Dynamic Multi-Objective Optimization Algorithm Based Decomposition and Preference

Yaru Hu[a,b], Jinhua Zheng[b,c,*], Juan Zou[b,*], Shouyong Jiang[e,*], Shengxiang Yang[d,*]

[a] *the Department of Mathematics and Computational Science, Xiangtan University, Xiangtan 411105, China*
[b] *Key Laboratory of Intelligent Computing and Information Processing (Ministry of Education), Xiangtan University, Xiangtan, Hunan, 411105, China*
[c] *Hunan Provincial Key Laboratory of Intelligent Information Processing and Application, Hengyang Normal University, Hengyang, 421002, China*
[d] *School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K.*
[e] *School of Computer Science, University of Lincoln, Lincoln LN6 7TS, U.K.*

Abstract

Most of the existing dynamic multi-objective evolutionary algorithms (DMOEAs) are effective, which focuses on searching for the approximation of Pareto-optimal front (POF) with well-distributed in handling dynamic multi-objective optimization problems (DMOPs). Nevertheless, in real-world scenarios, the decision maker (DM) may be only interested in a portion of the corresponding POF (i.e., the region of interest) for different instances, rather than the whole POF. Consequently, a novel DMOEA based decomposition and preference (DACP) is proposed, which incorporates the preference of DM into the dynamic search process and tracks a subset of Pareto-optimal set (POS) approximation with respect to the region of interest (ROI). Due to the presence of dynamics, the ROI, which is defined in which DM gives both the preference point and the neighborhood size, may be changing with time-varying DMOPs. Consequently, our algorithm moves the well-distributed reference points, which are located in the neighborhood range, to around the preference point to lead the evolution

* Corresponding author
Email addresses: huyaru1199@gmail.com (Yaru Hu), jhzheng@xtu.edu.cn (Jinhua Zheng), zoujuan@xtu.edu.cn (Juan Zou), math4neu@gmail.com (Shouyong Jiang), syang@dmu.ac.uk (Shengxiang Yang)

of the whole population. When a change occurs, a novel strategy is performed for responding to the current change. Particularly, the population will be reinitialized according to a promising direction obtained by letting a few solutions evolve independently for a short time. Comprehensive experiments show that this approach is very competitive in comparison with state-of-the-art methods.
Keywords:
Dynamic multi-objective evolutionary algorithms (DMOEAs), the region of interest (ROI), reference points, changing preference point

1. Introduction

The multi-objective optimization problems [1, 2] with time-varying and conflicting objectives are very common in real-world applications [3, 4, 5, 6, 7], such as dynamic responses for a rail freight wagon [8] and vehicle routing optimization [9]. Consequently, the dynamic multi-objective optimization (DMO) has attracted increasing attention in the evolutionary computation community. This kind of optimization problem, whose benchmark functions and related parameters may change over time, is called dynamic multi-objective optimization problems (DMOPs). During the process of optimization, a DMOP may include two or more Pareto-optimal fronts (POFs) and/or Pareto-optimal sets (POSs). Hence, DMOPs can be divided into four types [10, 11] according to the dynamic characteristics of POF and POS: **a)** POS changes over time and POF is fixed; **b)** both POS and POF change over time; **c)** POS is fixed, and POF changes over time; **d)** both POS and POF are fixed, but the problem changes over time. DMOPs pose tremendous challenges [2, 12] to multi-objective optimization evolutionary algorithms (MOEAs) that are effective for static multi-objective optimization problems (MOPs) [13, 14]. This is because the optimization of MOPs needs to find a POS approximation within sufficient time. But, the optimization of DMOPs needs to approximate a series of POFs and/or POSs during the whole evolutionary process as DMOPs are dynamic in nature [15]. To effectively solve DMOPs, a number of dynamic MOEAs (DMOEAs) have emerged in recent years. They are combining some excellent change response strategies with the existing MOEAs, which is very widespread, especially in DMO [1, 16].

Unlike MOEAs [17, 18], the optimization period utilizing DMOEAs finding the approximation of each POS or POF during a DMOP is limited rather than sufficient [15, 19]. Generally, an effective DMOEA, which is by change response strategies assisting an MOEA to respond to the dynamic environmental change, can search for the whole POF approximation with well-distribution in each limited time interval [20, 21]. Nevertheless, the ultimate goal of DMOEAs is to provide solutions that satisfy the preferences of DM in solving DMOPs. In real-world scenarios, most DMs do not always want to find all possible solutions (i.e., the whole POF approximation) on DMOPs, but is only interested in a portion of POF (i.e., the region of interest, called ROI) that is defined by DM expressed preferences [22, 23]. Consequently, searching for solutions that satisfy the DM's preferences is a kind of key technology unsolved for DMOPs in DMO.

Besides, the expressed preferences of DM may change over time due to the dynamism of DMOPs, which is very common in real-world scenarios. **For example**, a cycle of software development usually contains four phases [12, 20], including **1)** inception, **2)** elaboration, **3)** construction, and **4)** transition, which is different from each other. However, the requirements of the customer are unavoidably happened with time-varying according to the demand of the market, leading one or more phase(s) of four phases to need to be redesigned in the next life cycle of the software development. In other words, DM may express their different preferences in different cycles of software development. Similar situations happen in the hydro-thermal power resource scheduling (i.e., its total fuel cost of thermal generation and emission properties as optimal objective), which involves the allocation of power to all concerned units. Particularly, the concerned unit to the requirement of hydro-thermal power is different in the different time interval of one day. And, the worker of the hydro-thermal power resource scheduling (while can saw as DM) for objective optimization may exist in the different regions of interest to satisfy other requirements from concerned units.

The concept of handling the DM's preference has been wildly recognized and studied [24, 25], resulting in an emerging number of algorithms for static MOPs [26, 27]. This kind of MOEAs usually is called preference-based MOEAs, in the evolutionary computation community. Although most existing MOEAs effectively handle static MOPs with DMs preference, they are ineffective in addressing DMOPs with DMs preference if directly using these algorithms. The major reasons involve the following three parts. **1)** The essential characteristics of DMOPs and MOPs are different since DMOPs are dynamic in nature. **2)** DM's preference information may change with the optimization problems time-varying resulting in the serious diversity loss of population. **3)** Searching a POF approximation during DMOPs utilizing DMOEAs is in a limited time interval, which totally differs from the preference-based algorithms with a sufficient time for MOPs. Additionally, most existing DMOEAs focus on searching for the POS approximation with respect to the entire POF with well-distribution, which may be computationally expensive if DM only is interested in a subpart of the whole POF. Based on these aforementioned considerations, the methods incorporating the expressed preferences of DM into DMOEAs have been gradually presenting [20, 21] to focus the searching the solutions that correspond to the region of interest (ROI). To our best knowledge, this big concern is still in its infancy in the DMO fields.

The above-mentioned concerns have inspired us to search for the varying ROI on DMOPs in a brand-new way that is a novel DMOEAs based decomposition and changing preference (DACP). DACP incorporates the preference of DM into the dynamic search process and focuses on searching for a subset of POS approximation with respect to the ROI with well-distributed and well-converged features. The DACP takes into account two aspects. **a)** Considering how to assist the preference provided by DM, quickly search for the ROI in a limited time interval. **b)** Considering how to respond to change is to assist the population with searching a promising direction toward the corresponding ROI when a change occurs for DMOPs. Consequently, the major contributions of this paper include the following.

(1) With the dynamic emergence of DMOPs, the DM's ROI, which is defined by that DM gives both the preference point and the neighborhood size, may change with time-varying DMOPs in the whole optimization process. Meanwhile, our algorithm moves the well-distributed reference points located in the neighborhood range to around the preference point in each limited time interval. Leading the whole population evolves toward the ROI and improves the population diversity loss since the time-varying ROI.

(2) When a change occurs, a response strategy is performed to respond to the environmental change and considers the evolutionary direction toward the corresponding ROI. The population will be reinitialized according to a promising direction obtained by letting a few solutions evolve independently for a short time. Thereby, the response strategy not only accelerates the convergence speed of population toward the ROI in the new environment but also balances the diversity loss of population from environmental change and ROI change.

The rest of this paper is organized as follows. Section 2 presents some basic definitions, related works on DMOEAs for DMOPs, related works on combining DM preference into DMOEAs, challenges regarding combining DM preference into DMOEAs for handling DMOPs. Section 3 shows the proposed algorithm, including the entire framework of DACP, optimization Module, updating preference, dynamic handing strategy. Section 4 introduces the experimental settings. The experimental results and discussion are presented in Section 5 in detail. Section 6 concludes this paper and describes directions for future work.

2. Preliminaries

This section introduces four subsections mainly around basic definitions [15, 28], related works on DMO, related works on the preference of DM, and the challenges of handling the DM's preference for DMOPs.

2.1. Basic Definitions

In this paper, we only consider the minimization of DMOPs with unconstrained, which can be defined as follows:

$$\begin{cases} minimize \;\; F(x,t) = (f_1(x,t), f_2(x,t), ..., f_m(x,t))^T, \\ subject \;\; to \;\; x \in \Omega, \end{cases} \tag{1}$$

where $t \in T = \{0, 1, 2, \cdots\}$ represents the time index whose definition has been amply demonstrated [29]; $\Omega = \prod_{i=1}^{n} [a_i, b_i] \subset R^n$ defines the feasible region of the decision space, where a_i and b_i are the lower and upper bounds of the ith variable x_i, respectively; $x = (x_1, x_2, \cdots, x_n)^T$ is the n-dimensional decision variable vector. $F = (f_1, f_2, \cdots, f_m)^T$ is the m-dimensional objective vector.

2.2. Related Works on DMOEAs for DMOPs

In DMO, researchers mainly work on several important aspects, involving test problems, performance metrics, and algorithm design (i.e., the design of DMOEAs). Since DMOEAs are a kind of efficient tool for solving DMOPs, algorithm design has attracted increasing attention in DMO[30]. Consequently, a great deal of progress has been made about the design of DMOEAs in recent decades [31]. These DMOEAs in most existing DMO literature are classified as prediction approaches [32], diversity approaches [33], memory approaches [34, 35].

If the current population has converged before a change occurs, the loss of population diversity may exist in the new environment. Hence, how to address the loss of population diversity is an urgent issuer in the new environment. In most DMO literature, this crucial issue has been well addressed by mutation of existed solutions [36], multi-population ways [37], other methods [38]. Those technologies are generally called diversity approaches, which are effective in exploring the new search space whenever there is a change. Besides, most existing DMOPs are cyclic. Consequently, for memory approaches, when the new environment is similar to one of the previous environments [34, 39], the previous solutions usually are introduced or transferred to the new environment.

The memory approaches can help the population to accelerate the convergence speed in the new environment for handling DMOPs with cyclic. Additionally, prediction approaches [19] usually are based on computational models to respond to change, such as Kalman Filter [16] and autoregression (AR) model [39], to balance the convergence and diversity. They utilize the population information from historical environments to help the current population quickly track the varying POF before the next change occurs.

These several kinds of algorithms can effectively and efficiently handle the majority of DMOPs [40]. They mainly focus on searching the approximation of POS with respect to the whole POF with well-distributed before the next change occurs, which is the final form of currently most DMOEAs. However, this form may be difficult for the DM to select the solution set in the ROI from a large number of candidate solutions before the next change occurs. And it also inadvertently increases excessive computational burden during the evolutionary process [33]. Consequently, in recent years, a few work [23] presents combing the preference of DM into DMOEAs, which searches for ROI according to the preference information obtained by DM.

2.3. Related Works on combining DM preference into DMOEAs

In many real-world optimization applications, it is very common that DM by expressed preference [41, 42] way presents their desired ROI in objective space for DMOPs. Additionally, combining the DM's preference into the evolutionary optimization process has been widely studied and applied for static MOPs [27, 26]. However, for DMOPs is still scarce in DMO. Here, we mainly discuss some works integrating the preference of DM into DMOEAs for DMOPs.

To the best of our knowledge, the concept of the preferences of DM for DMOPs has been proposed by Deb et al. for the first time in [36]. In this paper, the authors only present the concept of DM preference under dynamic environments but have not discussed it in detail. In 2016, Helbig et al. in [23] analyzed key challenges and future directions of DMO in detail. And, [23] clearly indicates that one of the future challenges in DMO is how to design DMOAs that can efficiently incorporate the DM preferences into the dynamic evolutionary process. In [23], it only demonstrates that the work about incorporating decision making or preferences of the DM into DMOAs is rare, which suggests that more researches need to be required on searching for solutions based on DM's preferences.

In 2020, a knee-guided prediction approach [20] for DMO was proposed by Zou et al., which maintains non-dominated solutions near knees and boundary regions in the dynamic environment. The knee of problems in this paper is regarded as the final optimal solution which DM prefers. In 2018, Nebro put forward an approach [21], which allows DM preferences to be incorporated into the search process for DMOPs. It indicates in [21] that the DM can not only express her/his preferences utilizing one or more preference points, but these points can also be modified interactively. Although the algorithms with handling the preferences of DM for DMOPs are infrequent, they will gradually become a concerning topic in the evolutionary computing community in future research. Summarily, combining the DM preference into the evolutionary process for DMOPs is in infancy, so much effort to achieve still is required.

2.4. Challenges regarding combining DM preference into DMOEAs for DMOPs

Generally, in real-world applications, the ultimate goal of solving DMOPs is to provide a subset of POS approximation to DM. Afterward, DM can select the best appropriate solutions based on DM preferences, quickly and precisely. Consequently, handling the preference of DM for DMOPs requires optimal algorithms to quickly track the varying POF and/or POS before the next change occurs and search for solutions with well-diversity located in the ROI. This section mainly discusses some challenges.

1) Searching for a final solution satisfying for DM is a difficult issue because most DMOPs are unknown for DMs and researchers, which poses big challenges in providing specific DM preferences during exhibitions for researchers. Therefore, the preference expressed by DM is regarded as rough information from the researchers point of view. To better satisfy the DM preference for DMOPs, researchers can provide a subset of approximate POS according to DM preference information to choose from rather than a final solution.

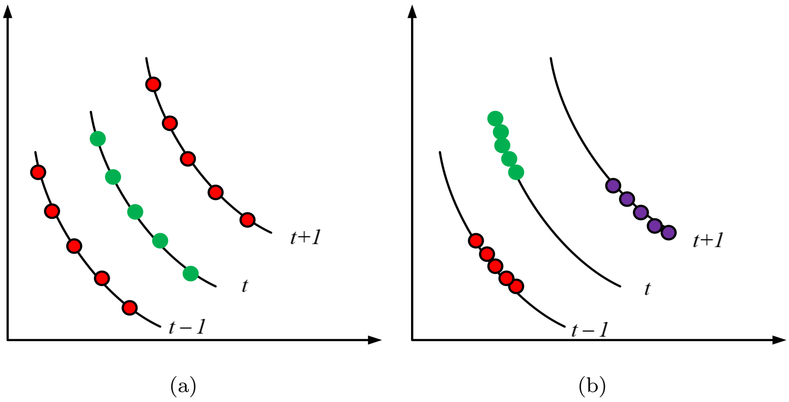


Figure 1: An example that uniformly distributed weights in ROI in a 2-m objective space.

2) Driving the population towards a particular region of the corresponding POF could significantly decrease population diversity when a change of ROI is detected in dynamic environments. Furthermore, once a change is detected, and preference information is updated, the non-diversified population whether evolves from its local geographical position in the search space towards the new ROI or not, which is a tremendous challenge for DMOEAs and preference-based MOEAs. For the convenience of discussing, we take an $m = 2$ dimensional decision space as an example (i.e., Fig. 1). In this example, we consider the diversity loss which is caused by the time-varying ROI. Fig. 1 supposes that the red, green and purple points respectively represent the solutions of ROI on the corresponding POF at $(t-1)$th, tth, and $(t+1)$th time interval. Most DMOEAs focus on searching for the whole POF approximation of DMOPs at each time interval (such as 1(a)). However, DM may only interest in a part on POF at any time interval, such as red, green, and purple points of 1(b) (i.e., ROI). Particularly, for the $(t+1)$th time interval, the ROI from t may totally different from the ROI in the new environment. If using the population, which approximates of ROI at tth time interval, directly evolves to get the approximates of ROI at $(t+1)$th time interval, this is very difficult. Therefore, designing an excellent response strategy is an urgent assignment to improve diversity loss due to the dynamic environment and time-varying ROI.

3. Proposed Algorithm

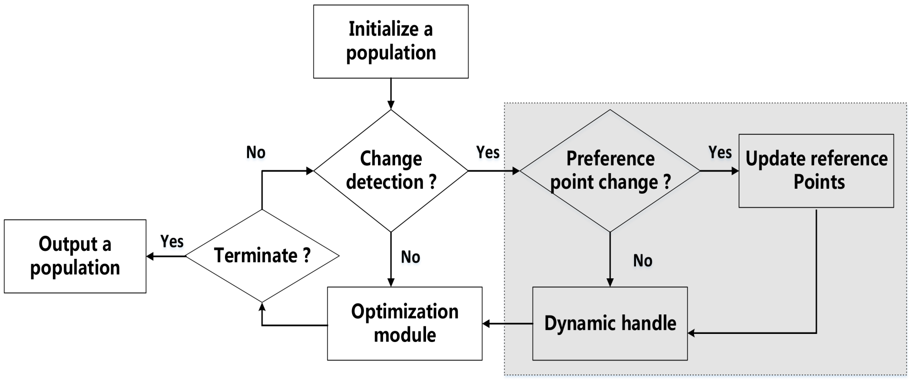


Figure 2: Flow chart of DACP.

In this section, we present an overall flowchart of the proposed algorithm in Fig. 2. From this figure, our proposed algorithm consists of three modules: *optimization*, *updating reference points* and *dynamic handing*. In each iteration, change detection is done by the approach in [16], which assumes an environmental change if the objective values of random detectors are different before and after reevaluation. If there is a change, the updating reference points module is to shift evenly distributed reference points to new positions near the DM-supplied preferred vector when a change of preference point is also detected. Afterward, a dynamic handing module, which is designed to generate new solutions that are close to ROI on new POS, is performed. Otherwise, an optimization module based on MOEA/D [29] in Fig. 2 is provided for finding the DM's preferred solutions. In the following subsections, we will introduce each module step by step.

3.1. Optimization Module

The goal of optimization modules is to search for the DM's preferred solutions. In practice, any decomposition-based optimization approach can be used for achieving this purpose well. In this case, MOEA/D [29] is selected for optimizing the solutions when t remains the same. Meanwhile, the basic idea of MOEA/D is to decompose a MOP into several subproblems and simultaneously optimizes them by neighborhoods. The neighborhoods are generated by the reference points and the distances. For decomposition-based approaches, this paper uses the Tchebycheff function [43] defined as:

$$\begin{aligned} &minimize\ \ g(x|w, z^*) = \max_{1 \le i \le m} |f_i(x) - z_i^*|/w_i \\ &subject\ \ to\ \ x \in \Omega \end{aligned} \tag{2}$$

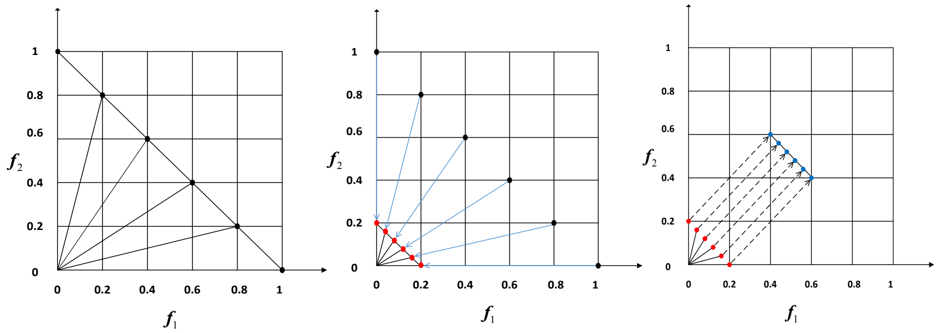
where $z^* = (z_1^*, z_2^*, \cdots, z_m^*)$ is the ideal point and $w = (w_1, w_2, \cdots, w_m)$ is the reference point; z_1^* and w_i are the ith dimension in the objective space from z^* and w, respectively.

It is worth mentioning that the purpose of MOEA/D is to search the whole POF approximation with well-distribution according to its' generated way of reference points. This is different from this paper's goal that searches the approximate subset of solutions according to the preference information in ROI in a dynamic environment. Consequently, the following subsection describes how to update the reference points to construct well-distribution vectors around a preference point in detail to track the ROI in a limited time interval quickly.

3.2. Updating reference points

It is known that existing DMOPs with conflicting objectives involve finding a set of tradeoff solutions by MOEA/D. They pose enormous challenges for selecting a final subset of POS approximation expected by DM, which is caused by the following several aspects. **a)** DMOPs involve more than one objective with conflicting each other and these POF and/or POS are change over time; **b)** DMOPs are unknown for DM; **c)** The time of finding each POF and/or

POS during the DMO process is limited. The first three points may be difficult for DM in providing a reasonable preference. Hence, DM usually provides a neighborhood range around the preference point of DM to have a bigger probability of making the desired solutions located at ROI. In other words, the ROI is defined by both the preference point and the size of the neighborhood in any limited time interval, in which the size of the neighborhood is fixed in the whole optimal process.



(a) Evenly distributed reference points (b) Moved reference points (c) Shifted reference points

Figure 3: An example that uniformly distributed weights in ROI in a 2-D decision space.

With this consideration in mind, reference points, which play an important role in decomposition-based methods, are shifted to around the preference point. Thereinto, the neighborhood's size is fixed to 0.2, representing the size around the preference point on each objective in the objective space in the whole optimization process. For the convenience of discussion, we take a $D = 2$ dimensional objective space as an example in Fig. 3. In Fig. 3, assuming that the preference point is (0.5, 0.5) and the size of the neighborhood is fixed to 0.2. Fig. 3 illustrates the configuration of moving reference points to around the preference point in it. First, reference points in Fig. 3(a) (i.e., black points in Fig. 3(a)) are evenly generated by employing the Das and Dennis' proposed method [43]. In this paper, the algorithms need to search the approximate solutions in the ROI, rather than a whole POF. As a result, the reference points in

Fig. 3(a) need to be shifted to around the preference point to guide the evolution of the population toward ROI effectively. Each reference point $\mathbf{p}$ from the black reference points is moved to the corresponding $\mathbf{p}^{'}$ from the red reference points in 3(b) according to the following formula:

$$\mathbf{p}^{'} = \mathbf{p} * \mathbf{b}, \tag{3}$$

where $\mathbf{b}$ is a vector, representing the size of preference provided by DM on each objective in objective space.

In order to get the corresponding reference point p^u from the red reference points in Fig. 3(c) based on a preference point provided by DM, $\mathbf{p}^{'}$ needs to be shifted to the corresponding reference point $\mathbf{p}^u$ from the blue reference points in Fig. 3(c) close to the ROI. Accordingly, $\mathbf{p}^u$ is calculated by

$$\mathbf{p}^u = \frac{\mathbf{p}^{'} + \mathbf{r}}{||\mathbf{p}^{'} + \mathbf{r}||}, \tag{4}$$

where $\mathbf{r}$ represents preference point and $|| * ||$ represents the L^2 norm. It is worth mentioning in the design of the proposed algorithm that the number of the reference points is the same as the population's size.

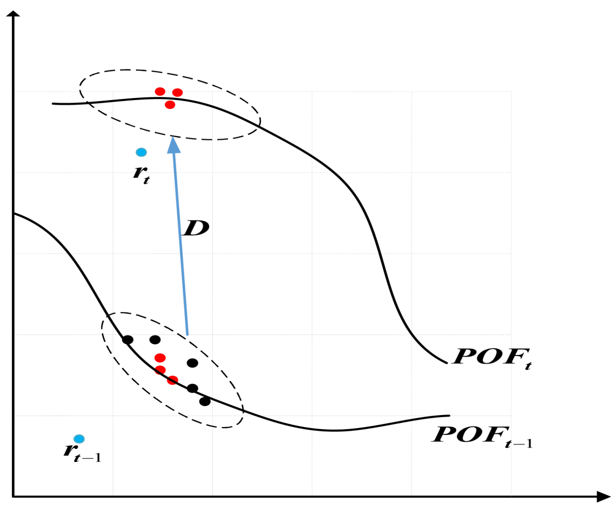


Figure 4: An example of dynamic handing strategy between two consecutive time intervals in a 2-D decision space and the population size $N = 8$.

Algorithm 1 change response

Require: current population, P; current time instant, t; the centroid at time t, C_t; the centroid of R at time t, C_R; i (the current number of change); m (the number of objectives).

Ensure: updated population P and empty set r.

if $t \equiv 0$ **then**
  Create empty set r;
end if
$j = 0$
if $t \equiv 0$ or $rand < 0.5$ **then**
  while $j < m$ **do**
    $r_t^j = rand$, $j = j + 1$;
  end while
else
  $r_t = r_{t-1}$;
end if
$r \leftarrow r_t$;
if $r_t \neq r_{t-1}$ or $t \equiv 0$ **then**
  The preference point in the new environment is judgment as change.
  The set of reference points is updated, which is described in detail by Section 3.2.
end if
Select some solutions from P to R according to the Euclidean distance and R evolves independently for some generations ;
Compute the centroid C_P and C_R by Eq. 5 and Eq. 6, respectively;
Compute the search direction D by Eq. 7;
for $x_t \in$ P\R **do**
  Reinitiaze x_t by Eq. 8 and reevaluate the new solution x_{t+1}.
end for

3.3. Dynamic handing strategy

A certain DMOP can be regarded as a series of MOPs in a dynamic evolutionary process, in which a selected algorithm optimizes each MOP to search for ROI during a limited time interval. Meanwhile, the preferences of DMs may be different in different MOPs resulting in ROI over time. Additionally, although DMOPs change over time, the preferences about these problems may remain unchanged in any two continuous time steps. Particularly, the preference information provided by DM exists possibly the same within any two continuous time steps. In practice, the preference of DM should be unknown in future environmental changes. As a result, they in this paper are set randomly and maybe the same under a certain probability. Particularly, in each limited time interval, this probability is set to 0.5, meaning that there is a half chance where the previous preference of DM can be maintained in the new environment.

Suppose that we create a empty set $r = r_1, r_2, \cdots$ to preserve the preference points provided by DM, in which $r_i = r_i^1, r_i^2, \cdots, r_i^m$ at ith time step and $i \in 1, 2, \cdots, t$. For easy understanding, the implementation of possible producing preference points is described in detail in lines from 1 to 16 of Algorithm 1. Particularly, the preference point (i.e., r_t) is judgment as unchanged if it meets $r_t \equiv r_{t-1}$ in the new environment, which r_{t-1} is the preference point at $t-1$ time interval. Else, the preference point in the new environment is regarded as change. For the experiment of fairness, all preference points are saved to a set r and are applied to the optimization process of the compared algorithms to search for approximating ROI, respectively.

Besides, when a change is detected, some actions need to be taken to respond to change. Besides, part of solutions that meet DM preference during the last environment is selected to explore the new promising region that is close to ROI in the new environment. In Fig. 4, we provide an example to illustrate the process. Particularly, we select some solutions (to construct a new set R with the size h) from the current population based on the smallest Euclidean distance from each solution to preference point r_{t-1}. Afterward, suppose that C_t is the centroid of P_t, and it can be calculated by

$$C_t = \frac{1}{|P_t|} \sum_{x_t \in P_t} x_t, \tag{5}$$

where P_t is the non-dominated set of P at the time t; $|P_t|$ is the size of the set P_t.

Additionally, let the solutions in R (e.g., the red solutions at nearby POF_{t-1} in Fig. 4) independently evolve for a short time by the genetic way, which is terminated when its total evaluation number is N. Afterward, by this means obtains a new set R (e.g., the red solutions at nearby POF_t in Fig. 4) which more approximate ROI of the new environment. Then, the centroids C_R of R

can be calculated as follows:

$$C_R = \frac{1}{|R|} \sum_{x_t \in R} x_t, \tag{6}$$

where $|R|$ is the size of R.

The moving direction D to guide the population evolution toward ROI is called the direction of autonomic evolution, which is calculated by

$$D = C_R - C_t. \tag{7}$$

For each solution x_t in P\R, it is regenerated as follows:

$$x_{t+1} = x_t + D, \tag{8}$$

We choose the direction of autonomic evolution to replace the direction based on previous information. Particularly, the preference points obtained by DM in different environments may be completely different, which may be causing the change of ROI with time. Consequently, the information from the previous population may be useless for the new environment. Nevertheless, the direction of autonomic evolution can overcome the issuer above because it utilizes the population's spontaneous evolution towards the direction of the new ROI. Hence, the autonomic evolution direction is promising.

3.4. Computational Complexity Analysis of DACP

In each generation, the computational complexity of DACP in this paper is analyzed mainly by three modules, i.e., optimization, updating reference points and dynamic handing. Detailedly, [44] mentions that [30] (i.e., optimization module in this paper) is the computation cost of $O(mLN)$ that the reference points generate. L is the number of subproblems. At the beginning stage of each environment, updating reference points moves the well-distributed reference points to around the preference point, which needs the time cost of $O(Nn)$. In dynamic handing, they select some solutions from the population to R according to the smallest Euclidean distance from each solution to the preference point

r_{t-1}. Consequently, the solutions from R, choice and evolution need $O(2N)$. Afterward, the rest solutions are forecasted according to the center direction, which requires $O(N - |R|)$. Due to $|R|$ is very smaller than N, $O(|R|)$ is overlooked in the paper. Summarily, the whole DACP's process needs $O(3N + mLN)$.

4. Experimental Settings

4.1. Benchmark Problems

Test problems with dynamic changes play an important role in assessing and analyzing the performance of DMOEAs when searching for the ROI according to DM preference information. Therefore, the FDA, dMOP and DF1-DF4 benchmark test suites [39, 45, 46], which possess bi-objective and tri-objective benchmark functions with various properties, are used to study the performance of the proposed algorithm systematically. Particularly, the FDA benchmark suite is commonly used in the performance of DMOEAs, in which two of the problems (FDA3 and FDA5) have time-varying density distribution of solutions along with the POF. Moreover, the dMOP benchmark problems [39] are an extension of the FDA benchmark suite [45] to test further performances of DMOEAs.

4.2. Performance Measures

To evaluate the algorithms' performance for approximating the ROI, we employ two authoritative metrics in the evolutionary computation community. They are the mean generational distance (MGD) [47] and the mean hypervolume (MHV) [19], which are defined as the average of the GD [28] and HV [1] for few time steps in a run for DMOPs, respectively. Assuming that DMOPs are real-world problems, the technology of picking points in [48] is employed at $t = 1, 2, 3, \cdots$, respectively.

4.3. Compared Algorithms

Several state-of-the-art and popular algorithms are selected for experimental studies to demonstrate that the proposed algorithm is very competitive in searching for the ROI. They are:

(a) *NUMS-MOEA/D:* Li et al. in [49] put forward integration of preferences in decomposition multi-objective optimization (i.e., NUMS-MOEA/D), which can deal with approximating the preferred solutions in the ROI well for static MOPs. The basic idea of NUMS-MOEA/D is by the originally evenly distributed reference points on a canonical simplex that can be mapped to new positions close to the DM-specified aspiration-level vector. Here, we combine dynamic handling strategies, which are common in the compared experiment in DMO with NUMS-MOEA/D, to respond to change for D-MOPs, i.e., NUMS-MOEA/D-A and NUMS-MOEA/D-B. Particularly, in NUMS-MOEA/D-A, 20% the population are randomly reinitialized when a change occurs; in NUMS-MOEA/D-B, 20% of the population are reinitialized with mutated solutions of existing solutions when a change occurs.

(b) *DNSGA-II:* Deb et al. proposed DNSGA-II in 2007s [36]. DNSGA-II includes two versions (i.e. DNSGA-II-A and DNSGA-II-B), which always are selected for the compared experience in DMO. In the first version, 20% of the population is replaced with randomly created solutions whenever there is a change in problem; in the second version, 20% of the population is replaced with mutated solutions of existing solutions, which can deal with the DMOPs well, when there exists a small change. Additionally, DNSGA-II mainly focuses on searching for the approximation of whole POF, rather than the approximation of ROI in terms of the preference provided by D-M. Accordingly, the Pareto dominance in DNSGA-II was replaced as g-dominance [24], i.e., g-NSGA-II-A and g-NSGA-II-B from Tables 2 and 5. Particularly, g-dominance presented by Molina et al., which approximates the efficient set around the area of the most preferred point without using any scalarizing function.

4.4. Parameter Settings

The parameters of the compared algorithms in the experiment were referenced from their original papers. Some key parameters have been summarized in Table 1 and make briefly explained as follows:

Table 1: Parameter settings

Number of decision variables, n	10 for all test problems;
Population size	100 and 200, for $m = 2$ and $m = 3$;
Number of the environmental changes:	50;
Dynamic setting	Severity of changes: $n_t = 5, 10, 20$; Frequency of change: $\tau_t = 10, 20, 30$;
The neighborhood size:	0.2;
Number of detectors:	10 detectors for all algorithms;
Terminating criterion:	$\tau > \tau_t * t$;
The significance level of [50]	5% significance level;

1) The Wilcoxon rank-sum test has been used to indicate significance between different results at the 5% significance level. The number of decision variables is set as $n = 10$ with the assigned population size of $N = 100$ and $N = 200$, respectively, for 2 and 3 objective problems.
2) Dynamic environments have been simulated with $num = 50$ environmental changes. The severity of change (n_t) is fixed to $n_t = 10$ and the frequency of change (τ_t) is set as $\tau_t = 10, 20, 30$ to study the impact of change frequency on algorithms. Besides, the neighborhood size around the preference point provided by DM is 0.2;
3) In our empirical studies, the total number of generations is set to $\tau > \tau_t * t$, in which τ is the generation counter. Each algorithm ran independently 20 times on each problem.

It is worth noting that other parameters (except for the parameters above) in the compared algorithms used the same settings as in their original studies.

5. Experimental Studies

To show the remarkable performance of the proposed algorithm on searching for the ROI, the proposed algorithm is compared with four state-of-the-art algorithms. And the statistical results in MGD metric, which is obtained by different algorithms, have been presented in Tables 2 and 5. Thereinto, the best results which are obtained by one of the five algorithms are highlighted in boldface. Besides, when the Wilcoxon rank-sum test has been carried out for the experimental results, ‡ and † indicate DACP performs significantly better than and equivalently to the corresponding algorithm, respectively.

5.1. Experimental Results and Discussion

The severity of change (i.e., n_t) fixed as 10, and the change frequency (i.e., τ_t) has been set as 10, 20 and 30, respectively, to study the influence of the change frequency on the searching ability of algorithms in dynamic environments. The τ_t settings from 10 to 30 mean that the change frequency is from high to low with an increase of τ_t. The other parameters have been introduced in Section 4.4. Table 2 lists the MGD values obtained by all compared algorithms on a series of DMOPs with different change frequencies.

From the results shown in Table 2, we can clearly see that the values on the MGD metric obtained by the proposed algorithm are very small. This situation indicates that the solutions obtained by DACP are much closer to the ROI defined by DM expressed preferences, compared with other algorithms on the majority of problems. Meanwhile, this demonstrates that DACP can generate a good population that approximates the ROI and shows that it can excellently track the changing POS and/or POF for DMOPs. However, for F-DA4, it is clearly seen that the performance of the two algorithms, i.e., NUMS-MOEA/D-A and NUMS-MOEA/D-B, is very outstanding than DACP. This may because FDA4 will lead to a hypersurface in expanding from a curve to a surface for a change [45]. Generally speaking, the decomposition-based algorithms are outperformance, especially DACP. i.e., the metric values obtained

Table 2: Mean and standard deviation values of MGD obtained by five algorithms

Fun.	(τ_t, n_t)	g-NSGA-II-A	g-NSGA-II-B	NUMS-MOEA/D-A	NUMS-MOEA/D-B	DACP
FDA1	(10,10)	1.426e-1(1.219e-2)‡	1.118e-1(4.294e-3)‡	3.533e-2(7.198e-3)‡	3.008e-2(5.025e-3)‡	**5.963e-3(2.376e-7)**
	(20,10)	2.167e-2(2.110e-3)‡	2.031e-2(1.330e-3)‡	6.297e-3(1.060e-3)‡	6.415e-3(8.669e-4)‡	**1.074e-3(1.553e-8)**
	(30,10)	6.675e-3(4.245e-4)‡	6.149e-3(9.381e-4)‡	2.734e-3(3.589e-4)‡	2.733e-3(3.185e-4)‡	**2.754e-4(3.847e-10)**
FDA2	(10,10)	2.257e-1(2.727e-3)‡	2.272e-1(8.791e-3)‡	2.339e-1(7.289e-3)‡	2.394e-1(6.184e-3)‡	**2.359e-3(5.352e-8)**
	(20,10)	2.124e-1(1.406e-3)‡	2.165e-1(1.997e-3)‡	2.275e-1(5.075e-3)‡	2.368e-1(3.868e-3)‡	**3.574e-4(4.011e-9)**
	(30,10)	2.090e-1(4.562e-4)‡	2.103e-1(1.504e-3)‡	2.221e-1(4.697e-3)‡	2.318e-1(5.433e-3)‡	**1.042e-4(1.074e-10)**
FDA3	(10,10)	1.263e-1(3.615e-3)‡	1.156e-1(3.690e-3)‡	1.748e-1(6.639e-3)‡	1.721e-1(8.033e-3)‡	**1.932e-2(1.557e-6)**
	(20,10)	6.019e-2(3.025e-3)‡	6.166e-2(3.653e-3)‡	1.412e-1(6.716e-3)‡	1.335e-1(4.281e-3)‡	**1.568e-2(4.929e-7)**
	(30,10)	5.659e-2(2.598e-3)‡	5.589e-2(2.515e-3)‡	1.260e-1(5.930e-3)‡	1.327e-1(4.928e-3)‡	**1.549e-2(1.499e-7)**
FDA4	(10,10)	2.490e-1(1.383e-2)‡	4.117e-1(2.223e-2)‡	1.120e-2(1.109e-3)	**1.585e-2(1.340e-3)**	7.532e-2(1.357e-5)
	(20,10)	1.318e-1(8.171e-3)‡	1.413e-1(1.343e-2)‡	**5.684e-3(2.024e-4)**	6.210e-3(5.641e-4)	6.499e-2(1.781e-5)
	(30,10)	8.453e-2(7.696e-3)‡	7.564e-2(6.269e-3)‡	5.614e-3(8.947e-5)	**5.613e-3(1.529e-4)**	5.854e-2(2.114e-5)
FDA5	(10,10)	2.612e-1(2.550e-2)‡	2.500e-1(2.840e-2)‡	3.729e-1(9.430e-4)‡	3.722e-1(1.843e-3)‡	**1.169e-1(7.528e-6)**
	(20,10)	2.815e-1(1.226e-2)‡	3.077e-1(3.944e-2)‡	3.761e-1(8.102e-4)‡	3.766e-1(6.396e-4)‡	**1.174e-1(3.963e-6)**
	(30,10)	3.341e-1(6.047e-3)‡	3.306e-1(2.057e-2)‡	3.758e-1(2.876e-4)‡	3.762e-1(7.731e-4)‡	**1.127e-1(1.631e-6)**
dMOP1	(10,10)	1.383e-1(4.180e-2)‡	1.066e-1(3.127e-2)‡	3.859e-2(9.280e-3)‡	2.468e-2(8.590e-3)‡	**9.745e-3(7.646e-6)**
	(20,10)	1.007e-2(4.550e-3)‡	1.103e-2(6.693e-3)‡	4.922e-3(6.681e-4)‡	7.413e-3(1.277e-3)‡	**2.355e-3(3.060e-7)**
	(30,10)	2.557e-3(1.167e-3)‡	2.589e-3(1.245e-3)‡	3.122e-3(4.563e-4)‡	5.528e-3(5.827e-4)‡	**1.497e-3(5.205e-9)**
dMOP2	(10,10)	1.677e-1(1.795e-2)‡	1.284e-1(1.183e-2)‡	5.030e-2(3.678e-3)‡	4.859e-2(1.167e-2)‡	**8.406e-3(7.304e-7)**
	(20,10)	1.790e-2(8.053e-4)‡	1.676e-2(9.179e-4)‡	9.492e-3(1.310e-3)‡	8.853e-3(9.737e-4)‡	**2.209e-3(1.542e-8)**
	(30,10)	6.070e-3(3.494e-4)‡	5.755e-3(3.029e-4)‡	4.372e-3(6.643e-4)‡	3.906e-3(4.272e-4)‡	**1.577e-3(6.514e-10)**
dMOP3	(10,10)	1.464e+2(2.862e+0)‡	1.262e+2(3.939e+0)‡	1.486e-1(1.520e-2)‡	1.512e-1(1.525e-2)‡	**1.606e-2(2.040e-6)**
	(20,10)	1.339e+2(6.140e+0)‡	1.208e+2(1.721e+0)‡	2.980e-2(4.927e-3)‡	2.965e-2(3.958e-3)‡	**3.069e-3(2.275e-7)**
	(30,10)	1.186e+2(3.211e+0)‡	1.113e+2(4.787e+0)‡	5.907e-3(1.176e-3)‡	5.890e-3(1.568e-3)‡	**6.571e-4(1.439e-8)**
DF1	(10, 10)	8.501e-2(4.338e-2)‡	1.276e-1(7.460e-2)‡	3.783e-2(2.144e-2)‡	5.516e-2(3.586e-3)‡	**3.432e-3(4.506e-4)**
	(20, 10)	2.109e-2(8.440e-4)‡	1.916e-2(1.221e-3)‡	8.235e-3(1.471e-3)‡	8.580e-3(1.100e-3)‡	**3.320e-3(7.187e-4)**
	(30, 10)	5.583e-3(1.250e-4)‡	5.318e-3(6.068e-5)‡	3.070e-3(5.203e-4)‡	3.324e-3(4.240e-4)‡	**3.001e-3(3.602e-4)**
DF2	(10, 10)	1.395e+2(1.040e+1)‡	1.227e+2(8.156e+0)‡	7.801e-2(4.283e-2)‡	1.159e-1(1.426e-2)‡	**6.447e-3(7.146e-4)**
	(20, 10)	1.281e+2(2.945e+0)‡	1.153e+2(3.342e+0)‡	2.139e-2(2.324e-3)‡	2.031e-2(4.321e-3)‡	**5.055e-3(1.226e-3)**
	(30, 10)	1.207e+2(8.443e+0)‡	1.079e+2(5.371e+0)‡	5.129e-3(2.730e-4)‡	4.631e-3(7.214e-4)‡	**4.541e-3(6.371e-4)**
DF3	(10, 10)	3.094e-1(5.015e-2)‡	2.714e-1(2.898e-2)‡	2.344e-1(5.351e-3)	**2.302e-1(2.996e-3)**	2.432e-1(1.034e-3))
	(20, 10)	**2.371e-1(3.027e-3)**	2.371e-1(5.218e-3)	2.431e-1(2.648e-3)†	2.375e-1(2.364e-3)	2.412e-1(1.824e-3)
	(30, 10)	2.474e-1(2.068e-3)†	2.449e-1(4.102e-3)†	2.437e-1(2.288e-3)†	2.414e-1(2.539e-3)†	**2.409e-1(8.725e-4)**
DF4	(10, 10)	3.965e-1(1.318e-1)‡	3.129e-1(5.416e-2)‡	1.376e-1(7.693e-3)	1.472e-1(9.369e-3)‡	**1.340e-1(2.138e-3)**
	(20, 10)	2.666e-1(4.296e-2)‡	2.129e-1(3.245e-2)‡	1.236e-1(1.519e-3)‡	2.375e-1(1.614e-3)‡	**1.213e-1(8.966e-4)**
	(30, 10)	1.817e-1(6.395e-3)‡	1.583e-1(6.049e-3)‡	1.212e-1(2.155e-3)†	1.347e-1(2.255e-3)‡	**1.164e-1(6.228e-4)**

Table 3: Median values of MGD obtained by five algorithms

Fun.	(τ_t, n_t)	g-NSGA-II-A	g-NSGA-II-B	NUMS-MOEA/D-A	NUMS-MOEA/D-B	DACP
FDA1	(10,10)	1.387e-1(1.219e-2)‡	1.116e-1(4.294e-3)‡	3.462e-2(7.198e-3)‡	2.766e-2(5.025e-3)‡	**5.963e-3(2.376e-7)**
	(20,10)	2.113e-2(2.110e-3)‡	2.020e-2(1.330e-3)‡	5.984e-3(1.060e-3)‡	6.643e-3(8.669e-4)‡	**1.074e-3(1.553e-8)**
	(30,10)	6.651e-3(4.245e-4)‡	6.076e-3(9.381e-4)‡	2.545e-3(3.589e-4)‡	2.733e-3(3.185e-4)‡	**2.754e-4(3.847e-10)**
FDA2	(10,10)	2.260e-1(2.727e-3)‡	2.276e-1(8.791e-3)‡	2.274e-1(7.289e-3)‡	2.408e-1(6.184e-3)‡	**2.359e-3(5.352e-8)**
	(20,10)	2.122e-1(1.406e-3)‡	2.154e-1(1.997e-3)‡	2.349e-1(5.075e-3)‡	2.368e-1(3.868e-3)‡	**3.574e-4(4.011e-9)**
	(30,10)	2.089e-1(4.562e-4)‡	2.102e-1(1.504e-3)‡	2.203e-1(4.697e-3)‡	2.318e-1(5.433e-3)‡	**1.042e-4(1.074e-10)**
FDA3	(10,10)	1.277e-1(3.615e-3)‡	1.150e-1(3.690e-3)‡	1.776e-1(6.639e-3)‡	1.728e-1(8.033e-3)‡	**1.932e-2(1.557e-6)**
	(20,10)	5.842e-2(3.025e-3)‡	6.009e-2(3.653e-3)‡	1.418e-1(6.716e-3)‡	1.335e-1(4.281e-3)‡	**1.568e-2(4.929e-7)**
	(30,10)	5.662e-2(2.598e-3)‡	5.658e-2(2.515e-3)‡	1.219e-1(5.930e-3)‡	1.327e-1(4.928e-3)‡	**1.549e-2(1.499e-7)**
FDA4	(10,10)	2.507e-1(1.383e-2)‡	4.032e-1(2.223e-2)‡	1.132e-2(1.109e-3)	**1.501e-2(1.340e-3)**	7.532e-2(1.357e-5)
	(20,10)	1.311e-1(8.171e-3)‡	1.418e-1(1.343e-2)‡	**5.651e-3(2.024e-4)**	6.210e-3(5.641e-4)	6.499e-2(1.781e-5)
	(30,10)	8.297e-2(7.696e-3)‡	7.415e-2(6.269e-3)‡	5.539e-3(8.947e-5)	**5.613e-3(1.529e-4)**	5.854e-2(2.114e-5)
FDA5	(10,10)	2.667e-1(2.550e-2)‡	2.503e-1(2.840e-2)‡	3.720e-1(9.430e-4)‡	3.719e-1(1.843e-3)‡	**1.169e-1(7.528e-6)**
	(20,10)	2.825e-1(1.226e-2)‡	2.700e-1(3.944e-2)‡	3.760e-1(8.102e-4)‡	3.766e-1(6.396e-4)‡	**1.174e-1(3.963e-6)**
	(30,10)	3.296e-1(6.047e-3)‡	3.341e-1(2.057e-2)‡	3.755e-1(2.876e-4)‡	3.762e-1(7.731e-4)‡	**1.127e-1(1.631e-6)**

by the decomposition-based algorithms are outstandingly small compared with the other algorithms on the majority of problems.

Additionally, in Table 3, we extract the median on the MGD in each run, then calculate the median's average and standard deviation values. In Table 4, noting that IQR is quartile on the MGD in each run, then calculate the IQR's average and standard deviation values. The results in Tables 3 and 4, are roughly consistent with the MGD ones illustrated in Table 2. Clearly, DACP is more promising than the other algorithms to solve most FDA and dMOP instances.

Finally, it is worth noting that the influence of the frequency of change is very apparent in the algorithms' ability to search the time-varying ROI. Particularly, the performance on all selected algorithms is gradually better with the increasing τ_t, which demonstrates that the effect of τ_t decreases when environmental changes from 10 to 30. All in all, the performance of the proposed algorithm outperforms other algorithms on the majority of problems for dynam-

Table 4: IQR values of MGD obtained by five algorithms

Fun.	(τ_t, n_t)	g-NSGA-II-A	g-NSGA-II-B	NUMS-MOEA/D-A	NUMS-MOEA/D-B	DACP
	(10, 10)	2.113e-2(2.026e-3)‡	1.982e-2(1.330e-3)‡	5.563e-3(1.060e-3)‡	6.609e-3(8.669e-4)‡	**1.012e-3(1.553e-8)**
FDA1	(20, 10)	2.113e-2(2.110e-3)‡	2.020e-2(1.330e-3)‡	5.984e-3(1.060e-3)‡	6.643e-3(8.669e-4)‡	**1.063e-3(1.553e-8)**
	(30, 10)	2.284e-2(2.110e-3)‡	2.171e-2(1.330e-3)‡	6.294e-3(1.060e-3)‡	5.680e-3(8.669e-4)‡	**1.113e-3(1.553e-8)**
	10, 10)	2.110e-1(1.406e-3)‡	2.145e-1(1.997e-3)‡	2.276e-1(5.075e-3)‡	2.294e-1(3.868e-3)‡	**3.553e-4(4.011e-9)**
FDA2	(20, 10)	2.122e-1(1.406e-3)‡	2.154e-1(1.997e-3)‡	2.349e-1(5.075e-3)‡	2.364e-1(3.868e-3)‡	**3.571e-4(4.011e-9)**
	(30, 10)	2.137e-1(1.406e-3)‡	2.177e-1(1.997e-3)‡	2.361e-1(5.075e-3)‡	2.396e-1(3.868e-3)‡	**3.582e-4(4.011e-9)**
	10, 10)	5.778e-2(3.025e-3)‡	5.913e-2(3.653e-3)‡	1.349e-1(6.716e-3)‡	1.302e-1(4.281e-3)‡	**1.423e-2(4.929e-7)**
FDA3	(20, 10)	5.842e-2(3.025e-3)‡	6.009e-2(3.653e-3)‡	1.418e-1(6.716e-3)‡	1.332e-1(4.281e-3)‡	**1.572e-2(4.929e-7)**
	(30, 10)	6.122e-2(3.025e-3)‡	6.112e-2(3.653e-3)‡	1.442e-1(6.716e-3)‡	1.357e-1(4.281e-3)‡	**1.632e-2(4.929e-7)**
	10, 10)	1.267e-1(8.171e-3)‡	1.314e-1(1.343e-2)‡	**5.549e-3(2.024e-4)**	5.934e-3(5.641e-4)	6.325e-2(1.781e-5)
FDA4	(20, 10)	1.311e-1(8.171e-3)‡	1.418e-1(1.343e-2)‡	**5.651e-3(2.024e-4)**	6.165e-3(5.641e-4)	6.487e-2(1.781e-5)
	(30, 10)	1.393e-1(8.171e-3)‡	1.538e-1(1.343e-2)‡	**5.792e-3(2.024e-4)**	6.213e-3(5.641e-4)	6.499e-2(1.781e-5)
	10, 10)	2.603e-1(1.226e-2)‡	2.631e-1(3.944e-2)‡	3.758e-1(8.102e-4)‡	3.761e-1(6.396e-4)‡	**1.161e-1(3.963e-6)**
FDA5	(20, 10)	2.825e-1(1.226e-2)‡	2.700e-1(3.944e-2)‡	3.760e-1(8.102e-4)‡	3.777e-1(6.396e-4)‡	**1.173e-1(3.963e-6)**
	(30, 10)	2.868e-1(1.226e-2)‡	2.714e-1(3.944e-2)‡	3.768e-1(8.102e-4)‡	3.792e-1(6.396e-4)‡	**1.184e-1(3.963e-6)**

ic environments, which is significant on the Wilcoxon rank-sum test.

5.2. Influence of Severity of Change

The change frequency (i.e., τ_t) has been fixed as 20, and the severity of change (i.e., n_t) has been set as 5, 10 and 20, respectively, to study the influence of the severity of change on the algorithms' performance in dynamic environments. This means that the severity of change for DMOPs is from large to small with the increase of n_t, i.e., from a large movement step-size for severe changes ($n_t = 5$) to a small movement step-size for slight ones ($n_t = 20$). The other parameters are the same with Section 4.4. Table 5 lists the MGD values obtained by all compared algorithms on a series of DMOPs with different severity of change.

As observed in Table 5, the setting of the change severity has a significant effect on the algorithms' performance because all algorithms' MGD values become better and better (i.e., more and more small) with the increase of n_t.

Table 5: Mean and standard deviation values of MGD obtained by five algorithms

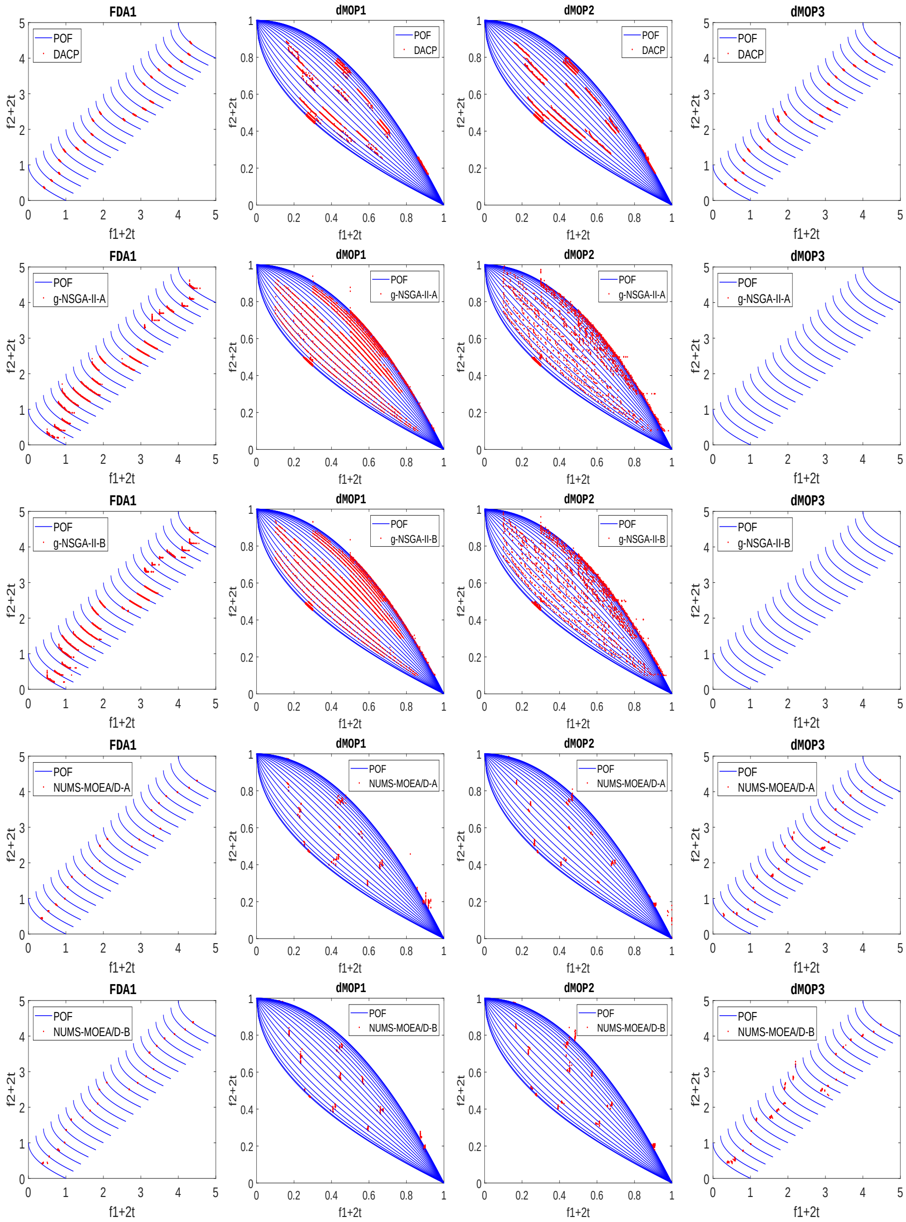
Fun.	(τ_t, n_t)	g-NSGA-II-A	g-NSGA-II-B	NUMS-MOEA/D-A	NUMS-MOEA/D-B	DACP
	(20,5)	6.044e-2(4.636e-3)‡	5.759e-2(4.757e-3)‡	9.943e-3(1.186e-3)‡	9.775e-3(1.273e-3)‡	**1.878e-3(2.262e-8)**
FDA1	(20,10)	2.167e-2(2.110e-3)‡	2.031e-2(1.330e-3)‡	6.297e-3(1.060e-3)‡	6.415e-3(8.669e-4)‡	**1.074e-3(1.553e-8)**
	(20,20)	9.420e-3(1.855e-3)‡	8.723e-3(1.669e-3)‡	4.494e-3(7.423e-4)‡	4.314e-3(1.118e-3)‡	**7.377e-4(1.905e-9)**
	(20,5)	2.178e-1(1.686e-3)‡	2.174e-1(1.749e-3)‡	2.310e-1(7.039e-3)‡	2.375e-1(5.675e-3)‡	**3.942e-4(2.410e-9)**
FDA2	(20,10)	2.124e-1(1.406e-3)‡	2.165e-1(1.997e-3)‡	2.275e-1(5.075e-3)‡	2.368e-1(3.868e-3)‡	**3.574e-4(4.011e-9)**
	(20,20)	2.346e-1(1.423e-3)‡	2.373e-1(3.766e-3)‡	2.468e-1(5.285e-3)‡	2.568e-1(6.746e-3)‡	**3.607e-4(1.914e-9)**
	(20,5)	7.015e-2(2.469e-3)‡	6.975e-2(2.363e-3)‡	1.080e-1(6.283e-3)‡	1.069e-1(6.761e-3)‡	**1.875e-2(4.214e-7)**
FDA3	(20,10)	6.019e-2(3.025e-3)‡	6.166e-2(3.653e-3)‡	1.412e-1(6.716e-3)‡	1.335e-1(4.281e-3)‡	**1.568e-2(4.929e-7)**
	(20,20)	1.982e-2(6.715e-4)‡	1.946e-2(8.100e-4)‡	3.850e-2(3.225e-3)‡	3.708e-2(1.835e-3)‡	**5.196e-3(7.427e-8)**
	(20,5)	2.061e-1(1.040e-2)‡	3.812e-1(1.307e-2)‡	**5.832e-3(5.743e-4)**	6.051e-3(3.467e-4)	6.419e-2(2.428e-5)
FDA4	(20,10)	1.318e-1(8.171e-3)‡	1.413e-1(1.343e-2)‡	**5.684e-3(2.024e-4)**	6.210e-3(5.641e-4)	6.499e-2(1.781e-5)
	(20,20)	8.246e-2(8.924e-3)‡	6.187e-2(2.721e-3)‡	**5.972e-3(4.348e-4)**	6.028e-3(3.027e-4)	5.927e-2(8.716e-6)
	(20,5)	2.935e-1(1.912e-2)‡	2.957e-1(1.863e-2)‡	3.634e-1(2.855e-4)‡	3.637e-1(4.655e-4)‡	**1.188e-1(5.461e-6)**
FDA5	(20,10)	2.815e-1(1.226e-2)‡	3.077e-1(3.944e-2)‡	3.761e-1(8.102e-4)‡	3.766e-1(6.396e-4)‡	**1.174e-1(3.963e-6)**
	(20,20)	3.567e-1(1.213e-2)‡	3.731e-1(1.051e-2)‡	4.269e-1(7.021e-4)‡	4.289e-1(5.694e-4)‡	**1.160e-1(1.904e-6)**
	(20,5)	1.970e-2(4.927e-4)‡	1.274e-2(1.153e-3)‡	8.988e-2(9.641e-4)‡	1.182e-2(3.875e-4)‡	**2.625e-3(4.971e-0)**
dMOP1	(20,10)	1.007e-2(4.550e-3)‡	1.103e-2(6.693e-3)‡	4.922e-3(6.681e-4)‡	7.413e-3(1.277e-3)‡	**2.355e-3(3.060e-7)**
	(20,20)	1.200e-2(4.672e-3)‡	1.170e-2(6.166e-3)‡	4.635e-3(8.664e-4)‡	5.827e-3(7.978e-4)‡	**1.595e-3(2.710e-7)**
	(20,5)	5.665e-3(6.934e-4)‡	5.186e-3(5.709e-4)‡	1.228e-2(1.577e-3)‡	2.113e-3(2.268e-4)‡	**3.085e-3(7.867e-0)**
dMOP2	(20,10)	1.790e-2(8.053e-4)‡	1.676e-2(9.179e-4)‡	9.492e-3(1.310e-3)‡	8.853e-3(9.737e-4)‡	**2.209e-3(1.542e-8)**
	(20,20)	9.390e-3(2.697e-4)‡	9.046e-3(5.090e-4)‡	6.555e-3(8.507e-4)‡	6.334e-3(8.268e-4)‡	**1.609e-3(1.563e-8)**
	(20,5)	1.424e+2(2.548e+0)‡	1.279e+2(2.666e+0)‡	3.850e-2(4.938e-3)‡	4.107e-2(5.155e-3)‡	**4.713e-3(3.535e-7)**
dMOP2	(20,10)	1.790e-2(8.053e-4)‡	1.676e-2(9.179e-4)‡	9.492e-3(1.310e-3)‡	8.853e-3(9.737e-4)‡	**2.209e-3(1.542e-8)**
	(20,20)	1.186e+2(3.736e+0)‡	9.110e+1(7.638e+0)‡	2.160e-2(5.095e-3)‡	2.474e-2(2.561e-3)‡	**2.594e-3(1.784e-7)**

This suggests that the effect of all algorithms' performance gradually decreases when n_t becomes from large to small. It is worth noting that the performance of DACP significantly outperforms other algorithms on the majority of the s-elected DMOPs under the different n_t. I.e., for most of DMOPs, the metric values obtained by DACP in Table 5 when n_t increases from 5 to 20 usually are relatively smaller than other algorithms. This conclusion demonstrates that the performance of DACP is stably good whatever the setting of n_t on searching for the approximation of ROI on the majority of DMOPs. For FDA4, Table 5 obviously shown that the performance of NUMS-MOEA/D-A is best on the MGD metric, and DACP is worse than NUMS-MOEA/D-A and NUMS-MOEA/D-B. However, DACP is better g-NSGA-II-A and g-NSGA-II-B with the increase of n_t, i.e., MGD metric values obtained by DACP are smallest than g-NSGA-II-A and g-NSGA-II-B. Consequently, our algorithm needs to be improved future to address FDA4 well. Overall, although the performance of algorithms is very sensitive to the setting of the severity change, DACP still is very competitive for searching the approximation of ROI on the majority of DMOPs, which is significant on the Wilcoxon rank-sum test.

5.3. The tracked ability of different algorithms

Apart from the tabular presentation, we also plot their obtained POFs of FDA1, dMOP1-dMOP3 over 21-time windows, which are shown in Fig. 5. Here, the change frequency (i.e., τ_t) was fixed as 10, and the severity of change (i.e., n_t) was set as 10. The other parameters were the same as Section 4.4.

For FDA1 in Fig. 5(a), it is obviously shown that the tracked ability of DACP is the best than other algorithms, followed by the NUMS-MOEA/D-A and NUMS-MOEA/D-B. As shown in Fig. 5(b), although we can see that the convergence and distribution obtained by DACP, g-NSGA-II-A, and g-NSGA-II-B outperform the NUMS-MOEA/D-A and NUMS-MOEA/D-B, the searching range of g-NSGA-II-A and g-NSGA-II-B is still big since the defined of g-dominance [49, 27] on the part of time windows. The convergence and distribution of DACP are the best on all time windows for dMOP2 and dMOP3 in Fig. 5(c) and Fig. 5(d), compared with other algorithms. Although the solutions obtained by NUMS-MOEA/D-A and NUMS-MOEA/D-B locate in the preference region, they have not approximated the ROI on all windows for dMOP1-dMOP3. It is very difficult for g-NSGA-II-A and g-NSGA-II-B to search for the approximation of ROI. Because the switch of the position-related variable of dMOP3 can cause severe diversity loss to the population and the selection of g-dominance is insufficient [24, 39]. All in all, the ability to track the POS and searching for the ROI of DACP between five algorithms is the best.



(a) FDA1 (b) dMOP1 (c) dMOP2 (d) dMOP3

Figure 5: Obtained POFs for five problems with $n_t = 10$ and $\tau_t = 10$.

5.4. Analysis of MHV

In the experimental results before this section, we maintain the original preference information and size of each compared algorithm. In other words,

Table 6: Mean and standard deviation values of MHV obtained by five algorithms.

Fun.	(τ_t, n_t)	g-NSGA-II-A	g-NSGA-II-B	NUMS-MOEA/D-A	NUMS-MOEA/D-B	DACP
FDA1	$(10, 10)$	1.555e-3(1.366e-1)‡	4.033e-3(4.033e-2)‡	1.269e-3(1.181e-2)‡	1.719e-3(6.719e-3)‡	**5.261e-3(5.132e-3)**
FDA2	$(10, 10)$	3.878e-3(3.730e-2)‡	3.387e-3(3.387e-3)‡	**5.422e-3(8.150e-3)**	5.210e-3(5.210e-3)	4.997e-3(4.908e-3)
FDA3	$(10, 10)$	5.033e-2(4.769e-2)‡	2.477e-2(2.477e-2)‡	1.609e-2(1.430e-2)‡	1.627e-2(1.627e-2)‡	**9.156e-2(1.135e-2)**
FDA4	$(10, 10)$	5.974e-1(5.752e-1)‡	6.664e-1(6.664e-1)‡	4.240e-3(4.060e-3)‡	1.085e-2(1.085e-2)‡	**4.936e+0(5.064e-30)**
FDA5	$(10, 10)$	5.106e-1(4.950e-1)‡	7.265e-1(7.265e-1)‡	3.631e-2(3.684e-2)‡	3.302e-2(3.302e-2)‡	**2.177e+0(2.150e-3)**

maintaining the original preference information and the size of compared algorithms is not to change their own paper advantages that have been discussed detailedly. Therefore, the previous parts of Section 5 focus on discussing the algorithm's performance on the MGD.

Afterward, this subsection analyses the MHV's results obtained by all compared algorithms in Table 6. Although changing the compared algorithms' settings is unfair for the paper itself, the preference information is set at the same when analyzing the MHV on FDA1-FDA5. The other parameter settings (except for the show in Table 6) have been described in Section 4.4 in detail. As in Table 6, the MHV's results of DACP are the best on most problems, which means that the DACP's performance outperforms the other algorithms.

6. Conclusion and Future Directions

This paper presents a novel DMO algorithm based decomposition and changing preference, which considers the ROI that may be varying with change with time. Suppose that the ROI is defined by the preference point and the neighborhood size provided by DM in the initial stage of an environment and within the neighborhood size is fixed in the whole optimal process. Moving the well-distribution reference points which locate in the neighborhood range to around the preference point effectively leads to the evolution of the population in a limited time interval. If a change is detected, the proposed strategy is performed to respond to change. This strategy is assisted by a promising direction obtained by letting a few solutions evolve independently for a short time. Consequently, the proposed algorithm can effectively track the corresponding POF and quickly search for approximating the corresponding ROI in a limited time interval, which has been examined by comprehensive experiments.

Although DACP is very effective on the majority of selected DMOPs, it still needs to be improved for handling FDA4 and DF3 well. Additionally, Figure 5 shown that the population obtained by DACP has the best distribution than the other algorithms. This suggests that DACP not only can simultaneously balance multi-objective optimization and dynamic optimization but also quickly recognize the ROI change in dynamic environments. Consequently, the comprehensive experiments (i.e., Section 5) significantly exhibit that DACP is the best than the other algorithms.

In our future work, we will explore more technologies, to effectively search for the approximation of the corresponding ROI with time-varying, to satisfy the DM's requirement in dynamic environments for DMOPs with different challenges. Meanwhile, detecting the severity of change and incorporating it with the DMOEAs is another attempt, which explores its availability under dynamic environments.

Acknowledgement

This work was supported by the research projects: the National Natural Science Foundation of China under Grant Nos. 61772178, 61876164, 61673331, Xiangtan university graduate research and innovation project under Grant No. XDCX2019B057.

References

[1] R. Rambabu, P. Vadakkepat, K. C. Tan, M. Jiang, A mixture-of-experts prediction framework for evolutionary dynamic multiobjective optimization, IEEE Transactions on Cybernetics 50 (12) (2020) 5099–5112. doi:10.1109/TCYB.2019.2909806.

[2] M. Jiang, Z. Wang, H. Hong, G. G. Yen, Knee point based imbalanced transfer learning for dynamic multi-objective optimization, IEEE Transactions on Evolutionary Computation (2020) 1–1doi:10.1109/TEVC.2020.3004027.

[3] J. Dutta, P. Barma, A. Mukherjee, S. Kar, T. De, A multi-objective open set orienteering problem, Neural Computing and Applications 32. doi:10.1007/s00521-020-04798-7.

[4] S. Majumder, S. Kar, T. Pal, Rough fuzzy quadratic minimum spanning tree problem, Expert Systems 36. doi:10.1111/exsy.12364.

[5] G. Ismayilov, H. R. Topcuoglu, Dynamic multi-objective workflow scheduling for cloud computing based on evolutionary algorithms, in: 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion), 2018, pp. 103–108. doi:10.1109/UCC-Companion.2018.00042.

[6] M. Orouskhani, D. Shi, X. Cheng, A fuzzy adaptive dynamic nsga-ii with fuzzy-based borda ranking method and its application to multimedia data analysis, IEEE Transactions on Fuzzy Systems (2020) 1–1doi:10.1109/TFUZZ.2020.2979119.

[7] S. Majumder, S. Kar, Multi-criteria shortest path for rough graph, Journal of Ambient Intelligence and Humanized Computing 9 (2018) 1–25. doi:10.1007/s12652-017-0601-6.

[8] M. Pandey, R. Datta, R. Dey, B. Bhattacharya, Multi-objective optimisation of dynamic responses for a rail freight wagon using regression models, in: 2019 IEEE 18th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC), 2019, pp. 361–365. doi:10.1109/ICCICC46617.2019.9146044.

[9] Y. N. Guo, J. Cheng, S. Luo, D. Gong, Y. Xue, Robust dynamic multi-objective vehicle routing optimization method, IEEE/ACM Transactions

on Computational Biology and Bioinformatics 15 (6) (2018) 1891–1903. doi:10.1109/TCBB.2017.2685320.

[10] M. M. Hasan, K. Lwin, A. Shabut, M. A. Hossain, Design and development of a benchmark for dynamic multi-objective optimisation problem in the context of deep reinforcement learning, in: 2019 22nd International Conference on Computer and Information Technology (ICCIT), 2019, pp. 1–6. doi:10.1109/ICCIT48885.2019.9038529.

[11] S. B. Gee, K. C. Tan, H. A. Abbass, A benchmark test suite for dynamic evolutionary multiobjective optimization, IEEE Transactions on Cybernetics 47 (2) (2017) 461–472. doi:10.1109/TCYB.2016.2519450.

[12] R. Chen, K. Li, X. Yao, Dynamic multiobjectives optimization with a changing number of objectives, IEEE Transactions on Evolutionary Computation 22 (1) (2018) 157–171. doi:10.1109/TEVC.2017.2669638.

[13] Y. Liu, Y. Chen, G. Yang, Developing multiobjective equilibrium optimization method for sustainable uncertain supply chain planning problems, IEEE Transactions on Fuzzy Systems 27 (5) (2019) 1037–1051. doi:10.1109/TFUZZ.2018.2851508.

[14] Z. Jiang, Z. Fan, W. H. Ip, X. Chen, Fuzzy multiobjective modeling and optimization for one-shot multiattribute exchanges with indivisible demand, IEEE Transactions on Fuzzy Systems 24 (3) (2016) 708–723. doi:10.1109/TFUZZ.2015.2476516.

[15] L. Cao, L. Xu, E. D. Goodman, C. Bao, S. Zhu, Evolutionary dynamic multiobjective optimization assisted by a support vector regression predictor, IEEE Transactions on Evolutionary Computation 24 (2) (2020) 305–319. doi:10.1109/TEVC.2019.2925722.

[16] A. Muruganantham, K. C. Tan, P. Vadakkepat, Evolutionary dynamic multiobjective optimization via kalman filter prediction, IEEE Transactions on Cybernetics 46 (12) (2016) 2862–2873. doi:10.1109/TCYB.2015.2490738.

[17] K. Nag, T. Pal, R. K. Mudi, N. R. Pal, Robust multiobjective optimization with robust consensus, IEEE Transactions on Fuzzy Systems 26 (6) (2018) 3743–3754. doi:10.1109/TFUZZ.2018.2848261.

[18] Y. Chi, J. Liu, Learning of fuzzy cognitive maps with varying densities using a multiobjective evolutionary algorithm, IEEE Transactions on Fuzzy Systems 24 (1) (2016) 71–81. doi:10.1109/TFUZZ.2015.2426314.

[19] Q. Zhang, S. Yang, S. Jiang, R. Wang, X. Li, Novel prediction strategies for dynamic multiobjective optimization, IEEE Transactions on Evolutionary Computation 24 (2) (2020) 260–274. doi:10.1109/TEVC.2019.2922834.

[20] F. Zou, G. Yen, L. Tang, A knee-guided prediction approach for dynamic multi-objective optimization, Information Sciences 509. doi:10.1016/j.ins.2019.09.016.

[21] A. Nebro, A. B. Ruiz, C. Barba-Gonzlez, J. Garca-Nieto, M. Luque, J. Aldana Montes, Indm2: Interactive dynamic multi-objective decision making using evolutionary algorithms, Swarm and Evolutionary Computation 40. doi:10.1016/j.swevo.2018.02.004.

[22] K. Li, R. Chen, D. Savi, X. Yao, Interactive decomposition multiobjective optimization via progressively learned value functions, IEEE Transactions on Fuzzy Systems 27 (5) (2019) 849–860. doi:10.1109/TFUZZ.2018.2880700.

[23] M. Helbig, K. Deb, A. Engelbrecht, Key challenges and future directions of dynamic multi-objective optimisation, in: 2016 IEEE Congress on Evolutionary Computation (CEC), 2016, pp. 1256–1261. doi:10.1109/CEC.2016.7743931.

[24] J. Molina, L. Santana-Quintero, A. Hernndez-Daz, C. Coello, R. Caballero, G-dominance: Reference point based dominance for multiobjective metaheuristics, European Journal of Operational Research 197 (2009) 685–692. doi:10.1016/j.ejor.2008.07.015.

[25] L. Ben Said, S. Bechikh, K. Ghedira, The r-dominance: A new dominance relation for interactive evolutionary multicriteria decision making, IEEE Transactions on Evolutionary Computation 14 (5) (2010) 801–818. doi:10.1109/TEVC.2010.2041060.

[26] Y. N. Guo, X. Zhang, D. W. Gong, Z. Zhang, J. J. Yang, Novel interactive preference-based multiobjective evolutionary optimization for bolt supporting networks, IEEE Transactions on Evolutionary Computation 24 (4) (2020) 750–764. doi:10.1109/TEVC.2019.2951217.

[27] J. Yi, J. Bai, H. He, J. Peng, D. Tang, ar-moea: A novel preference-based dominance relation for evolutionary multiobjective optimization, IEEE Transactions on Evolutionary Computation 23 (5) (2019) 788–802. doi:10.1109/TEVC.2018.2884133.

[28] S. Jiang, S. Yang, A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization, IEEE Transactions on Evolutionary Computation 21 (1) (2017) 65–82. doi:10.1109/TEVC.2016.2574621.

[29] H. Li, Q. Zhang, Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii, IEEE Transactions on Evolutionary Computation 13 (2) (2009) 284–302. doi:10.1109/TEVC.2008.925798.

[30] M. Rong, D. Gong, Y. Zhang, Y. Jin, W. Pedrycz, Multidirectional prediction approach for dynamic multiobjective optimization problems, IEEE Transactions on Cybernetics 49 (9) (2019) 3362–3374. doi:10.1109/TCYB.2018.2842158.

[31] S. Majumder, M. Kar, S. Kar, T. Pal, Uncertain programming models for multi-objective shortest path problem with uncertain parameters, Soft Computing 24. doi:10.1007/s00500-019-04423-3.

[32] M. Jiang, Z. Wang, L. Qiu, S. Guo, X. Gao, K. C. Tan, A fast dynamic evolutionary multiobjective algorithm via mani-

fold transfer learning, IEEE Transactions on Cybernetics (2020) 1–12doi:10.1109/TCYB.2020.2989465.

[33] G. Ruan, G. Yu, J. Zheng, J. Zou, The effect of diversity maintenance on prediction in dynamic multi-objective optimization, Applied Soft Computing 58 (2017) 631–647. doi:10.1016/j.asoc.2017.05.008.

[34] Z. Liang, S. Zheng, Z. Zhu, Hybrid of memory and prediction strategies for dynamic multiobjective optimization, Information Sciences 485. doi:10.1016/j.ins.2019.01.066.

[35] D. Gong, B. Xu, Y. Zhang, Y. Guo, S. Yang, A similarity-based cooperative co-evolutionary algorithm for dynamic interval multiobjective optimization problems, IEEE Transactions on Evolutionary Computation 24 (1) (2020) 142–156. doi:10.1109/TEVC.2019.2912204.

[36] K. Deb, U. N, K. Sindhya, Dynamic multi-objective optimization and decision-making using modified nsga-ii: A case study on hydro-thermal power scheduling, 2007, pp. 803–817. doi:10.1007/978-3-540-70928-2_60.

[37] C. Goh, K. C. Tan, A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization, IEEE Transactions on Evolutionary Computation 13 (1) (2009) 103–127. doi:10.1109/TEVC.2008.920671.

[38] S. B. Gee, K. C. Tan, C. Alippi, Solving multiobjective optimization problems in unknown dynamic environments: An inverse modeling approach, IEEE Transactions on Cybernetics 47 (12) (2017) 4223–4234. doi:10.1109/TCYB.2016.2602561.

[39] A. Zhou, Y. Jin, Q. Zhang, A population prediction strategy for evolutionary dynamic multiobjective optimization, IEEE Transactions on Cybernetics 44 (1) (2014) 40–53. doi:10.1109/TCYB.2013.2245892.

[40] R. Liu, J. Li, J. fan, C. Mu, L. Jiao, A coevolutionary technique based on multi-swarm particle swarm optimization for dynamic multi-

objective optimization, European Journal of Operational Research 261. doi:10.1016/j.ejor.2017.03.048.

[41] S. Wang, S. Ali, T. Yue, M. Liaaen, Integrating weight assignment strategies with nsga-ii for supporting user preference multiobjective optimization, IEEE Transactions on Evolutionary Computation 22 (3) (2018) 378–393. doi:10.1109/TEVC.2017.2778560.

[42] L. Li, H. Chen, J. Li, N. Jing, M. Emmerich, Integrating region preferences in multiobjective evolutionary algorithms based on decomposition, in: 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI), 2018, pp. 379–384. doi:10.1109/ICACI.2018.8377488.

[43] Q. Zhang, H. Li, Moea/d: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731. doi:10.1109/TEVC.2007.892759.

[44] J. Ou, J. Zheng, G. Ruan, Y. Hu, J. Zou, M. Li, X. Tan, A pareto-based evolutionary algorithm using decomposition and truncation for dynamic multi-objective optimization, Applied Soft Computing 85 (2019) 105673. doi:10.1016/j.asoc.2019.105673.

[45] M. Farina, K. Deb, P. Amato, Dynamic multiobjective optimization problems: test cases, approximations, and applications, IEEE Transactions on Evolutionary Computation 8 (5) (2004) 425–442. doi:10.1109/TEVC.2004.831456.

[46] S. Jiang, S. Yang, X. Yao, K. C. Tan, M. Kaiser, N. Krasnogor, Benchmark functions for the cec'2018 competition on dynamic multiobjective optimization, Tech. rep., Newcastle University (2018). doi:dora.dmu.ac.uk/handle/2086/15156.

[47] J. Ding, C. Yang, Q. Xiao, T. Chai, Y. Jin, Dynamic evolutionary multiobjective optimization for raw ore allocation in mineral processing, IEEE

Transactions on Emerging Topics in Computational Intelligence 3 (1) (2019) 36–48. doi:10.1109/TETCI.2018.2812897.

[48] J. Wang, W. Ren, Z. Zhang, H. Huang, Y. Zhou, A hybrid multiobjective memetic algorithm for multiobjective periodic vehicle routing problem with time windows, IEEE Transactions on Systems, Man, and Cybernetics: Systems 50 (11) (2020) 4732–4745. doi:10.1109/TSMC.2018.2861879.

[49] K. Li, R. Chen, G. Min, X. Yao, Integration of preferences in decomposition multiobjective optimization, IEEE Transactions on Cybernetics 48 (12) (2018) 3359–3370. doi:10.1109/TCYB.2018.2859363.

[50] K. Li, Q. Zhang, S. Kwong, M. Li, R. Wang, Stable matching-based selection in evolutionary multiobjective optimization, IEEE Transactions on Evolutionary Computation 18 (6) (2014) 909–923. doi:10.1109/TEVC.2013.2293776.