A Novel Scalable Framework for Constructing Dynamic Multi-objective Optimization Problems

Qingshan Tan$^{1,2}$, Changhe Li$^{1,2}$, Hai Xia$^{1,2}$, Sanyou Zeng$^{3}$, Shengxiang Yang$^4$

$^1$School of Automation, China University of Geosciences, Wuhan, 430074, China
$^2$Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems
$^3$School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan, 430074, China
$^4$School of Computer Science and Informatics, De Montfort University, Leicester, LE1 9BH, United Kingdom

Email: qingshan.t@cug.edu.cn, changhe lw@gmail.com (Corresponding author: Changhe Li), strawberry9583@gmail.com
Email: sanyouzeng@gmail.com
Email: sanyouzeng@gmail.com

Abstract—Modeling dynamic multi-objective optimization problems (DMOPs) has been one of the most challenging tasks in the field of dynamic evolutionary optimization. Based on the analysis of the existing DMOPs, several features widely existed in real-world applications are not taken into account: different objectives may have different function models and variables to be optimized; and the number of conflicting variables should be independent from the number of objectives; the time-linkage property is not considered. In order to overcome the above issues, a novel framework for constructing DMOPs is proposed, where all objectives can be designed independently, and the number of the conflicting variables can be tuned by users. Moreover, it is easy to add new dynamic features to this framework. Several classical dynamic multi-objective optimization algorithms are tested on four scenarios, results show that these characteristics are challenging for the existing algorithms.

Index Terms—Dynamic multi-objective optimization, Benchmark design, Time-linkage.

I. INTRODUCTION

Dynamic multi-objective optimization problems (DMOPs) have their own difficulties against dynamic optimization problems (DOPs) and multi-objective optimization problems (MOPs). Once more than one objective changes over time, only limited resources could be used to locate and track new optima, and historical information will be outdated which results in lacking of heuristic information. Furthermore, the Pareto optimal front (PF) and/or the Pareto optimal set (PS) will change with time, which requires the algorithms to have a good ability to adapt to the abundant shapes of the PF. Algorithms must keep a good diversity and convergence at the same time to overcome these challenges.

There have been some classical DMOPs proposed to simulated the features of the multi-objective optimization in dynamic environments, such as FDA [1], dMOPs [2], HE [3], JY [4], GTA [5], SDP [6] and so on. But they have some common flaws: objectives have common factors, similar mathematical models and the number of conflicting variables are related to the number of objectives which may be not true in real world. And also, they lack of dynamic features abstracted from reality life, for example, the time-linkage features [7], [8], which is common in game theory and strategy decision fields but rarely exists in the current benchmarks. Although so many dynamic multi-objective optimization algorithms (DMOAs) have been tested on the above test suites and achieved good results, they still should be gone through more scenarios especially some complex dynamic characteristics from the real-world application point of view. Therefore, designing a benchmark which includes more structure features and dynamic characteristics is a meaningful work.

This paper will introduce a general framework to design dynamic multi-objective benchmark. The motivation is to make objectives independent and the conflicting variables and conflicting regions between all objectives all could be designed by users. To achieve these, an object-oriented constructing methods is proposed, which makes it easy to control the shape of objectives and conflicting regions. Furthermore, the time-linkage change mode is also added to the proposed framework. With these novel features, the benchmark could cover more optimization models in different domains.

In the rest of the paper, methods for constructing classical DMOPs and some existing benchmarks are present in section II. Section III introduces the proposed framework and design details. Experiments are performed to test the proposed framework in section IV. Finally, section V draws conclusions and gives future works.

II. RELATED WORKS

This section describes several common ways to construct DMOPs. And then, the characteristics of the existing benchmarks are also introduced.

A. Traditional Construction Methods for DMOPs

Existing construction methods for DMOPs are mainly derived from MOPs, and then a dynamic factor is added to make the problems change. There are four major methods to construct DMOPs, which are introduced bellow.

1) Constructing by Composite Functions: It was first proposed by Deb [9] and is also called “from bottom to top” construction method. A typical feature is that the PF and PS...
are designed independently. Afterwards, Simon et al. [10] used a distance function and a position function to describe the PS and PF. A general mathematical formulation is shown as bellow:

\[
\begin{align*}
\min F_1 &= (1 + g(x_1, t)) \times k_1(x_{11}, t) \\
... & \\
\min F_{m-1} &= (1 + g(x_1, t)) \times k_{m-1}(x_{m-1}, t) \\
\min F_m &= (1 + g(x_1, t)) \times h(k_1, ..., k_{m-1}, g(t))
\end{align*}
\]

where \( m \) is the number of objectives; \( g(x_1, t) \) is the distance function which denotes the degree of convergence; and \( g(x_1, t) = 0 \) is the necessary condition if a solution belongs to the PS; \( h(k_1, ..., k_{m-1}, g(t)) \) is the position function which define the relationships among all objectives when the former \( m - 1 \) objectives get their own optima. These two functions use two groups of variables for the convenience of designing the PF and PS.

2) Constructing by Multi-objective Combination: This method works on combining existing MOPs with dynamic weights. It is inspired by using a dynamic weight sum method when they designed a many-objective optimization problem and then a dynamic bio-objective optimization method works on combining existing MOPs with dynamic weights. It is inspired by using a dynamic weight sum method when they designed a many-objective optimization problem. The combination process is shown in Eq. (2). It is understandable that original problem is a triple-objective problem and then a dynamic bio-objective optimization problem is generated by combining objectives with the dynamic weights.

\[
\begin{align*}
\min \{ F_1, F_2 \} \\
F_1 &= \omega(t) \cdot f_1 + (1 - \omega(t)) \cdot f_2 \\
F_2 &= \omega(t) \cdot f_1 + (1 - \omega(t)) \cdot f_3
\end{align*}
\]

3) Constructing by Subspace Combination: This way partitions the search space into many non-overlapping subspaces, and then design objective functions in each subspace respectively. It has been realized in [12], [13], and they all focused on dynamic single-objective optimization problems. Li et al. [14] designed a framework where constructing dynamic multi-objective test suites based on the method of decomposing search space by the k-dimensional tree [15]. Each objective in each subspace is an unimodal function. The dynamic features could be realized by adjusting the height, width and the location of the peaks. This method has a prominent advantage that objectives have their own features. It is more suitable for real-world optimization models.

4) Constructing by Minimum Distance Function: Ishibuchi et al. proposed a minimum distance function construction method when they designed an existing many-objective optimization problem [16]. It supposes that there are \( m \) anchor points \( P_1, P_2, ..., P_m \) in a \( n \)-dimensional search space, and any points in the search space have a Euclidean distance with these \( m \) points. The \( i \)th objective \( F_i \) is the value of Euclidean distance between the search point and \( P_i \). Heiner et al. [17] extended it into a DMOP by setting the number of the anchor points, number of variables, position of the anchor points dynamically and so on. It is very easy to adjust the dynamics, but its real PF shape or formulation is not known. In fact, it is similar to a type of problem whose objectives are formed by an unimodal function, and it could be regarded as a sub-problem in the third construction method.

B. Classical DMOPs

Based on the above constructing methods, several classical DMOPs have been created and used widely. This subsection introduces them briefly and summarizes some important features which a dynamic multi- or many-objective optimization benchmark should have.

FDA [1] is one of the widely used benchmarks, and its prototype is based on ZDT [18], which is a classical MOP in static environments. The prominent advantage is that the PF and PS could be analyzed easily. In terms of dynamic characteristics, the shape or location of the PF and PS are controlled by a trigonometric function.

dMOPs [2] is very similar to the FDA, but all test functions just have two objectives. The only difference is that in dMOP3, the first objective function is set as an arbitrary variable in each new environment, which have no effects on the PF, but for the PS, the distribution at successive moments may be orthogonal which may result in big difficulties for algorithms.

ZJZ [19] is also a biobjective function and the originality is that the shape of the PS is nonlinear. Afterwards, F benchmark [20] is proposed, where the shape of the PS is also nonlinear and the location has a big jump periodically. But it is just for two or three objectives.

Hellig et al. designed the HE benchmark [3], which focuses on biobjective and emphasizes the complex shape of the PS expression. The dynamics are mainly focus on the shape of PF, but the PS will not change during the whole process. In fact, dynamic problems should pay more attention to dynamic location of the optima.

Biswas et al. designed UDF [21]. It is constructed by grouping variables. The most contribution is that the PS and PF could shift or rotate over time, even the position of PS has some random factors.

Gee et al. thought that the attention of the benchmarks should be paid to dynamic characteristics rather than the difficulties of the multi-objective optimization problems. Therefore, they designed GTA [5] and its features are dynamic modality of the objectives, dynamic degradation and continuity of the PF.

Jiang and Yang have proposed a benchmark named JY [4]. Its advantages are that the shape of the PF could change from concave to convex, each variable of the PS has different change rate and the change type may be random in the test case. Recently, they designed a new extensible benchmark named SDP [6], where all test functions could be extensible in terms of the number of variables and objectives. Furthermore, partial detectability and predictability are introduced into the benchmarks. Some other features like time-varying objective bias, dynamic range of the PF are also included.

Based on the above summarization, it can be concluded that:

- Composite functions are the most popular to design DMOPs because it is easy to get a mathematical for-
mulation of the PS and PF. However, it can not avoid similarity between objectives.

- The shape of the PF is simple in most existing benchmarks except the recent SDP. When the number of objectives increases, the PF always displays as a hyperplane or a part of surface of a hypersphere.
- Existing benchmarks have paid much attention to the changes of the PF. In fact, the search difficulties are mainly from the structure of the search space rather than from the objective space. The shape of the PF just imply the relationship between objectives on optimal sets.
- Although these methods all group variables, the whole conflicting variable spaces may be potential solutions of the PS, which is rare in real world.
- Dynamic characteristics are limited, especially the dynamic change modes of the PS, which could represent some real-world changes.

In order to expand the ability of the DMOP, constructing methods should be more general and flexible. One of the valid methods is to decouple all objectives and to design specific regions for conflicts between objectives. A construction framework will be introduced in the next section.

III. PROPOSED FRAMEWORK

The framework is inspired by the idea of object-oriented programming. For a MOP, each objective can be regarded as an object, the landscape of an objective is comprised of a number of peaks. Thus, each peak also can be an object. Another important work is designing the PS. It can be extracted as a region with different shapes and also can be treated as an object. Therefore, any MOPs designed by this framework can be depicted as a structure shown in Fig. 1: all the objectives are made up of peaks and all peaks and the PS regions are objects. Details will be introduced as follows.

A. Grouping Variables

In the proposed framework, the search space is divided into two parts. A simple illustration is shown in Fig. 2, where m objectives need to be optimized, and each objective is comprised of n1 private variables and n2 public variables. For example, from x0 to x_{n1-1} are just optimized by the first objective, from x_{mn1} to x_{mn1+n2-1} belong to the public space which means all objectives should optimize them. The number of public variables could be adjusted freely which is different from the way used in [3], [20], [21]. The dimension of the whole search space is mn1 + n2.

\[ x_0 \ldots x_{n1-1} \ldots x_{mn1} \ldots x_{mn1+n2-1} \]

Fig. 2. Grouping variables for m objectives

B. Designing Objectives

After grouping variables, since these two spaces are independent and orthometric, the landscape is a simple extension between spaces. Therefore, each objective could be made up of two parts with an additive formulation as bellow:

\[
F_i(x_{\text{pri}}, x_{\text{pub}}, t) = 1 - \frac{f_{\text{pri}}^i(x_{\text{pri}}, t) + f_{\text{pub}}^i(x_{\text{pub}}, t)}{\max(f_{\text{pri}}^i) + \max(f_{\text{pub}}^i)}
\]

where \(F_i\) is the ith objective; \(f_{\text{pri}}^i\) denotes the part of objective function in the private space and \(f_{\text{pub}}^i\) is in the public space; \(x_{\text{pri}}^i\) is the variables belong to the ith objective in the private space; and \(x_{\text{pub}}^i\) is the variables shared by all objectives.

Then both \(f_{\text{pri}}^i\) and \(f_{\text{pub}}^i\) could be made up of the moving peak benchmark (MPB) [22] function or other existed multimodal functions [23]. Here the MPB function is adopted because it is easy to design and adjust. For example, in the private space, the MPB function is shown as

\[
f_{\text{pri}}^i(x_{\text{pri}}^i, t) = \max_{j=1,...,P_i} \frac{H_j^i(t)}{1 + \sum_{j=1,...,P_i} \|x_{\text{pri}}^i - X_j^i(t)\|_p} \]

where \(P_i\) is the number of peaks belongs to the ith objective in the private space; \(H_j^i\) is the height; \(W_j^i\) is the width and \(X_j^i\) is the location of the jth peak of the ith objective in the private space; \(p_i\) is the norm of the ith objective in the private space.

As for the public space, because the conflicting relationship between objectives will be implemented in this space, a good choice is to allocate special regions for designing the PS which will be introduced in the next subsection. The rest regions in the public space will not become the PS but could have some local attraction regions. Therefore, the formulation of \(f_{\text{pub}}^i\) will be a piecewise function as shown in Eq. (6). Similarly, the public space is also made up of peaks.

\[
f_{\text{pub}}^i(x_{\text{pub}}, t) = \begin{cases} f_{\text{PS}}^i, & \text{if } x_{\text{pub}} \in \text{PS}, \\ \max_{j=1,...,P_i} \frac{H_j^i(t)}{1 + \sum_{j=1,...,P_i} \|x_{\text{pub}} - Y_j^i(t)\|_p}, & \text{else} \end{cases}
\]

where \(\text{PS}\) is the designed PS region; \(f_{\text{PS}}^i\) denotes the value of the ith objective in the PS regions introduced in Eq. (7); \(Y_j^i\) is the location of the jth peak of the ith objective in the public space.

C. Designing the PS Regions

For MOPs, the optimal non-dominated sets always locate in the variables space where is common between all objectives. Therefore, these PS regions are abstracted into some specific
shapes such as circles, rectangles and so on. Once a solution is located in these places, it will belong to the PS. Here simple spheres or hyperspheres are used as the PS regions in the public space. Because it is easy to calculate the range of objective values in the PS.

After designing the shape of the PS regions, all objectives will have an expression in these regions. Especially for the first \( m - 1 \) objectives, they are all an unimodal function defined in Eq. (7). The height of the peak is designed elaborately in order to ensure points which are in the PS regions must dominate others outside of the PS regions when optima are obtained in the private spaces. The \( m \)th objective is a function of all the former \( m - 1 \) objectives like in [24], to ensure all points in the PS regions are non-dominated with each other.

\[
f_i^{PS} = \begin{cases} 
\frac{h_i(t)}{1 + w_i(t)||x^{pub} - z_i(t)||_{p_i}}, & i = 1, \ldots, m - 1, \\
\frac{f_{\text{base}}^{\text{pub}}(x^{\text{pub}}, t) + 16 \cdot (h(X, t) - m_{\text{off}})}, & i = m.
\end{cases}
\]  

(7)

\[
f_{\text{base}}^{\text{pub}}(x^{\text{pub}}, t) = \max(f_m^{\text{pub}}(x^{\text{pub}}, t)), x^{\text{pub}} \notin \text{PS}, \quad m_{\text{off}} = \min(h(\sqrt{m - 1}, t), 0) \quad (8)
\]

where \( h_i \) is the height; \( u_i \) is the width and \( z_i \) is the location of the peak of the \( i \)th objective in the PS regions; \( f_{\text{base}}^{\text{pub}} \) is the maximum value of the \( m \)th objective out of the PS regions and \( x^{\text{pub}} \) is the variables in the public space; \( m_{\text{off}} \) is a offset value in order to avoid the \( m \)th objective has a negative value when \( X \in [0, 1] \).

\[X = \sqrt{\sum_{i=1}^{m-1} \left( \frac{f_i^{PS}}{\max(f_i^{PS})} \right)^2} \quad (10)\]

\[k = 2 \cdot \left( \frac{X}{\beta(t)} - \left\lfloor \frac{X}{\beta(t)} \right\rfloor \right) + 1 \quad (11)\]

\[h(X, t) = 1 - \frac{\beta(t)}{2} \cdot \text{sign}(k) \cdot |k^{(t)}| - \beta(t) \cdot \left\lfloor \frac{X}{\beta(t)} - \frac{1}{2} \right\rfloor \quad (12)\]

Finally, points which are in the PS regions will be located on the PF, therefore, any points on the PF can be described as

\[
\begin{align*}
F_i &= 1 - \frac{\max(f_i^{\text{pri}})+f_i^{\text{pub}}}{\max(f_i^{\text{pri}})+\max(f_i^{\text{pub}})}, \quad i \neq m, \\
F_m &= 1 - \frac{\max(f_m^{\text{pri}})+f_m^{\text{pub}}+16 \cdot (h(X, t) - m_{\text{off}})}{\max(f_m^{\text{pri}})+\max(f_m^{\text{pub}})},
\end{align*}
\]

(13)

\[\beta(t) = 2^k \cdot |\sin(0.5t)| - 2 \quad (14)\]

**E. Mathematic Model of the Designed Framework**

Based on the above description, the designed DMOP framework can be instantiated as

\[
\begin{align*}
\min F_i &= 1 - \frac{f_i^{\text{pri}}(x_i^{\text{pri}}, t) + f_i^{\text{pub}}(x_i^{\text{pub}}, t)}{\max(f_i^{\text{pri}})+\max(f_i^{\text{pub}})}, \\
\min F_m &= 1 - \frac{f_m^{\text{pri}}(x_m^{\text{pri}}, t) + f_m^{\text{pub}}(x_m^{\text{pub}}, t)}{\max(f_m^{\text{pri}})+\max(f_m^{\text{pub}})},
\end{align*}
\]

(14)

where \( m_k \) denotes the change severity; \( m_{\text{eval}} \) is the total number of the effective evaluations and \( \text{change}_f \text{re} \) is the change frequency which is the counter of the effective evaluations; \( \alpha \) and \( \beta \) have been introduced in the Eq. (12).

With regard to dynamic changes, because all factors have been designed as objects, changing the attributions of the objects can achieve a rich combination of the dynamic characteristics. For example, it is easy to adjust the height, width, and the location of peaks. Once they change, the problem will be a DMOP. And other dynamic characteristics could be realized by changing the number of objectives, variables, the PS regions, peaks in the private and public spaces, changing the radius of the PS regions and so on. Due to space limitations, this paper does not discuss these features detailedly.

The differences from traditional methods are that all objectives could associate with any number of the private variables and the dimension of the public space is not related to the number of objectives. Furthermore, the PS regions are abstracted into independent regions which makes it possible to design local attraction regions in the public space. With these features, all objectives and the PS regions can be designed independently and it makes the framework more powerful to simulate the real-world applications than the above traditional methods.

**IV. Experimental Study**

In this section, the designed DMOP generator is used to test the performance of algorithms. Five classical DMOAs which are used widely are selected for comparison. They include DNSGA-II-A and DNSGA-II-B [25], population prediction strategy (PPS) [20], kalman filter prediction (KFP) [26] and steady-state and generational evolutionary algorithm (SGEA) [27].
A. Performance Metrics

1) Inverted Generational Distance (IGD): IGD [28] is used to measure convergence and diversity of the population. When it comes to DMOPs, the most common operation is to calculate the mean IGD (MIGD) value against all environmental changes in a run.

\[
MIGD = \left( \sum_{t=1}^{T} \frac{\sum_{s \in \mathbb{PF}_t^*} dist(s, \mathbb{PF}_t)}{|\mathbb{PF}_t^*|} \right)/T \tag{15}
\]

where \( \mathbb{PF}_t^* \) is a set of reference points on the real PF; \( \mathbb{PF}_t \) is the approximate PF at time \( t \); and \( T \) denotes the number of changes in one run.

2) Average Convergence Distance (ACD): ACD measures the average distance between the approximate PF and the location of highest peak in the private space during all runs.

\[
ACD_t = \left( \sum_{r=1}^{N} \frac{\sum_{p \in \mathbb{PF}_t} dist(p, X_{r}^{\max})}{|\mathbb{PF}_t|} \right)/N \tag{16}
\]

where \( ACD_t \) is the ACD value of the \( r \)th objective in the private space; \( \mathbb{PF}_t^* \) is the approximate PF at time \( t \) during the \( r \)th runs; \( X_{r}^{\max} \) is the location of the highest peak of the \( r \)th objective in the private space; \( N \) is the total number of runs.

3) Population Convergence Ratio (PCR): PCR measures the ratio of the approximate PF locates in the real PS regions in the public space during all runs.

\[
PCR = \left( \sum_{r=1}^{N} \frac{\sum_{p \in \mathbb{PF}_t} \left( \frac{1}{2} \text{sign}(R - dist(p, x_{\text{pub}}^{\text{PS}})) + \frac{1}{2} \right)}{|\mathbb{PF}_t|} \right)/N \tag{17}
\]

where \( x_{\text{pub}}^{\text{PS}} \) is the center of the PS region and \( R \) is its radius in the public space; \( \text{sign}(\cdot) \) is the sign function.

B. Parameter Settings

The parameters of the problems, algorithms and performance metrics are set as follows:

1) The number of the private variables for each objective and the number of the public variables were all set to 2. The number of the PS regions was 1 and the initial radius was 0.1. \( \beta \) and \( \alpha \) in the Eq. (13) was set to 0.25 and 0.2, respectively.

2) The population size was set to 100, 10% individuals were selected randomly to be evaluated before each generation to detect environmental changes.

3) In DNSGA-II-A, 20% random individuals were introduced to response changes. In DNSGA-II-B, Hypermutation probability was set to 0.5 when problems changed, and 20% individuals were replaced randomly. In PPS, autoregression order was set to 3 and the length of prediction series was set to 23, multi-objective optimizer was NSGA-II [29] rather than RM-MEDA [30]. In KFP, the order of the state vector was set to 2.

4) The number of reference points on the real PF were set to 2000 for all the tested scenarios.

5) The change severity \( n_t \) was set to 10, and change frequency \( \text{change}_t \) was set to 4000 effective evaluations. The experimental termination condition was set to undergo 100 environments and each experiment was run 30 times independently for all algorithms.

C. Experimental Results

This subsection carries on experiments to test the proposed construction method on four scenarios:

- Scenario 1: The PS changes over time and the number of peaks in the private and public space is 1.
- Scenario 2: The PS changes over time but the number of peaks in the private and public space is 2.
- Scenario 3: The PS changes by time-linkage mode and the number of peaks in the private and public space is 1.
- Scenario 4: The PS changes by time-linkage mode and the number of peaks in the private and public space is 2.

The performance metric is collected in the Table I, concrete comparison between different scenarios are described below.

1) Scenario 1: In this scenario, all objectives have only one peak in the private and public space. It is easy to converge in the private space, but in the public space, the unique peak is a trap because it is a local optima in the public space, and it is dominated by the PS regions. Fig. 3 shows the change of the ACD and PCR in the private and public space over the first 30 environmental changes for all the algorithms. Variables in the private space nearly all converge to the optima. However, variables can not converge to the PS region in most environmental changes when the radius of the PS region is small in the public space, because the designed PS region is away from the trapping peak.

2) Scenario 2: This scenario still follows the former change mode, but adds one more peak in the private and public space for all objectives in order to test the influence of the local optima. Fig. 4 shows the changes of ACD and PCR in the two spaces for all the algorithms.

After adding one more local optimum in the space for each objective, it is apparent that all the algorithms have troubles to converge to the optima in both private and public spaces. In comparison with the results in Fig. 3, Fig. 4 shows that a lot of individuals fail to find the optima in the private space for the first objective and the algorithms all have much smaller probability to converge to the PS region after adding one local peak in the public space. Fig. 5 shows the converge process of the all tested algorithms over 30 runs. All these results prove that the addition of the local attraction regions in the public space challenges all the tested algorithms.

3) Scenario 3: In this case, the location of the PS region is not related to time \( t \) but the distribution of the current population, which is an implementation of the time-linkage feature in some real-world applications in this paper. The next PS region is placed as far as possible away from the current population. It is a dynamic adversarial behavior which is common in combat games. Similarly, all objectives have only one peak in the private and public spaces.
TABLE I

MIGD METRIC VALUES, STANDARD DEVIATIONS FOR DESIGNED FOUR SCENARIOS

<table>
<thead>
<tr>
<th>test scenes</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Scenario 3</th>
<th>Scenario 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNSGA-II-A</td>
<td>0.1659(±0.0273)</td>
<td>0.2313(±0.0205)</td>
<td>0.3996(±0.0263)</td>
<td>0.2961(±0.0462)</td>
</tr>
<tr>
<td>DNSGA-II-B</td>
<td>0.1993(±0.0385)</td>
<td>0.2788(±0.0215)</td>
<td>0.4508(±0.0067)</td>
<td>0.3185(±0.0322)</td>
</tr>
<tr>
<td>PPS</td>
<td>0.0468(±0.0110)</td>
<td>0.1234(±0.0200)</td>
<td>0.4645(±0.0280)</td>
<td>0.2110(±0.0439)</td>
</tr>
<tr>
<td>KFP</td>
<td>0.0160(±0.0069)</td>
<td>0.2458(±0.0199)</td>
<td>0.4216(±0.0237)</td>
<td>0.2482(±0.0680)</td>
</tr>
<tr>
<td>SGEA</td>
<td>0.1444(±0.0229)</td>
<td>0.1789(±0.0286)</td>
<td>0.4660(±0.0231)</td>
<td>0.3184(±0.0463)</td>
</tr>
</tbody>
</table>

(a) ACD value of the 1st objective  
(b) ACD value of the 2nd objective  
(c) PCR value in the public space

Fig. 3. The changes of ACD and PCR of all the algorithms in the two spaces in scenario 1

(a) ACD value of the 1st objective  
(b) ACD value of the 2nd objective  
(c) PCR value in the public space

Fig. 4. The changes of ACD and PCR of all the algorithms in the two spaces in scenario 2

(a) DNSGA-II-A  
(b) DNSGA-II-B  
(c) PPS  
(d) KFP  
(e) SGEA

Fig. 5. The change of the IGD of all the algorithms between scenarios 1 and 2

Fig. 6 shows the changes of ACD and PCR in the two spaces for all the algorithms. In comparison with the results in scenario 1, the performance of all the algorithms gets worse in scenario 3, especially the value of PCR in the public space. Because only the PS region follows the time-linkage change mode, most algorithms still can converge in the private space, but they have smaller converge ratio in the public space. One of the reasons is that the change step is bigger under this mode. Fig. 7 shows the comparison of the convergence process and the performance difference between scenarios 1 and 3.

4) Scenario 4: When adding peaks in the private and public spaces under the time-linkage change mode, both the private spaces and the public space have more attraction regions, which makes the problem more difficult.

Fig. 8 shows the changes of ACD and PCR in the two spaces for all the algorithms. In comparison with the results in Fig. 6, Fig. 8 shows that more local optima result in more attraction regions, but it is interesting to see that when the number of local peaks gets bigger, the performance gets better in this situation. This is because when the population converges to different local PS, it may be closer to the true PS region in
V. CONCLUSIONS

This paper proposes a novel framework to construct DMOPs. It is inspired from object-oriented programming, and all objectives are made up of peaks in the private and public spaces. All objectives share the same PS regions in the public space but have their own peaks in the private space. In this framework, objectives can have their own features, the number of conflicting variables could be tuned by users. In addition, the time-linkage feature which has hardly arisen in the existing benchmarks is introduced. Finally, four scenarios are designed to test the effectiveness of the proposed framework, through the comparison results, they have shown the flexibility of the proposed framework, and created more challenges for the existing DMOAs.

In the future, it is necessary to refine the dynamic characteristics on the basis of this research, such as the number of the PS regions, the number of objectives and other original change modes which may exist in real-world applications. New DMOAs to solve the proposed test suites are also on the schedule.

VI. ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China under Grants 62076226, 61673355 and 61673331, in part by the Fundamental Research Funds for the Central Universities China University of Geosciences (Wuhan) under Grants CUG170603 and CUGGCC02, in part by the Hubei Provincial Natural Science Foundation of China under Grant 2015CFA010, in part by the 111 project under Grant B17040.

REFERENCES

multidimensional binary search trees used for associative
[2] Bentley J L. “Multidimensional binary search trees used for associa-