Complex & Intelligent Systems (2023) 9:1229–1245
https://doi.org/10.1007/s40747-021-00631-3

ORIGINAL ARTICLE

A special point-based transfer component analysis for dynamic multi-objective optimization

Ruochen Liu[1] · Nanxi Li[1] · Luyao Peng[1] · Kai Wu[1]

Received: 6 July 2021 / Accepted: 26 November 2021 / Published online: 23 February 2022

Abstract

To solve dynamic multi-objective optimization problems better, the key is to adapt quickly to environmental changes and track the possible changing optimal solutions in time. In this paper, we propose a special point-based transfer component analysis for dynamic multi-objective optimization algorithm (SPTr-RM-MEDA). To be specific, when a change occurs, the neighbors of some special points are selected from the optimal set at previous time, and the transfer component analysis makes the use of minimizing the distance between the mapped previous optima and the mapped current optima. Accordingly, the purpose is to predict a part of next initial population from the neighborhoods of special points by transfer component analysis. To adapt to the change well, SPTr-RM-MEDA also reevaluates the previous optimal set. In addition, an adaptive diversity introduction strategy is adopted to maintain the population size. SPTr-RM-MEDA is performed on 12 test problems under 8 kinds of environmental changes, and experimental results show that it is superior to other five state-of-the-art algorithms on most of test problems.

Keywords Special point · Transfer component analysis · Dynamic multi-objective optimization

الأهداف متعدد الديناميكي التحسين . النقل مكون تحليل . خاصة نقطة . الرئيسيه الكلمات

Introduction

Dynamic multi-objective optimization problem (DMOP) refers to a problem containing multiple objectives under a time-varying environment. Every time the environment is changed, the optimization problems may also be changed. In the real world, DMOP exists everywhere, such as dynamic portfolio optimization problem in deregulated electricity markets, hydro-thermal power scheduling problem, training neural networks, and so on [5,11,13]. Different from the static multi-objective optimization problems, the optimal solutions of DMOPs may always change with time. Therefore, the traditional static optimization algorithms are not suitable to deal with DMOPs. Affected by some factors like the severity and frequency of environmental changes, the objective function, the constraint function, and other parameters of DMOPs will be changed [4]. A good algorithm for DMOPs not only needs a fast convergent rate before the next change happens, but also can maintain the diversity after each change [8], which is still a trade-off to be balanced for dynamic multi-objective optimization.

In recent years, many studies have made great achievements by taking evolutionary algorithms and swarm intelligence algorithms as the common frameworks in DMOPs [1,2,17,18,21]. According to the response strategy after changing, the existing algorithms can be divided into four categories: diversity-based, memory-based, multi-population-based, and prediction-based method [1].

The diversity-based method introduces the individuals with great difference into a population to preserve diversity after a change occurs. Hypermutation [16] is introduced to make the population tend to diverge by increasing the mutation rate when a change is detected. In 2007, Deb modified the static NSGA-II into a dynamic version (i.e., dynamic NSGA-II) [5]. After a change was detected, part of solutions were introduced in a random way, or they were mutated. To

✉ Ruochen Liu
ruochenliu@xidian.edu.cn

[1] Key Lab of Intelligent Perception and Image Understanding of Ministry of Education, International Center of Intelligent Perception and Computation, Xidian University, Xi'an 710071, China

adapt to the change well, the key to maintain diversity is to introduce some solutions different from ones in the populations last time in each iteration. However, the method of how many solution to be replaced is just out.

The memory-based method improves the adaptability by reusing the stored historical information when changes occur. Goh and Tan [8] used a temporary memory method in their proposed algorithm to deal with previous solutions in the archive (an external set used to store non-dominant solutions). However, this method could not make the full use of history information. Later, another algorithm developed an adaptive population management strategy [1] to select the number of random and historical solutions according to the severity of changes. This kind of methods is more suitable for the DMOPs with periodic changes.

The multi-population-based method is to track the change of multiple peaks with several populations at the same time, which is capable of increasing diversity. The self-organizing scouts' method was proposed by Branke et al. [3]. When a peak was found, a sub-population would be generated to monitor the mountain, and the rest of sub-populations would continue to search for new peaks. Li and Yang [15] proposed a multi-population particle swarm optimization algorithm, in which evolutionary programming explored the most promising region in the whole search space. Meanwhile, several sub-populations used the fast particle swarm optimization algorithm to find the local optimal solutions. The difficulty of multi-population method was that the operation of each sub-population needs to coordinate with each other. Therefore, it took more computing time than the single-population method in most situation [7].

The method based on prediction is to design a model by combining the existing historical information with some technologies from machine learning, which can predict a new initial population and adapt to changes better. Hatzakis and Wallace [9] adopted a prediction method based on the forward-looking model to predict the optimal solutions after an environmental change. Zhou et al. [24] proposed an algorithm based on population prediction strategy, which takes into account the distribution characteristic of optima in objective space, and establishes time series on center points of population. Jiang et al. [13] applied the transfer learning into DMOP. The main shortcoming of prediction-based method is that the solutions sampled from prediction models will be adopted into the next population no matter whether they will help the population evolve toward the right direction or not.

To use historical information and construct a suitable prediction model, this paper proposes a new special point-based transfer component analysis (TCA) for dynamic multi-objective optimization algorithm (SPTr-RM-MEDA). The main contributions can be given as follows:

(1) To predict the solutions closer to the potential optimal sets in target domain using TCA, it is necessary to select some special points in the source domain which are preferably approximately uniformly distributed at the Pareto fronts. Therefore, we select the five kinds of special points and their neighborhoods to build the prediction model using TCA.
(2) The non-dominated set at previous time are reevaluated using the objective functions at current time, and then, those non-dominated solution at current time will go into the next generation directly together with those predicted by TCA.
(3) Moreover, if the number of solutions gotten by the two strategies above is not enough, the adaptive diversity introduction strategy is acted as supplement by introducing remain number of individuals which are generated randomly by Gaussian mutation.
(4) This automatic adaptation mechanism is combined with RM-MEDA and experimental results show that SPTr-RM-MEDA is superior to other five typical dynamic multi-objective evolutionary algorithms.

The rest of this paper is organized as follows. Section 2 introduces the related work. Section 3 elaborates the framework of the proposed algorithm in detail. Section 4 shows the experimental settings, comparative experimental results, and analysis. Section 5 gives a conclusion and discusses the potential research work in the future.

Related work

Dynamic multi-objective optimization problem

Generally, a DMOP can be defined as follows:

$$\begin{aligned} \min_{\boldsymbol{x}\in\Omega} \quad & \boldsymbol{f}(\boldsymbol{x},t) = (f_1(\boldsymbol{x},t), f_2(\boldsymbol{x},t), \ldots, f_M(\boldsymbol{x},t))^T \\ \text{s.t.} \quad & \boldsymbol{g}(\boldsymbol{x},t) \leq 0 \\ & \boldsymbol{h}(\boldsymbol{x},t) = 0, \end{aligned} \tag{1}$$

where t is the time variable, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$ is a vector including n decision variables, Ω is the search space for decision variables, $\boldsymbol{g}(\boldsymbol{x},t)$ and $\boldsymbol{h}(\boldsymbol{x},t)$ are the equality and inequality constraints, respectively. $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{t})$ is an M-dimension objective vector. The obtained optimal solutions to Eq. (1) are called Pareto fronts (PFs) in the objective space and Pareto solutions (PSs) in the decision space.

According to the different characteristics of change, DMOPs can be divided into the following four types [6]:

- Type I: the PS is changed, the PF remains unchanged.
- Type II: both the PS and the PF are changed.

- Type III: the PF is changed, but the PS remains unchanged.
- Type IV: both the PS and PF remain unchanged.

Transfer component analysis

TCA mainly aims at the problem of domain adaptive learning in the process of transfer learning [12,19]. It reuses the knowledge obtained from the source domain and performs tasks in the target domain, which is related to but different from the source domain. Because the domains are in different distributions, the data from them will be mapped together into a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ [20], where the distance between the source and target is minimized and their internal attributes are retained to the maximum extent. Intuitively, it is not easy to minimize the distance between them in the current space, so a mapping that makes them closest in the space is necessary to establish. TCA is designed to calculate distances more universal and simpler.

TCA [12] assumes that the marginal distribution of the source domain and the target domain is different, i.e., $P(X_s) \neq P(Y_t)$, where X_s and Y_t represent the source domain and the target domain, respectively. It assumes that there is a feature mapping ϕ, so that $P(\phi(X_s)) \approx P(\phi(Y_t))$. Maximum mean discrepancy (MMD) is used to calculate the distance, and is given as follows:

$$dist\left(X_s^{'}, Y_t^{'}\right) = \left\| \frac{1}{n_1} \sum\nolimits_{i=1}^{n_1} \phi\left(x_i\right) - \frac{1}{n_2} \sum\nolimits_{i=1}^{n_2} \phi\left(y_i\right) \right\|_{\mathcal{H}}^2, \tag{2}$$

where $X_s = \{x_1, x_2, \ldots, x_{n_1}\}$, and $Y_t = \{y_1, y_2, \ldots, y_{n_2}\}$. $X_s^{'}$ and $Y_t^{'}$ represent $\phi(X_s)$ and $\phi(Y_t)$, respectively. $\phi(\cdot) : \mathcal{X} \to \mathcal{H}$ is the domain mapped in RKHS. To facilitate the calculation in a easier way, TCA introduces a kernel matrix K and a coefficient matrix L as follows:

$$\begin{aligned} K &= \begin{bmatrix} K_{s,s} & K_{s,t} \\ K_{t,s} & K_{t,t} \end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)} \\ L_{ij} &= \begin{cases} \frac{1}{n_1^2} & x_i, x_j \in X_s \\ \frac{1}{n_2^2} & y_i, y_j \in Y_t \\ -\frac{1}{n_1 n_2} & \text{otherwise}, \end{cases} \end{aligned} \tag{3}$$

where $K_{u,v}$ is a kernel matrix. Taking $K_{s,t}$ as an example, each element $k_{i,j} = k\left(x_i, y_j\right) = \phi(x_i)^T \phi(y_j)$. $K_{s,s}$, $K_{t,s}$, and $K_{t,t}$ have the similar process to be calculated.

Thus, Eq. (2) above can be changed into the following form:

$$\text{dist}\left(X_s^{'}, Y_t^{'}\right) = \text{tr}\left(KL\right), \tag{4}$$

where $tr(\cdot)$ is the trace of matrix. Here, Eq. (2) can be transformed into a semi-definite programming problem as Eq. (4).

To reduce the computation cost, let $\varphi(x) = W^T k_x \in \mathbb{R}^d$, and then the MMD can be calculated in a dimensional reduction way, where W is an $(n_1 + n_2) \times d$ $(d \ll n_1 + n_2)$ weight matrix of samples, $k_x = [k(x_1, x), \ldots, k(x_{n_1}, x), k(y_1, x), \ldots, k(y_{n_2}, x)]^T$, and the kernel function K can be converted as the following formula:

$$\begin{aligned} \widetilde{K} &= \left[\varphi(x_1), \ldots, \varphi(x_{n_1}), \varphi(y_1), \ldots, \varphi(y_{n_2})\right]^T \\ &\quad \cdot \left[\varphi(x_1), \ldots, \varphi(x_{n_1}), \varphi(y_1), \ldots, \varphi(y_{n_2})\right] \\ &= \left[W^T k_{x_1}, \ldots, W^T k_{x_{n_1}}, W^T k_{y_1}, \ldots, W^T k_{y_{n_2}}\right]^T \\ &\quad \cdot \left[W^T k_{x_1}, \ldots, W^T k_{x_{n_1}}, W^T k_{y_1}, \ldots, W^T k_{y_{n_2}}\right] \\ &= K^T W W^T K \\ &= K W W^T K, \end{aligned} \tag{5}$$

where K is a symmetric matrix and, therefore, $K^T = K$.

After sorting out the above formulas, TCA can be expressed as

$$\begin{aligned} &\underset{W}{\arg\min}\ tr\left(W^T K L K W\right) + \mu\ tr\left(W^T W\right) \\ &\qquad\qquad \text{s.t.} \quad W^T K H K W = I, \end{aligned} \tag{6}$$

where $H = I_{n_1+n_2} - \frac{1}{n_1+n_2}\mathbf{1}\mathbf{1}^T$ is a central matrix to maintain the different data characteristics, I is the identity matrix, and $\mathbf{1}$ is an all-ones matrix. For solving the above optimization problem, the second regularization term of Eq. (6) is added by the Lagrange duality to get W. It can be concluded that the solution of W is the first m eigenvectors.

In conclusion, TCA can be concluded as follows. The inputs are two matrices of source and target domain, respectively, which are two populations generated randomly from the previous and current time. We first calculate L and H matrices, select a common kernel function for mapping such as linear kernel and Gaussian kernel, and then calculate K. Finally, the output is W, which is constructed by the first m eigenvectors of $(KLK + \mu I)^{-1} KHK$.

Regularity model-based multi-objective estimation of distribution algorithm

When the environmental change is not detected, the static algorithm is utilized for seeking the optimal solutions at current time. In 2008, the regularity model-based multi-objective estimation of distribution algorithm (RM-MEDA) was proposed by Zhang [22], which is as the static optimization algorithm after each change in this paper.

RM-MEDA makes the advantage of the regularity of PS. In other words, the distribution of PS in the feasible region is an $(M - 1)$-dimension manifold, where M is the number of objectives. To be more exact, the shape of parent population from PS is mapped into $(M - 1)$-dimension linear space,

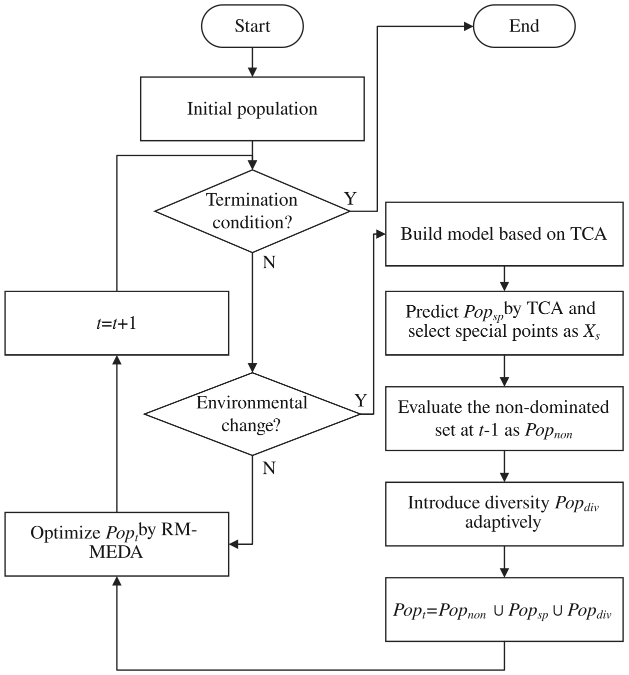

Start
End
Initial population
Termination condition?
Y
N
Build model based on TCA
t=t+1
Predict Pop_{sp} by TCA and select special points as X_s
Environmental change?
Y
N
Evaluate the non-dominated set at t-1 as Pop_{non}
Introduce diversity Pop_{div} adaptively
Optimize Pop_t by RM-MEDA
$Pop_t=Pop_{non} \cup Pop_{sp} \cup Pop_{div}$

Fig. 1 The diagram of the overall algorithm

and the offspring is produced by sampling on multiple linear spaces.

RM-MEDA algorithm divides the parent population into K classes, and each class forms a linear subspace. The process is as follows: first, K linear Spaces are generated by dividing population into K sub-populations randomly. And then calculate the distance between each point in parent population and centers from different linear subspace, and cluster sub-populations again by the minimum distance. After dividing points and calculating the new centers, principal component analysis is used to update the new linear subspace. The process is repeated until each class is no longer changed. As for sampling, when linear fitting above is over, the upper and lower bounds of each linear subspace need to be demarcated. Then, the probability of the new solution in the truncated linear subspace is based on the ratio of the volume of each truncated linear space to the total volume. A new solution is obtained by selecting a manifold with this probability, sampling on the manifold uniformly, and adding noise. The offspring population can be generated by repeating the sampling process.

The proposed algorithm

Overall flowchart of SPTr-RM-MEDA

This section introduces the proposed SPTr-RM-MEDA. Figure 1 shows the diagram of the proposed algorithm. When a change is detected, a mechanism of automatic adaptation to change would generate the initial population under new environment, using TCA with special points and an adaptive diversity introduction. And then, the initial population is optimized by RM-MEDA [22]. Compared with NSGA-II, RM-MEDA showed more advantages with TCA in experiments [13]. The pseudo-code of SPTr-RM-MEDA is shown in Algorithm 1, and the $N_{Pop_{sp}}$ and $N_{Pop_{non}}$ are the size of Pop_{sp} and Pop_{non}, respectively.

Algorithm 1 SPTr-RM-MEDA

```
Input:
  A DMOP; the termination condition: Ter; the population size: $N$;a kernel function.
Output:
  Approximation of PS: $X$; Approximation of PF: $F(X)$.
  Initialization: The number of time change and the initial parent population $Pop_0$.
  Step1: Randomly select 5% individuals to reevaluated. if environment is changed, go to Step 2. Otherwise, go to Step 6.
  Step2: Predict a new population $Pop_{sp}$ by the prediction strategy in Algorithm 2.
  Step3: Reevaluate the previous non-dominated solutions $Pop_{non}$ in $t+1$ time.
  Step4: Calculate the number of diversity introduction $N_{Pop_{div}} = N - N_{Pop_{non}} - N_{Pop_{sp}}$, and introduce random individuals $Pop_{div}$.
  Step5: Get $Pop_{t+1} = Pop_{sp} + Pop_{non} + Pop_{div}$ by Steps 2, 3 and 4. $t = t + 1$.
  Step6: Optimize $Pop_t$ by RM-MEDA.
  Step7: If Ter is met, stop. Otherwise go to the step 1.
```

Automatic adaptation to change

Prediction strategy

Special points can outline the PF for most multi-objective problems. Five kinds of special points are introduced in this paper as follows.

The boundary point [23] refers to the point of minimum value of each object in the objective space for a minimization problem. Its mathematical expression is given as follows:

$$B = \{(f_{1min}(X), \ldots, f_M(X)), \ldots, (f_1(X), \ldots, f_{Mmin}(X))\}, \quad (7)$$

where M is the number of objectives and B represents the set of boundary points. To be more specific, there are two or three boundary points in two-objective or three-objective optimization, respectively.

The knee point [23] has the maximum marginal regression rate in the objective space. In the other word, it can be defined as the point farthest from the vertical distance of hyperplane S determined by boundary points. The dimension of hyperplane varies which is equal to the dimension of objective space minus one in this paper. For example, if the number of objectives M is 2 or 3, the dimension of S is 1 or 2, respectively.

For $\forall x \in S$, the equation $W^T X + b = 0$ is satisfied. The coefficient vector $W = [w_1, w_2, \ldots, w_M]^T$ and the coefficient term b is a constant. In the objective space, the variable vector $X = [x_1, x_2, \ldots, x_M]^T$. The knee point can be calculated by maximizing the vertical distance from the point X to the hyperplane S

$$K = \underset{X \in P}{\arg\max} \quad D(X_0, S) \tag{8}$$

$$D(X, S) = \frac{|W^T X + b|}{\|W\|}, \tag{9}$$

where X represents those points out of hyperplane S, and P stands for the non-dominated set at previous time t. If the point X is on the hyperplane S, the vertical distance is 0.

The point closest to the boundary (PCTB) can be understood as the point with the smallest distance to the sum of all boundary points. The formula is presented in the following:

$$\min_{1 \le i \le N_P} \sum_{j=1}^{M} \left\| X_i - B_j \right\|, \tag{10}$$

where B_j is the boundary point, and N_p is the number of non-dominated points X_i.

For the minimization problem, the ideal point is composed of the minimum values in each dimension of the objective space. However, in general, the ideal point does not exist in the non-dominated set. $E = [E_1, E_2, \ldots, E_M]^T$ is the ideal point, which is given in the following:

$$E_j = \min_{X \in P} f_j(X), \; j = 1, \ldots, M, \tag{11}$$

where P represents the non-dominated set at previous time t.

The center point usually refers to the center of all points in the non-dominated set. The calculation formula of the center point is shown as follows:

$$C_i = \frac{1}{|P|} \sum_{X \in P} X_i, \quad i = 1, 2 \ldots, M, \tag{12}$$

where P is the non-dominated set.

These special points and their neighbor points will be found from the PS and PF at previous time to form special point populations, which will participate in predicting points by TCA model in this paper. Because the center point or the ideal point in Eqs. (11) and (12) does not exist in the non-dominated set, the closest point from the non-dominated set P to either of them in the objective space is chosen as the actual center or ideal point, respectively. All the neighbor points are selected based on the Euclidean distance to the actual special points in the objective space. Note that the size of each neighborhood is 10. And only the neighborhoods of special points are used for predicting by TCA in the following strategy.

When the environmental change is detected, the prediction part is mainly based on the special points in the non-dominated set at previous moment. Each special point and its neighborhood will be partly sampled, and then, the sampling set will be acquired through TCA as a part of initial population at next time. In TCA prediction method, the objective values of individuals at t time are in the source domain, and that at $t + 1$ time is in the target domain. When establishing the mapping relationship, the individuals in new environment can be predicted by the transfer learning of the sampled special points' population. The prediction strategy based on TCA with special points is given in detail in Algorithm 2. At Step 1, the points on the neighborhood refer to closet to the special point and from the non-dominated set. In Steps 6 and 7, l represents the individuals from special points' neighborhoods $SPop$.

Algorithm 2 Prediction strategy based on TCA

```
Input:
  The objective function: $F(\cdot)$; PF at time $t$; a kernel function: $k(\cdot, \cdot)$;
  the sampling rate: $\gamma_s$; the size of population: $N$.
Output:
  $Pop_{sp}$ at time $t + 1$.
  Step1: Calculate the special points, and the sampling size around each special point's neighborhood is $\gamma_s N$. Then partial samplings are generated and denoted as $SPop$.
  Step2: Randomly generate two sets of initial solutions $X_s$ and $Y_t$, and then calculate objective function values $F_t(X_s)$ and $F_{t+1}(Y_t)$ at time $t$ and $t + 1$, respectively.
  Step3: Use TCA on $F_t(X_s)$ and $F_{t+1}(Y_t)$ to determine the mapping relationship and obtain the matrix $W$ by Eq. (6).
  Step4: Map the population $SPop$ at time $t$ into high-dimension space by $k(\cdot, \cdot)$.
  Step5: Use $\varphi(x) = W^T k_x$ to calculate the characteristic mapping of $Pop_{sp}$ in high-dimension space, $\varphi(F_t(l)) = W^T k_l, \; l \in SPop$.
  Step6: $\varphi(F_t(l))$ will search one by one for mapping solutions at $t + 1$, and $\varphi(F_{t+1}(x))$ is the closest to it. $x \leftarrow \arg\min_x \|\varphi(F_{t+1}(x)) - \varphi(F_t(l))\|$.
  Step7: The solution obtained in Step 6 is the predicted initial population $Pop_{sp}$.
```

Adaptive diversity introduction strategy

Because the number of boundary points is related to the objective dimension, the number of special points in two or three objectives is 6 or 7, respectively, so the size of total partial sampling (denoted as $N_{Pop_{sp}}$) is $6 \times \gamma_s \times N$ or $7 \times \gamma_s \times N$, respectively. And, the non-dominated set at t time are reevaluated using the objective functions at $t + 1$

time, and then, those non-dominated solutions (the size is denoted as $N_{Pop_{non}}$) will go into the next generation together with those predicted by TCA. If the total number of Pop_{sp} and Pop_{non} ($N_{Pop_{sp}}$ + $N_{Pop_{non}}$) is less than the population size N, the remain $N - (N_{Pop_{sp}} + N_{Pop_{non}})$ individuals are generated randomly around Pop_{non} and Pop_{sp} by adding the noise with normal distribution with the mean of 0 and the standard deviation of 0.5.

Experimental results

In this section, the proposed SPTr-RM-MEDA is compared with five other dynamic multi-objective optimization evolutionary algorithms, including transfer learning-based NSGA-II (Tr-NSGA-II) [13], transfer learning-based RM-MEDA (Tr-RM-MEDA) [13], two version of dynamic NSGA-II [5] (DNSGA-II-A and DNSGA-II-B), and individual transfer-based dynamic multi-objective evolutionary algorithm (IT-DMOEA) [14]. Tr-NSGA-II and Tr-RM-MEDA both used TCA to predict based on different static algorithms, NSGA-II and RM-MEDA. DNSGA-II-A and DNSGA-II-B are modified NSGA-II with different individuals introducing strategies. In DNSGA-II-A, randomly generated individuals are introduced and replaced part of population whenever there was a change. In DNSGA-II-B, some individuals are replaced by those generated by mutations. IT-DMOEA used a presearch strategy to filter out the good individuals and proposed an individual-based transfer learning to construct the initial population.

Experimental settings

To test the performance of SPTr-RM-MEDA, three types of DMOPs [10] were selected, as shown in Table 1. What's more, the time parameter in Eq.(1) is $t = \lfloor \tau/\tau_t \rfloor / n_t$, where n_t, τ_T and τ_t are the severity of change, the maximum number of iterations, and frequency of change, respectively, and τ is from $\tau_T/20$ to τ_T, there are totally 20 environment changes. The parameters of SPTr-RM-MEDA, SPTr-NSGA-II, Tr-RM-MEDA, and Tr-NSGA-II are presented in the following: the population size N is 200; the sample rate γ_s is 0.05. In TCA, the kernel function is a Gaussian kernel function, the expected dimension value m is 20, and the trade-off parameter μ is set at 0.5. Our algorithm was performed on 12 kinds of test functions under 8 groups of environmental changes, which is denoted as C1–C8, and different environmental parameter settings are shown in Table 2.

Performance metrics

MIGD [24] is used to calculate the average inverted generational distance (IGD) value in different time, and it is adopted to evaluate the performance in this paper. The definition is shown as follows:

Table 1 Functions settings

Function	Type	Dim	Objectives	Feature
FDA4	I	12	3	Non-convex
FDA5	II	12	3	Non-convex
FDA5$_{iso}$	II	12	3	Isolated
FDA5$_{dec}$	II	12	3	Deceptive
DIMP2	I	10	2	Continuous
DMOP2	II	10	2	Convex to concave
DMOP2$_{iso}$	II	10	2	Isolated
DMOP2$_{dec}$	II	10	2	Deceptive
DMOP3	I	10	2	Convex
HE2	III	30	2	Discontinuous
HE7	III	10	2	Complex PSs
HE9	III	10	2	Complex PSs

Table 2 Functions parameters' settings

	C1	C2	C3	C4	C5	C6	C7	C8
n_t	10	10	10	10	1	1	20	20
τ_T	100	200	500	1000	200	1000	200	1000
τ_t	5	10	25	50	10	50	10	50

$$\mathrm{MIGD}(P_t, PF_t, C) = \frac{1}{|T|} \sum_{t \in T} \frac{\sum_{v \in PF_t} d(v, P_t)}{|PF_t|}, \qquad (13)$$

where T means a set of discrete time points in a run. PF_t is Pareto optimal points set distributed uniformly in PF, P_t represents the approximated PF obtained by algorithms, $d\,(v, P_t) = \min_{u \in P_t} \|F(v) - F(u)\|$ represents the distance between the point v from PF_t and P_t. The parameter C is a symbol for eight environments, ranging from $C1$ to $C8$ shown in Table 2.

To evaluate algorithms under different environments, DMIGD is defined based on the MIGD, calculated by [13]

$$\mathrm{DMIGD}(P_t, PF_t) = \frac{1}{|E|} \sum_{C \in E} \mathrm{MIGD}(P_t, PF_t, C), \qquad (14)$$

where E means a set of different environment, and $|E|$ is the number of different environment in experiments. In our experiments, $|E|$ is equal to 8.

Table 3 The MIGD and DMIGD of 12 test functions compared with four algorithms

	Tr-RM-MEDA	DNSGA-II-A	DNSGA-II-B	IT-DMOEA	SPTr-RM-MEDA
FDA4					
C1	**8.59E−02**	6.47E−01(++)	6.50E−01(++)	3.57E−01(++)	9.54E−02
C2	**5.97E−02**	2.57E−01(++)	2.70E−01(++)	2.31E−01(++)	6.36E−02
C3	**3.41E−02**	3.86E−02(++)	3.95E−02(++)	9.38E−02(++)	3.83E−02
C4	2.86E−02	1.80E−02	**1.79E−02**	6.73E−02(++)	3.10E−02
C5	**6.00E−02**	3.47E+00(++)	3.38E+00(++)	6.10E−01(++)	2.28E−01
C6	**2.29E−02**	3.21E+00(++)	3.35E+00(++)	6.89E−02(++)	4.35E−02
C7	**5.63E−02**	7.47E−02(++)	7.42E−02(++)	2.25E−01(++)	6.79E−02
C8	2.92E−02	**1.41E−02**	1.44E−02	6.75E−02(++)	3.01E−02
DMIGD	**4.71E−02**	9.66E−01(++)	9.74E−01(++)	2.15E−01(++)	7.47E−02
FDA5					
C1	9.41E−01	1.15E+00	1.11E+00	1.44E+00	**7.20E−01**
C2	8.06E−01	7.89E−01	7.97E−01	1.20E+00	**6.54E−01**
C3	6.06E−01(++)	5.31E−01	**5.32E−01**	7.51E−01(++)	6.01E−01
C4	6.31E−01	5.45E−01	5.45E−01	**4.26E−01**	6.31E−01
C5	**9.86E−01**	2.55E+00(++)	2.55E+00(++)	1.26E+00(++)	1.21E+00
C6	**4.89E−01**	3.19E+00(++)	3.04E+00(++)	1.36E+00(++)	5.14E−01
C7	7.85E−01	6.95E−01	6.91E−01	1.21E+00	**6.21E−01**
C8	6.08E−01(++)	5.21E−01	5.21E−01	**3.77E−01**	6.07E−01
DMIGD	7.32E−01	1.25E+00	1.22E+00	1.00E+00	**6.95E−01**
FDA5$_{iso}$					
C1	9.35E−01	7.86E−01	7.89E−01	2.68E+01	**6.05E−01**
C2	8.27E−01(++)	8.02E−01(++)	8.05E−01(++)	**2.66E−01**	6.24E−01
C3	6.49E−01	8.11E−01	8.11E−01	2.80E+00	**5.92E−01**
C4	6.25E−01	8.05E−01	8.09E−01	3.23E+01	**6.25E−01**
C5	6.80E−01	**6.19E−01**	6.18E−01	7.17E+00	7.23E−01
C6	4.72E−01	7.25E−01	6.64E−01	2.70E+01	**4.71E−01**
C7	8.04E−01(++)	7.84E−01(++)	7.86E−01(++)	**2.66E−01**	6.02E−01
C8	6.02E−01	7.74E−01(++)	7.69E−01(++)	**1.13E−01**	6.02E−01
DMIGD	6.99E−01	7.63E−01	7.56E−01	1.21E+01	**6.06E−01**
FDA5$_{dec}$					
C1	1.88E+00(++)	9.50E−01	**9.44E−01**	5.11E+03(++)	1.30E+00
C2	1.63E+00(++)	**8.56E−01**	8.62E−01	2.37E+03(++)	1.15E+00
C3	1.33E+00(++)	**7.78E−01**	7.80E−01	2.58E+03(++)	8.21E−01
C4	9.95E−01	7.76E−01	7.76E−01	2.61E+03	**7.61E−01**
C5	9.65E+00(++)	1.47E+00	**1.29E+00**	1.31E+02(++)	1.77E+00
C6	**5.52E−01**	9.19E−01(++)	8.83E−01(++)	4.76E+03(++)	6.09E−01
C7	1.67E+00(++)	**8.12E−01**	8.22E−01	3.51E+03(++)	1.04E+00
C8	1.27E+00(++)	7.53E−01	**7.52E−01**	4.67E+03(++)	8.30E−01
DMIGD	2.37E+00(++)	9.14E−01	**8.89E−01**	3.22E+03(++)	1.04E+00
DIMP2					
C1	1.40E+01(++)	1.08E+01	1.08E+01	**1.05E+01**	1.13E+01
C2	1.18E+01(++)	6.14E+00	**6.10E+00**	8.83E+00	8.99E+00
C3	6.17E+00(++)	**2.15E+00**	2.21E+00	6.33E+00(++)	5.51E+00
C4	4.38E+00	**4.91E−01**	5.12E−01	5.20E+00(++)	4.42E+00
C5	1.14E+01(++)	**5.83E+00**	5.97E+00	8.62E+00	9.01E+00

Table 3 continued

	Tr-RM-MEDA	DNSGA-II-A	DNSGA-II-B	IT-DMOEA	SPTr-RM-MEDA
C6	4.54E+00	3.96E−01	**3.93E−01**	5.07E+00(++)	4.69E+00
C7	1.21E+01(++)	**5.81E+00**	5.95E+00	8.69E+00(++)	9.25E+00
C8	4.52E+00	**8.06E−02**	9.18E−02	5.22E+00(++)	4.66E+00
DMIGD	8.61E+00(++)	**3.96E+00**	4.00E+00	7.31E+00(++)	7.23E+00
DMOP2					
C1	3.57E+00(++)	3.09E+00(++)	2.80E+00(++)	**1.47E−01**	4.58E−01
C2	9.32E−01(++)	1.76E−01(++)	1.81E−01(++)	**7.46E−02**	1.17E−01
C3	5.67E−02	9.23E−02	8.84E−02	1.38E−02	**8.53E−03**
C4	2.42E−03	5.67E−01	3.37E−01	9.76E−03	**1.82E−03**
C5	2.96E+01(++)	3.74E+01(++)	3.73E+01(++)	**2.21E−01**	2.38E+01
C6	1.97E+01(++)	3.74E+01(++)	3.74E+01(++)	**1.37E−01**	1.91E+01
C7	9.46E−01(++)	4.76E−02	**3.80E−02**	7.55E−02	8.18E−02
C8	6.69E−03	8.94E−02	2.32E−03	8.48E−03	**1.84E−03**
DMIGD	6.85E+00(++)	9.86E+00(++)	9.77E+00(++)	**8.59E−02**	5.44E+00
DMOP2iso					
C1	**3.75E−04**	8.69E−02(++)	7.65E−02(++)	8.28E−03(++)	3.79E−04
C2	3.74E−04	2.30E−03	6.62E−03	8.27E−03	**3.74E−04**
C3	3.82E−04	3.83E−04	3.82E−04	8.27E−03	**3.77E−04**
C4	3.75E−04	3.78E−04	3.80E−04	8.27E−03	**3.75E−04**
C5	**1.27E−01**	1.31E−01(++)	1.38E−01(++)	1.40E−01(++)	1.28E−01
C6	1.27E−01	1.29E−01	1.29E−01	1.40E−01	**1.27E−01**
C7	3.77E−04	1.15E−03	7.85E−03	4.66E−03	**3.77E−04**
C8	3.78E−04	3.79E−04	3.79E−04	4.66E−03	**3.75E−04**
DMIGD	**3.21E−02**	4.40E−02	4.50E−02	4.03E−02	**3.21E−02**
DMOP2dec					
C1	2.90E+00	6.96E+00	7.36E+00	2.01E+00	**9.80E−01**
C2	1.70E+00	1.47E+01	1.69E+01	1.26E+00	**4.44E−01**
C3	2.65E−01	7.84E+02	4.77E+02	3.73E−01	**1.25E−01**
C4	5.79E−02	3.51E+04	2.94E+04	1.08E−01	**5.76E−02**
C5	6.40E+00	5.33E+05	6.21E+05	3.34E+00	**2.64E+00**
C6	2.50E−01(++)	9.39E+06(++)	9.15E+06(++)	**1.81E−01**	2.30E−01
C7	1.80E+00	2.85E+01	2.31E+01	1.10E+00	**3.14E−01**
C8	7.18E−02	2.57E+03	2.38E+03	8.02E−02	**3.99E−02**
DMIGD	1.68E+00	1.25E+06	1.23E+06	1.06E+00	**6.03E−01**
DMOP3					
C1	1.29E+00(++)	6.36E+00(++)	5.94E+00(++)	**7.70E−02**	3.39E−01
C2	3.55E−01(++)	1.71E+00(++)	1.80E+00(++)	**3.57E−02**	6.73E−02
C3	3.65E−02	1.35E−01	1.78E−01	5.69E−03	**4.32E−03**
C4	1.79E−03	1.34E−02	1.37E−02	2.78E−03	**1.39E−03**
C5	2.00E+01	3.43E+01(++)	3.48E+01(++)	**2.06E−01**	2.33E+01
C6	1.32E+01	2.55E+01(++)	2.50E+01(++)	**1.10E−01**	1.91E+01
C7	3.30E−01(++)	4.69E−01(++)	5.04E−01(++)	**3.41E−02**	5.59E−02
C8	2.29E−03	6.20E−03	5.11E−03	2.42E−03	**1.53E−03**
DMIGD	4.40E+00	8.55E+00(++)	8.53E+00(++)	**5.92E−02**	5.36E+00

Table 3 continued

	Tr-RM-MEDA	DNSGA-II-A	DNSGA-II-B	IT-DMOEA	SPTr-RM-MEDA
HE2					
C1	1.24E+00(++)	8.36E−02	**8.23E−02**	1.48E+00(++)	5.18E−01
C2	8.19E−01(++)	4.11E−02	**3.91E−02**	1.37E+00(++)	4.05E−01
C3	3.60E−01(++)	**3.13E−02**	3.14E−02	1.09E+00(++)	1.46E−01
C4	9.41E−02(++)	3.14E−02	**3.12E−02**	8.68E−01(++)	4.41E−02
C5	7.20E−01(++)	**4.44E−02**	4.50E−02	1.32E+00(++)	2.28E−01
C6	6.25E−02(++)	**3.64E−02**	3.64E−02	6.89E−01(++)	4.99E−02
C7	8.21E−01(++)	**3.90E−02**	3.99E−02	1.36E+00(++)	3.21E−01
C8	1.01E−01	3.15E−02	**3.15E−02**	8.66E−01(++)	5.18E−01
DMIGD	9.56E−01	9.88E−01	9.85E−01	1.01E+00	**8.43E−01**
HE7					
C1	1.53E+00	4.62E+00(++)	4.77E+00(++)	**1.78E−01**	2.42E+00
C2	1.70E+00	5.20E+00(++)	5.08E+00(++)	**1.29E−01**	3.59E+00
C3	1.88E+00	5.25E+00(++)	5.61E+00(++)	**8.59E−02**	4.82E+00
C4	2.58E+00	6.02E+00(++)	5.76E+00(++)	**8.30E−02**	5.36E+00
C5	1.32E+00	3.18E+00(++)	3.08E+00(++)	**1.68E−01**	2.23E+00
C6	1.88E+00	3.84E+00(++)	3.66E+00(++)	**1.36E−01**	3.26E+00
C7	1.68E+00	5.52E+00(++)	5.22E+00(++)	**1.32E−01**	3.75E+00
C8	2.70E+00	5.46E+00(++)	5.85E+00(++)	**8.65E−02**	5.29E+00
DMIGD	1.80E+00	4.45E+00(++)	4.45E+00(++)	**2.23E−01**	3.51E+00
HE9					
C1	2.55E+00	1.36E+03(++)	1.37E+03(++)	**3.20E−01**	2.53E+00
C2	2.16E+00	1.37E+03(++)	1.36E+03(++)	**2.99E−01**	3.36E+00
C3	2.18E+00	1.37E+03(++)	1.36E+03(++)	**2.75E−01**	5.98E+00
C4	2.56E+00	1.36E+03(++)	1.35E+03(++)	**2.60E−01**	6.54E+00
C5	2.05E+00	1.36E+03(++)	1.36E+03(++)	**2.63E−01**	2.66E+00
C6	2.34E+00	1.35E+03(++)	1.36E+03(++)	**2.24E−01**	4.55E+00
C7	2.04E+00	1.36E+03(++)	1.36E+03(++)	**3.04E−01**	3.54E+00
C8	2.44E+00	1.36E+03(++)	1.35E+03(++)	**2.63E−01**	6.28E+00
DMIGD	2.29E+00	1.36E+03(++)	1.36E+03(++)	**2.76E−01**	4.43E+00

Results and analysis

Comparison with other algorithms

To prove the universal applicability of the proposed SPTr-RM-MEDA, we compared it with Tr-RM-MEDA [13], DNSGA-II-A [5], DNSGA-II-B [5], and IT-DMOEA [14]. The population size and the maximum of iterations are the same with SPTr-RM-MEDA to make a fair comparison. The test functions are consistent with Table 1. Here, we choose the average MIGD of each problem under 8 different environments and DMIGD of each problem as the performance metrics. Note that "++" represents that SPTr-RM-MEDA performs better than this algorithm except for the best one in boldface. The experimental result for comparison is shown in Table 3.

As can be seen, the experimental results on 9 out of 12 groups of problems show that SPTr-RM-MEDA has better performance than DNSGA-II-A and DNSGA-II-B. More than half of the results of SPTr-RM-MEDA have the higher MIGD than the transfer learning algorithms like Tr-RM-MEDA and IT-DMOEA. And for type I and type II problems, SPTr-RM-MEDA is superior to DNSGA-II-A and DNSGA-II-B for most problems, especially for a series of DMOP. For type III problems (i.e., the complex HE7 and HE9), although our algorithm is better than DNSGA-II-A and DNSGA-II-B, it has poor convergence compared with Tr-RM-MEDA and IT-DMOEA. However, from the DMIGD value, SPTr-RM-MEDA takes up the first with 5 out of 12 best, and IT-DMOEA has 4 out of 12.

Table 4 The MIGD of 12 test functions compared with Tr-RM-MEDA and Tr-NSGA-II

		Tr-RM-MEDA	Tr-NSGA-II	SPTr-NSGA-II	SPTr-RM-MEDA
FDA4	C1	8.59E−02	**6.25E−02**	1.02E−01	9.54E−02
	C2	5.97E−02	**5.04E−02**	8.89E−02	6.36E−02
	C3	**3.41E−02**	6.51E−02	9.13E−02	3.83E−02
	C4	**2.86E−02**	5.19E−02	6.34E−02	3.10E−02
	C5	**6.00E−02**	6.82E−02	6.88E−02	2.28E−01
	C6	**2.29E−02**	3.66E−02	5.07E−02	4.35E−02
	C7	5.63E−02	**4.63E−02**	8.36E−02	6.79E−02
	C8	**2.92E−02**	5.33E−02	7.21E−02	3.01E−02
FDA5	C1	9.41E−01	1.10E+00	1.02E+00	**7.20E−01**
	C2	8.06E−01	9.57E−01	7.86E−01	**6.54E−01**
	C3	6.06E−01	1.16E+00	6.54E−01	**6.01E−01**
	C4	6.31E−01	9.77E−01	7.33E−01	**6.31E−01**
	C5	9.86E−01	1.16E+00	**6.22E−01**	1.21E+00
	C6	**4.89E−01**	1.13E+00	6.42E−01	5.14E−01
	C7	7.85E−01	9.56E−01	7.05E−01	**6.21E−01**
	C8	6.08E−01	8.56E−01	7.04E−01	**6.07E−01**
FDA5$_{iso}$	C1	9.35E−01	9.77E−01	7.77E−01	**6.05E−01**
	C2	8.27E−01	8.76E−01	7.59E−01	**6.24E−01**
	C3	6.49E−01	6.56E−01	7.32E−01	**5.92E−01**
	C4	6.25E−01	6.26E−01	7.19E−01	**6.25E−01**
	C5	6.80E−01	**5.79E−01**	5.99E−01	7.23E−01
	C6	4.72E−01	**4.53E−01**	5.88E−01	4.71E−01
	C7	8.04E−01	7.70E−01	7.30E−01	**6.02E−01**
	C8	6.02E−01	**5.76E−01**	7.19E−01	6.02E−01
FDA5$_{dec}$	C1	1.95E+00	6.28E+00	**1.28E+00**	1.30E+00
	C2	1.62E+00	4.89E+00	**8.04E−01**	1.15E+00
	C3	1.32E+00	2.88E+00	**6.47E−01**	8.21E−01
	C4	1.01E+00	9.76E−01	**6.28E−01**	7.61E−01
	C5	5.56E+00	8.07E+01	**1.11E+00**	1.77E+00
	C6	**5.40E−01**	5.03E+00	**7.87E−01**	6.09E−01
	C7	1.67E+00	1.88E+00	**7.40E−01**	1.04E+00
	C8	1.11E+00	9.10E−01	**6.98E−01**	8.30E−01
DIMP2	C1	1.40E+01	1.65E+01	1.64E+01	**1.13E+01**
	C2	1.18E+01	1.17E+01	1.08E+01	**8.99E+00**
	C3	6.17E+00	6.66E+00	6.22E+00	**5.51E+00**
	C4	4.38E+00	4.58E+00	**3.89E+00**	4.42E+00
	C5	1.14E+01	1.17E+01	**6.41E+00**	9.01E+00
	C6	4.54E+00	5.06E+00	**2.21E+00**	4.69E+00
	C7	1.21E+01	1.18E+01	**6.90E+00**	9.25E+00
	C8	4.52E+00	4.55E+00	**3.03E+00**	4.66E+00
DMOP2	C1	3.57E+00	1.26E+00	1.15E+00	**4.58E−01**
	C2	9.32E−01	9.52E−01	6.69E−01	**1.17E−01**
	C3	5.67E−02	2.27E−01	5.82E−02	**8.53E−03**
	C4	2.42E−03	3.21E−02	4.23E−02	**1.82E−03**
	C5	2.96E+01	3.33E+01	**1.01E+01**	2.38E+01
	C6	1.97E+01	2.01E+01	**1.19E−01**	1.91E+01
	C7	9.46E−01	9.76E−01	4.05E−01	**8.18E−02**
	C8	6.69E−03	2.85E−02	4.03E−02	**1.84E−03**

Table 4 continued

		Tr-RM-MEDA	Tr-NSGA-II	SPTr-NSGA-II	SPTr-RM-MEDA
DMOP2$_{iso}$	C1	**3.75E−04**	4.85E−04	4.94E−02	3.79E−04
	C2	3.74E−04	4.74E−04	4.87E−02	**3.74E−04**
	C3	3.82E−04	5.01E−04	4.87E−02	**3.77E−04**
	C4	3.75E−04	5.05E−04	4.86E−02	**3.75E−04**
	C5	**1.27E−01**	1.29E−01	2.36E−01	1.28E−01
	C6	1.27E−01	1.29E−01	2.37E−01	**1.27E−01**
	C7	3.77E−04	4.72E−04	4.63E−02	**3.77E−04**
	C8	3.78E−04	4.96E−04	4.66E−02	**3.75E−04**
DMOP2$_{dec}$	C1	2.90E+00	1.53E+00	1.68E+00	**9.80E−01**
	C2	1.70E+00	1.12E+00	7.84E−01	**4.44E−01**
	C3	2.65E−01	5.05E−01	1.55E−01	**1.25E−01**
	C4	5.79E−02	1.49E−01	6.18E−02	**5.76E−02**
	C5	6.40E+00	6.49E+00	4.52E+00	**2.64E+00**
	C6	2.50E−01	1.04E+00	8.81E−01	**2.30E−01**
	C7	1.80E+00	1.05E+00	6.07E−01	**3.14E−01**
	C8	7.18E−02	1.45E−01	7.23E−02	**3.99E−02**
DMOP3	C1	1.29E+00	1.08E+00	4.34E−01	**3.39E−01**
	C2	3.55E−01	7.49E−01	2.74E−01	**6.73E−02**
	C3	3.65E−02	1.97E−01	4.16E−02	**4.32E−03**
	C4	1.79E−03	4.16E−02	4.88E−03	**1.39E−03**
	C5	2.00E+01	3.15E+01	**8.98E+00**	2.33E+01
	C6	1.32E+01	1.96E+01	**1.29E−01**	1.91E+01
	C7	3.30E−01	6.46E−01	1.04E−01	**5.59E−02**
	C8	2.29E−03	2.91E−02	5.67E−03	**1.53E−03**
HE2	C1	1.24E+00	1.08E+00	7.48E−01	**5.18E−01**
	C2	8.19E−01	2.06E−01	**9.71E−02**	4.05E−01
	C3	3.60E−01	8.54E−02	**6.31E−02**	1.46E−01
	C4	9.41E−02	7.81E−02	6.19E−02	**4.41E−02**
	C5	7.20E−01	**5.95E−02**	1.74E−01	2.28E−01
	C6	6.25E−02	**4.73E−02**	1.47E−01	4.99E−02
	C7	8.21E−01	1.83E−01	**6.97E−02**	3.21E−01
	C8	1.01E−01	8.24E−02	**6.39E−02**	5.18E−01
HE7	C1	1.53E+00	**1.52E+00**	8.00E+01	2.42E+00
	C2	**1.70E+00**	9.62E+00	1.06E+02	3.59E+00
	C3	**1.88E+00**	6.31E+01	1.37E+02	4.82E+00
	C4	**2.58E+00**	7.20E+01	1.59E+02	5.36E+00
	C5	**1.32E+00**	4.12E+00	1.77E+02	2.23E+00
	C6	**1.88E+00**	2.34E+01	1.71E+02	3.26E+00
	C7	**1.68E+00**	4.41E+00	9.48E+01	3.75E+00
	C8	**2.70E+00**	1.01E+02	1.01E+02	5.29E+00
HE9	C1	2.55E+00	3.34E+00	6.95E+00	**2.53E+00**
	C2	**2.16E+00**	3.84E+00	7.51E+00	3.36E+00
	C3	**2.18E+00**	5.33E+00	7.62E+00	5.98E+00
	C4	**2.56E+00**	6.20E+00	7.54E+00	6.54E+00
	C5	**2.05E+00**	3.32E+00	8.19E+00	2.66E+00
	C6	**2.34E+00**	4.24E+00	8.28E+00	4.55E+00
	C7	**2.04E+00**	4.01E+00	6.60E+00	3.54E+00
	C8	**2.44E+00**	6.00E+00	6.53E+00	6.28E+00

Table 5 The CPU running time of each function

	Tr-RM-MEDA	Tr-NSGA-II	SPTr-NSGA-II	SPTr-RM-MEDA
DMOP3	1.13E+04	2.15E+04	7.13E+03	**1.32E+03**
DIMP2	8.73E+03	3.80E+02	**1.88E+02**	1.21E+03
FDA4	1.70E+04	1.13E+04	**3.69E+03**	6.09E+03
DMOP2	8.46E+03	1.82E+04	5.60E+03	**3.87E+03**
DMOP2$_{iso}$	3.39E+03	3.80E+03	**1.25E+03**	6.58E+03
DMOP2$_{dec}$	5.97E+03	2.26E+04	6.47E+03	**1.73E+03**
FDA5	1.25E+04	1.71E+04	6.53E+03	**4.37E+03**
FDA5$_{iso}$	4.97E+03	4.99E+03	**1.89E+03**	6.74E+03
FDA5$_{dec}$	5.38E+03	2.06E+04	5.94E+03	**4.38E+03**
HE2	1.40E+04	4.72E+04	9.55E+03	**9.09E+03**
HE7	2.24E+04	3.04E+04	**5.03E+03**	7.96E+03
HE9	1.44E+04	1.90E+04	**3.47E+03**	3.88E+03

Analysis of key technologies in SPTr-RM-MEDA

To figure out how the other static algorithm and the proposed automatic adaptation to change work, we compare Tr-RM-MEDA and Tr-NSGA-II [13] with SPTr-RM-MEDA and SPTr-NSGA-II in experiments, including MIGD metric in Table 4 and the CPU running time in Table 5. SPTr-NSGA-II with NSGA-II has the only difference with SPTr-RM-MEDA.

In Table 4, on the comparison between two static algorithms, including RM-MEDA and NSGA-II. From Tr-RM-MEDA and Tr-NSGA-II, or SPTr-RM-MEDA and SPTr-NSGA-II, there is not doubt that RM-MEDA is a good optimizer on the contribution of convergence and diversity than NSGA-II.

Otherwise, when Tr-NSGA-II is compared with SPTr-NSGA-II and Tr-RM-MEDA is compared with SPTr-RM-MEDA, it can still be seen that the automatic adaptation to change we proposed improves the efficiency of TCA model in Table 4. SPTr-RM-MEDA performs well on almost half of problems under different environments. This is because the PFs of these problems are changed significantly, and SPTr-RM-MEDA can accurately find the special points, improving the convergence rate of the algorithm.

Furthermore, in terms of experimental cost, CPU running time metric is adopted to show the efficiency of automatic adaptation to change. Here, we give the average CPU time under the 8 changes for each test function. The results are shown in Table 5. SPTr-RM-MEDA and SPTr-NSGA-II have less time complexity and faster convergent speed, which greatly saves the calculation cost. According to SPTr-RM-MEDA and SPTr-NSGA-II, almost all functions run half as fast. Especially for DMOP2 and FDA5, the running time is shortened by more than 70%.

To confirm the efficiency of automatic adaptation to change by another way, Tr-RM-MEDA is used to compare with SPTr-RM-MEDA in Fig. 2. Take the environmental setting $C3$ as an example, the two-objective PFs under 5 different environment changes are shown. The parameter T is the environment change varying from 1 to 20. The green dots are the true PFs, and the red dots are the PFs from the chosen algorithm. Altogether, the SPTr-RM-MEDA converges better than the Tr-RM-MEDA. However, there are still some problems with complex or static PFs like DIMP2, HE7, and HE9, the PFs of which are far away from the true PFs.

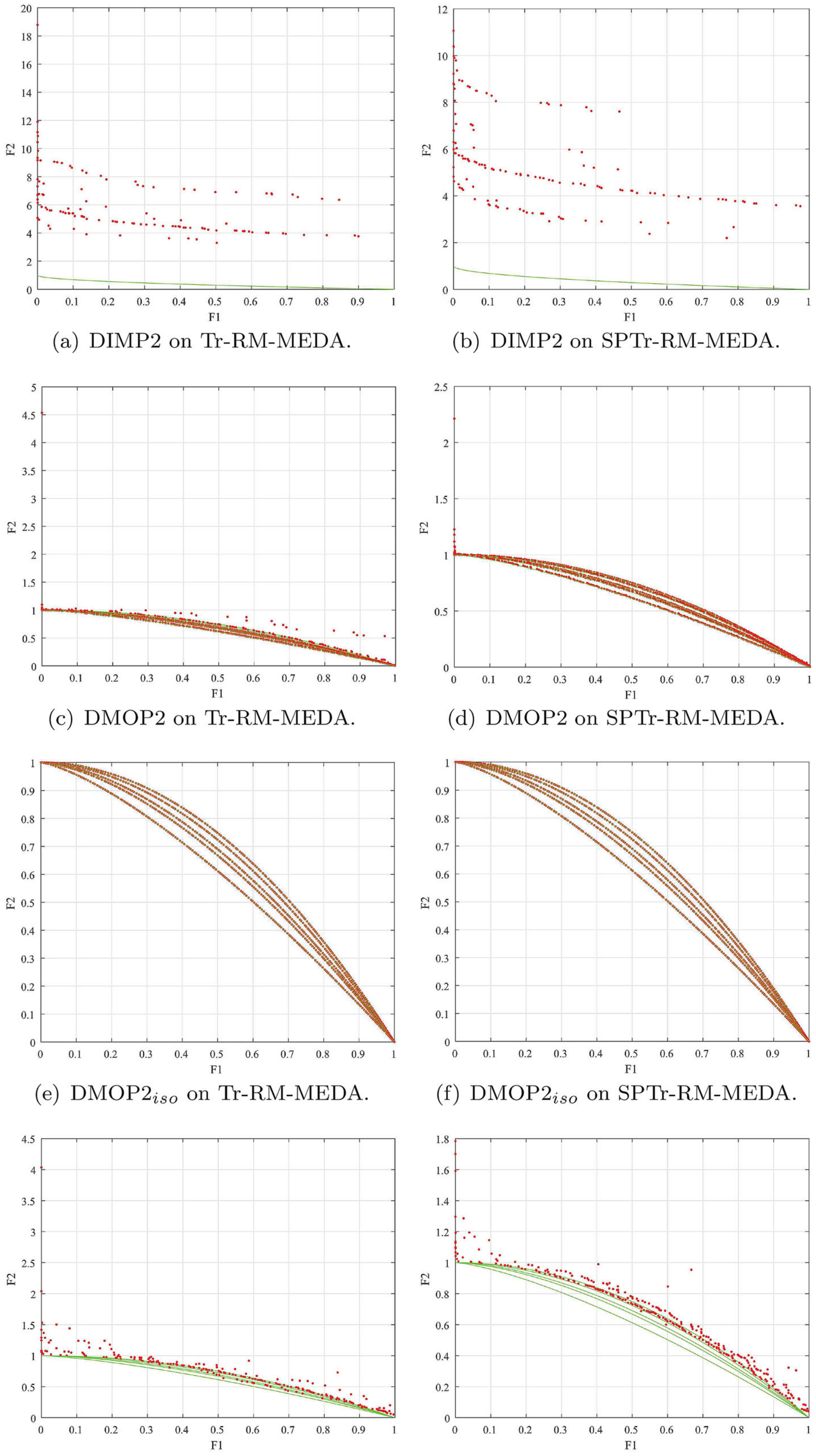
Furthermore, to confirm the convergence of three-objective PFs, SPTr-RM-MEDA is compared with Tr-RM-MEDA under 4 environment changes from $C3$ in Fig. 3. The blue dots are the true PFs, and the red star-like dots are the PFs from algorithm. Apart from the problem FDA4, the PFs from SPTr-RM-MEDA are more close to the true PFs than Tr-RM-MEDA. Because the true PFs are unchanged on FDA4, the difference between two algorithms is unclear.

Analysis of the size of neighborhood for each special point

The size of neighborhood of special points is an important coefficient in SPTr-RM-MEDA. In this section, the size ranges from 5 to 15 on DMOP2$_{iso}$ problem under 8 different environments, so as to confirm the effects of different size. The experimental result shows the MIGD of them in Table 6.

From Table 6, it is obvious that the size of neighborhood has no effect on the MIGD value under different environments. And the larger size of neighborhood is, the more time need for computing. Hence, we chose the middle scale of the size like 10.

Fig. 2 Comparative PFs on different environments (T = 5, 10, 15, 18, and 20 for two-objective problems under C3



(a) DIMP2 on Tr-RM-MEDA.

(b) DIMP2 on SPTr-RM-MEDA.

(c) DMOP2 on Tr-RM-MEDA.

(d) DMOP2 on SPTr-RM-MEDA.

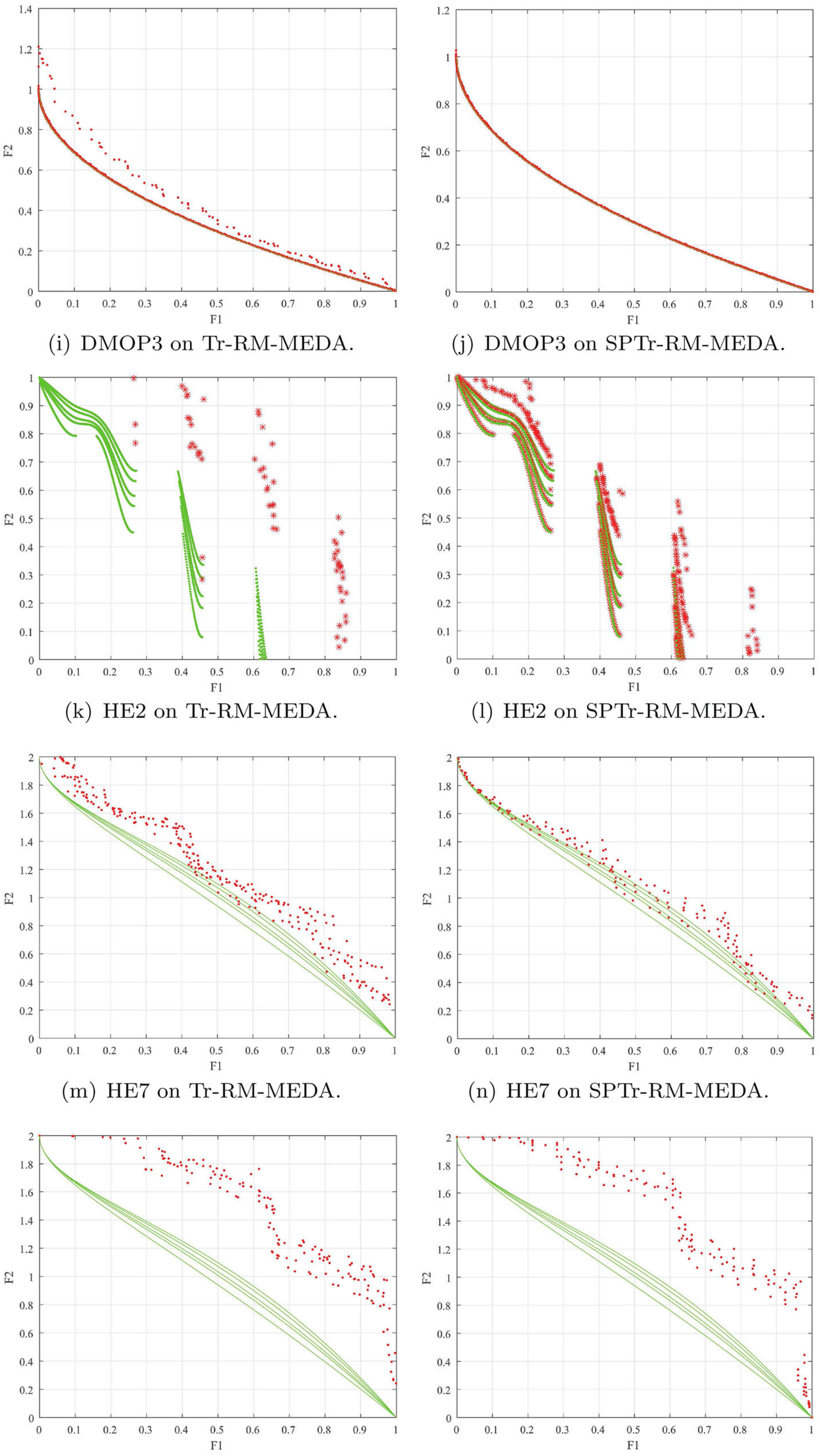
(e) DMOP2$_{iso}$ on Tr-RM-MEDA.

(f) DMOP2$_{iso}$ on SPTr-RM-MEDA.

(g) DMOP2$_{dec}$ on Tr-RM-MEDA.

(h) DMOP2$_{dec}$ on SPTr-RM-MEDA.

Fig. 2 continued



(i) DMOP3 on Tr-RM-MEDA.

(j) DMOP3 on SPTr-RM-MEDA.

(k) HE2 on Tr-RM-MEDA.

(l) HE2 on SPTr-RM-MEDA.

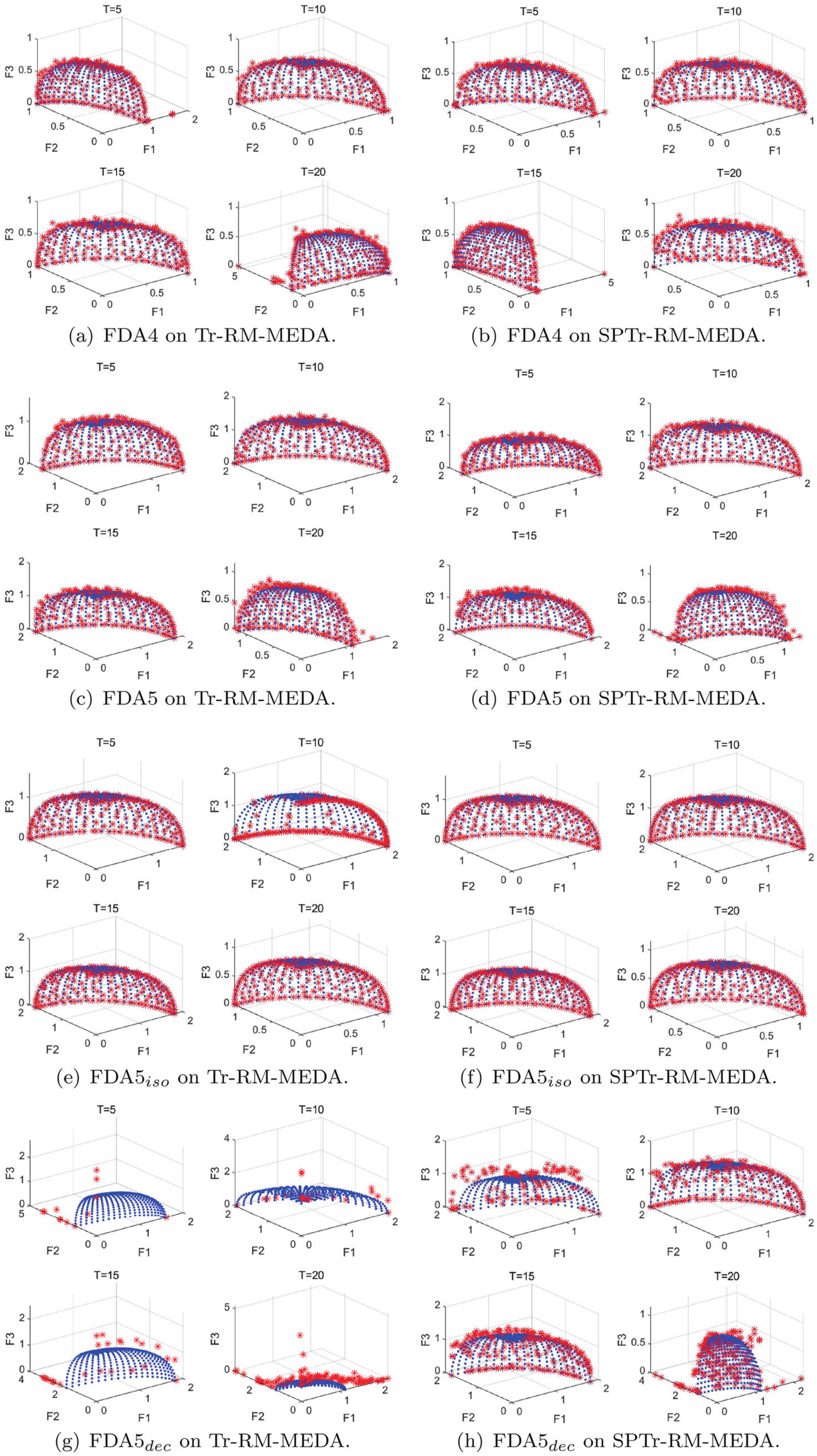
(m) HE7 on Tr-RM-MEDA.

(n) HE7 on SPTr-RM-MEDA.

(o) HE9 on Tr-RM-MEDA.

(p) HE9 on SPTr-RM-MEDA.

Fig. 3 Comparative PFs on different environments (T = 5, 10, 15 and 20) for three-objective problems under C3



(a) FDA4 on Tr-RM-MEDA.

(b) FDA4 on SPTr-RM-MEDA.

(c) FDA5 on Tr-RM-MEDA.

(d) FDA5 on SPTr-RM-MEDA.

(e) FDA5$_{iso}$ on Tr-RM-MEDA.

(f) FDA5$_{iso}$ on SPTr-RM-MEDA.

(g) FDA5$_{dec}$ on Tr-RM-MEDA.

(h) FDA5$_{dec}$ on SPTr-RM-MEDA.

Table 6 The MIGD of different size of neighborhood

		5	10	15
DMOP2$_{iso}$	C1	3.77E−04	3.79E−04	3.78E−04
	C2	3.78E−04	3.74E−04	3.79E−04
	C3	3.76E−04	3.77E−04	3.76E−04
	C4	3.74E−04	3.75E−04	3.77E−04
	C5	1.28E−01	1.28E−01	1.28E−01
	C6	1.27E−01	1.27E−01	1.27E−01
	C7	3.76E−04	3.77E−04	3.78E−04
	C8	3.78E−04	3.75E−04	3.74E−04

Conclusion

In this paper, SPTr-RM-MEDA based on special points and TCA prediction strategy was proposed for DMOPs. The main improvements focused on what the strategy could be undertook after each change happened, which is called as automatic adaptation to change. To improve the convergence, after partly sampling on special points and their neighborhoods, these points are used for transfer learning prediction. Meanwhile, the diversity strategy is introduced adaptively to balance the convergence and diversity. These strategies are prepared for the initial population, which acts as the input of static algorithm when the environment is changed.

The statistical results show that the proposed SPTr-RM-MEDA is very competitive in DMOPs compared with the other five representative algorithms, especially for type I and II problems. And, comparative experiments have shown that RM-MEDA and a mechanism of automatic adaptation to change contributed greatly to the proposed algorithm on different parts. The convergent speed of the population is accelerated with the automatic adaptation to change working from the perspective of CPU running time.

However, the presented work still needs improvements in the future. In terms of the selection of transfer learning methods and the diversity of introduction strategies, we can conduct a further research on the type III of DMOPs in our future work. That is still a question how to react when PS is not changed. And every time, the change happens, generating initial points by a more various and appropriate way needs to be researched.

Acknowledgements This work was supported by the Provincial Natural Science Foundation of Shaanxi of China (No. 2019JZ-26), and the National Natural Science Foundation of China (Nos. 61876141, 61373111).

References

1. Azzouz R, Bechikh S, Said LB (2017) A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy. Soft Comput 21(4):885–906. https://doi.org/10.1007/s00500-015-1820-4
2. Branke J (2012) Evolutionary optimization in dynamic environments, genetic algorithms and evolutionary computation, vol 3. Springer Science & Business Media. https://www.books.google.com.hk/books?id=XOLiBwAAQBAJ. Accessed 28 Nov 2020
3. Branke J, Kaußler T, Smidt C, Schmeck H (2000) A multi-population approach to dynamic optimization problems. In: Parmee IC. (eds) Evolutionary design and manufacture. Springer, pp 299–307
4. Chen Z, Bai X, Yang Q, Huang X, Li G (2015) Strategy of constraint, dominance and screening solutions with same sequence in decision space for interval multi-objective optimization. Acta Autom Sin 41(12):2115–2124. https://doi.org/10.16383/j.aas.2017.c150811
5. Deb K, Karthik S et al (2007) Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling. In: International conference on evolutionary multi-criterion optimization. Springer, pp 803–817. https://doi.org/10.1007/978-3-540-70928-2_60,
6. Farina M, Deb K, Amato P (2004) Dynamic multiobjective optimization problems: test cases, approximations, and applications. IEEE Trans Evol Comput 8(5):425–442. https://doi.org/10.1109/TEVC.2004.831456
7. Fukunaga K (2013) Introduction to statistical pattern recognition. Elsevier, Amsterdam
8. Goh CK, Tan KC (2008) A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. IEEE Trans Evol Comput 13(1):103–127. https://doi.org/10.1109/TEVC.2008.920671
9. Hatzakis I, Wallace D (2006) Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach. In: Proceedings of the 8th annual conference on Genetic and evolutionary computation, pp 1201–1208. https://doi.org/10.1145/1143997.1144187
10. Helbig M, Engelbrecht A (2015) Benchmark functions for cec 2015 special session and competition on dynamic multi-objective optimization. Department of Computer Science, University of Pretoria, Pretoria, South Africa, Report
11. Isaacs A, Puttige V, Ray T, Smith W, Anavatti S (2008) Development of a memetic algorithm for dynamic multi-objective optimization and its applications for online neural network modeling of uavs. In: 2008 IEEE international joint conference on neural networks (IEEE World congress on computational intelligence), pp 548–554. https://doi.org/10.1109/IJCNN.2008.4633847
12. Jiang M, Huang W, Huang Z, Yen GG (2017) Integration of global and local metrics for domain adaptation learning via dimensionality reduction. IEEE Trans Cybern 47(1):38–51. https://doi.org/10.1109/TCYB.2015.2502483
13. Jiang M, Huang Z, Qiu L, Huang W, Yen GG (2017) Transfer learning-based dynamic multiobjective optimization algorithms.

IEEE Trans Evol Comput 22(4):501–514. https://doi.org/10.1109/TEVC.2017.2771451

14. Jiang M, Wang Z, Guo S, Gao X, Tan KC (2020) Individual-based transfer learning for dynamic multiobjective optimization. IEEE Trans Cybern 51(10):4968–4981
15. Li C, Yang S (2008) Fast multi-swarm optimization for dynamic optimization problems. In: 2008 fourth international conference on natural computation, vol 7, pp 624–628. https://doi.org/10.1109/ICNC.2008.313
16. Li K, Fialho A, Kwong S, Zhang Q (2014) Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 18(1):114–130
17. Mavrovouniotis M, Li C, Yang S (2017) A survey of swarm intelligence for dynamic optimization: algorithms and applications. Swarm Evol Comput 33:1–17. https://doi.org/10.1016/j.swevo.2016.12.005
18. Nguyen TT, Yang S, Branke J (2012) Evolutionary dynamic optimization: a survey of the state of the art. Swarm Evol Comput 6:1–24. https://doi.org/10.1016/j.swevo.2012.05.001
19. Patel VM, Gopalan R, Li R, Chellappa R (2015) Visual domain adaptation: a survey of recent advances. IEEE Signal Process Mag 32(3):53–69. https://doi.org/10.1109/MSP.2014.2347059
20. Schölkopf B, Platt J, Hofmann T (2007) A kernel method for the two-sample-problem, pp 513–520. https://www.ieeexplore.ieee.org/document/6287330. Accessed 28 Nov 2020
21. Yazdani D, Cheng R, Yazdani D, Branke J, Jin Y, Yao X (2021) A survey of evolutionary continuous dynamic optimization over two decades–part a. IEEE Trans Evol Comput. https://doi.org/10.1109/TEVC.2021.3060014
22. Zhang Q, Zhou A, Jin Y (2008) Rm-meda: a regularity model-based multiobjective estimation of distribution algorithm. IEEE Trans Evol Comput 12(1):41–63. https://doi.org/10.1109/TEVC.2007.894202
23. Zhang X, Tian Y, Jin Y (2014) A knee point-driven evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 19(6):761–776. https://doi.org/10.1109/TEVC.2014.2378512
24. Zhou A, Jin Y, Zhang Q (2013) A population prediction strategy for evolutionary dynamic multiobjective optimization. IEEE Trans Cybern 44(1):40–53. https://doi.org/10.1109/TCYB.2013.2245892