov. 2023.

[3] M. Iqbal, B. Xue, H. Al-Sahaf, and M. Zhang, "Cross-domain reuse of extracted knowledge in genetic programming for image classification," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 569–587, Aug. 2017.

[4] L. He, R. Chiong, W. Li, S. Dhakal, Y. Cao, and Y. Zhang, "Multiobjective optimization of energy-efficient job-shop scheduling with dynamic reference point-based fuzzy relative entropy," *IEEE Trans. Ind. Informat.*, vol. 18, no. 1, pp. 600–610, Jan. 2022.

[5] N. Zeng, Z. Wang, W. Liu, H. Zhang, K. Hone, and X. Liu, "A dynamic neighborhood-based switching particle swarm optimization algorithm," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9290–9301, Sep. 2022.

[6] C. He, S. Huang, R. Cheng, K. C. Tan, and Y. Jin, "Evolutionary multiobjective optimization driven by generative adversarial networks (GANs)," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3129–3142, Jun. 2021.

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

[8] 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.

[9] 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.

[10] Q. Zhang, S. Yang, S. Jiang, R. Wang, and X. Li, "Novel prediction strategies for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 260–274, Apr. 2020.

[11] C. Wang, G. G. Yen, and M. Jiang, "A grey prediction-based evolutionary algorithm for dynamic multiobjective optimization," *Swarm Evol. Comput.*, vol. 56, Aug. 2020, Art. no. 100695.

[12] W. Song, S. Liu, X. Wang, Y. Guo, S. Yang, and Y. Jin, "Learning to guide particle search for dynamic multiobjective optimization," *IEEE Trans. Cybern.*, early access, Feb. 23, 2024, doi: 10.1109/TCYB.2024.3364375.

[13] X. Wang, Y. Zhao, L. Tang, and X. Yao, "MOEA/D with spatial-temporal topological tensor prediction for evolutionary dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, early access, Feb. 20, 2024, doi: 10.1109/TEVC.2024.3367747.

[14] G.-G. Wang and Y. Tan, "Improving metaheuristic algorithms with information feedback models," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 542–555, Feb. 2019.

[15] S. Han, K. Zhu, and M. Zhou, "Competition-driven dandelion algorithms with historical information feedback," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 2, pp. 966–979, Feb. 2022.

[16] Z. Pan, D. Lei, and L. Wang, "A bi-population evolutionary algorithm with feedback for energy-efficient fuzzy flexible job shop scheduling," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 8, pp. 5295–5307, Aug. 2022.

[17] Z.-X. Pan, L. Wang, J.-F. Chen, and Y.-T. Wu, "A novel evolutionary algorithm with adaptation mechanism for fuzzy permutation flowshop scheduling," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2021, pp. 367–374.

[18] Y. Zhang, G.-G. Wang, K. Li, W.-C. Yeh, M. Jian, and J. Dong, "Enhancing MOEA/D with information feedback models for large-scale many-objective optimization," *Inf. Sci.*, vol. 522, pp. 1–16, Jun. 2020.

[19] H. Sun, C. Wang, and Z. Hu, "A dual strategy of adaptive knee-point guidance and niche sampling for non-cyclic dynamic multiobjective optimization problems," *Eng. Appl. Artif. Intell.*, vol. 133, Jul. 2024, Art. no. 108144.

[20] Y. Chen, J. Zou, Y. Liu, S. Yang, J. Zheng, and W. Huang, "Combining a hybrid prediction strategy and a mutation strategy for dynamic multiobjective optimization," *Swarm Evol. Comput.*, vol. 70, Apr. 2022, Art. no. 101041.

[21] S. Jiang and S. Yang, "A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 65–82, Feb. 2017.

[22] K. Zhang, C. Shen, X. Liu, and G. G. Yen, "Multiobjective evolution strategy for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 974–988, Oct. 2020.

[23] Q. Chen, J. Ding, S. Yang, and T. Chai, "A novel evolutionary algorithm for dynamic constrained multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 4, pp. 792–806, Aug. 2020.

[24] Y. Hu, J. Zheng, S. Jiang, S. Yang, and J. Zou, "Handling dynamic multiobjective optimization environments via layered prediction and subspace-based diversity maintenance," *IEEE Trans. Cybern.*, vol. 53, no. 4, pp. 2572–2585, Apr. 2023.

[25] 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.

[26] Z. Liang, T. Wu, X. Ma, Z. Zhu, and S. Yang, "A dynamic multiobjective evolutionary algorithm based on decision variable classification," *IEEE Trans. Cybern.*, vol. 52, no. 3, pp. 1602–1615, Mar. 2022.

[27] Q. Zhao, B. Yan, Y. Shi, and M. Middendorf, "Evolutionary dynamic multiobjective optimization via learning from historical search process," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 6119–6130, Jul. 2022.

[28] S. Jiang, J. Zou, S. Yang, and X. Yao, "Evolutionary dynamic multiobjective optimisation: A survey," *ACM Comput. Surv.*, vol. 55, no. 4, pp. 1–47, 2022.

[29] K. Xu et al., "A cluster prediction strategy with the induced mutation for dynamic multi-objective optimization," *Inf. Sci.*, vol. 661, Mar. 2024, Art. no. 120193.

[30] J. Li, T. Sun, Q. Lin, M. Jiang, and K. C. Tan, "Reducing negative transfer learning via clustering for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 26, no. 5, pp. 1102–1116, Oct. 2022.

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

[32] 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.

[33] H. Zhang, J. Ding, M. Jiang, K. C. Tan, and T. Chai, "Inverse Gaussian process modeling for evolutionary dynamic multiobjective optimization," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 11240–11253, Oct. 2022.

[34] K. Yu et al., "A correlation-guided layered prediction approach for evolutionary dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 27, no. 5, pp. 1398–1412, Oct. 2023.

[35] R. Rambabu, P. Vadakkepat, K. C. Tan, and M. Jiang, "A mixture-of-experts prediction framework for evolutionary dynamic multiobjective optimization," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 5099–5112, Dec. 2020.

[36] M. Jiang, Z. Wang, S. Guo, X. Gao, and K. C. Tan, "Individual-based transfer learning for dynamic multiobjective optimization," *IEEE Trans. Cybern.*, vol. 51, no. 10, pp. 4968–4981, Oct. 2021.

[37] M. Jiang, Z. Wang, H. Hong, and G. G. Yen, "Knee point-based imbalanced transfer learning for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 1, pp. 117–129, Feb. 2021.

[38] C.-K. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 103–127, Feb. 2009.

[39] 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.

[40] S. Jiang, S. Yang, X. Yao, K. C. Tan, M. Kaiser, and N. Krasnogor, *Benchmark Functions for the CEC 2018 Competition on Dynamic Multiobjective Optimization*, Newcastle Univ., Newcastle upon Tyne, U.K., 2018.

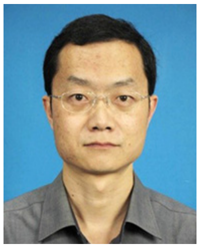
[41] J. Zou, Q. Li, S. Yang, H. Bai, and J. Zheng, "A prediction strategy based on center points and knee points for evolutionary dynamic multi-objective optimization," *Appl. Soft Comput.*, vol. 61, pp. 806–818, Dec. 2017.

[42] M. Jiang, Z. Wang, L. Qiu, S. Guo, X. Gao, and K. C. Tan, "A fast dynamic evolutionary multiobjective algorithm via manifold transfer learning," *IEEE Trans. Cybern.*, vol. 51, no. 7, pp. 3417–3428, Jul. 2021.

[43] L. Feng, W. Zhou, W. Liu, Y.-S. Ong, and K. C. Tan, "Solving dynamic multiobjective problem via autoencoding evolutionary search," *IEEE Trans. Cybern.*, vol. 52, no. 5, pp. 2649–2662, May 2022.

[44] J. Li, R. Liu, and R. Wang, "Handling dynamic multiobjective optimization problems with variable environmental change via classification prediction and dynamic mutation," *Inf. Sci.*, vol. 608, pp. 970–995, Aug. 2022.

[45] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.



Yong Wang (Member, IEEE) received the Ph.D. degree in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 2009.

He is currently a Professor with the School of Computer Science and Technology, Ocean University of China, Qingdao, China. His current research interests include software engineering and computational intelligence.

Kuichao Li received the B.S. degree in computer science and technology from Qingdao University, Qingdao, China, in 2021. He is currently pursuing the M.S. degree in computer technology with the Ocean University of China, Qingdao.

His research interests include dynamic multiobjective optimization and machine learning.

Gai-Ge Wang (Member, IEEE) received the bachelor's degree from Yili Normal University, Yining, China, in 2007, the master's degree from Northeast Normal University, Changchun, China, in 2010, and the Ph.D. degree in artificial intelligence algorithm and its applications from the University of Chinese Academy of Sciences, Beijing, China, in 2013.

He is a Professor with the Ocean University of China, Qingdao, China. His research interests are evolutionary computation and scheduling.

Dr. Wang was selected as one of the "2021 Highly Cited Researchers" by Clarivate. He was selected as one of the "2021 and 2020 Highly Cited Chinese Researchers" in computer science and technology by Elsevier.

Dunwei Gong (Senior Member, IEEE) received the Ph.D. degree in control theory and control engineering from the China University of Mining and Technology, Xuzhou, China, in 1999.

He is currently a Professor with the School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, China. His main research interests are evolutionary computation and search-based software engineering.

Keqin Li (Fellow, IEEE) received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 1985, and the Ph.D. degree in computer science from the University of Houston, Houston, TX, USA, in 1990.

He is currently a SUNY Distinguished Professor with the State University of New York at New Paltz, New Paltz, NY, USA, and a National Distinguished Professor with Hunan University, Changsha, China. He has authored or co-authored more than 950 journal articles, book chapters, and refereed conference papers.

Dr. Li received several best paper awards from international conferences, including PDPTA-1996, NAECON-1997, IPDPS-2000, ISPA-2016, NPC-2019, ISPA-2019, and CPSCom-2022. He is among the world's top five most influential scientists in parallel and distributed computing in terms of single-year and career-long impacts based on a composite indicator of the Scopus citation database. He was a 2017 recipient of the Albert Nelson Marquis Lifetime Achievement Award for being listed in Marquis Who's Who in Science and Engineering, Who's Who in America, Who's Who in the World, and Who's Who in American Education for more than 20 consecutive years. He received the IEEE TCCLD Research Impact Award from the IEEE CS Technical Committee on Cloud Computing in 2022 and the IEEE TCSVC Research Innovation Award from the IEEE CS Technical Community on Services Computing in 2023. He is a member of the SUNY Distinguished Academy. He is an AAAS Fellow and an AAIA Fellow. He is a member of Academia Europaea (Academician of the Academy of Europe).