This paper has been published in Proceeding of International Conference on Parallel Problem Solving from Nature, 2022.

# Dynamic Multi-modal Multi-objective Optimization: A Preliminary Study

Yiming Peng, Hisao Ishibuchi[⋆]

Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation,
Department of Computer Science and Engineering,
Southern University of Science and Technology, Shenzhen 518055, China.
11510035@mail.sustech.edu.cn, hisao@sustech.edu.cn

**Abstract.** Many real-world multi-modal multi-objective optimization problems are subject to continuously changing environments, which requires the optimizer to track multiple equivalent Pareto sets in the decision space. To the best of our knowledge, this type of optimization problems has not been studied in the literature. To fill the research gap in this area, we provide a preliminary study on dynamic multi-modal multi-objective optimization. We give a formal definition of dynamic multi-modal multi-objective optimization problems and point out some key challenges in solving them. To facilitate algorithm development, we suggest a systematic approach to construct benchmark problems. Furthermore, we provide a feature-rich test suite containing 10 novel dynamic multi-modal multi-objective test problems.

**Keywords:** Evolutionary multi-objective optimization · Multi-modal multi-objective optimization · Dynamic multi-objective optimization · Benchmark problems.

## 1 Introduction

Over the past few years, multi-modal multi-objective optimization problems (MMOPs) have received increasing attention from researchers and rapidly become a popular research area. This special class of multi-objective optimization problems is characterized by having multiple equivalent Pareto sets in the decision space. As pointed out in [4], equivalent Pareto sets are useful in practical applications since they can provide extra flexibility in the decision-making procedure. Thus, in addition to ensuring a good solution distribution over the Pareto front, a multi-modal multi-objective optimization algorithm is also required to ensure the diversity in the decision space to cover as many equivalent Pareto sets as possible. Various real-world optimization problems such as the rocket engine design problems [24], the neural architecture search problems [26], and the multi-objective knapsack problems [13] can be formulated as MMOPs.

Recently, many efficient algorithms have emerged to solve MMOPs efficiently, e.g., algorithms in [7,16,18,27]. However, up to now, the algorithm research on

[⋆] Corresponding author: Hisao Ishibuchi, hisao@sustech.edu.cn.

multi-modal multi-objective optimization has tended to focus on solving MMOPs in static environments. This greatly limits the value of these algorithms in real-world applications, where the environment in which the optimization problem is posed is often dynamically changing. Due to the dynamic environment, the Pareto front and/or the Pareto set of an MMOP may change over time. For example, it is not uncommon that a rocket engine design obtained by the above-mentioned approach [24] is no longer viable due to the change of some physical constraints. In this case, instead of simply restarting the algorithm to search for a new solution, it would be more efficient to utilize the original Pareto optimal solutions (i.e., the Pareto optimal solutions obtained before the change of the environment). In this paper, we refer to this type of optimization problems as dynamic MMOPs (dMMOPs). From the previous example, we can see that dMMOPs are essentially equivalent to solving a series of MMOPs, which can be viewed as an extension of dynamic multi-objective optimization problems (dMOPs) [8]. Therefore, existing techniques for handling dMOPs are also helpful for dMMOPs.

This paper provides a preliminary study on dMMOPs. We first give a formal definition of dMMOPs and analyze some key challenges in solving them. Being a novel type of optimization problems, dMMOPs pose unprecedented challenges to algorithm designers. To facilitate algorithm development, we provide a systematic approach for constructing dMMOPs for benchmarking the algorithm performance. Furthermore, we suggest an easy-to-use test suite containing 10 novel test problems based on the proposed approach.

## 2 Related Work

### 2.1 Multi-modal Multi-objective Optimization

In Section 1, we explained that MMOPs have multiple equivalent Pareto sets in the decision space. In our previous work [21], we provided a more precise definition for MMOPs. Furthermore, we pointed out that the main challenge in solving MMOPs comes from the need for the algorithm to maintain the diversity of populations in both the decision and objective spaces. One strategy is to select solutions with good diversity in both the decision and objective spaces in environmental selection. For example, both Omni-optimizer [7] and MO_Ring_PSO_SCD [27] use modified crowding distance [5] metrics which consider the diversity in both spaces. Another popular approach is to use niching [25] mechanisms to “divide” the population into several niches, each of which evolves independently. In this manner, solutions in different niches can converge to different Pareto sets. For example, MMOEA-DC [16] partitions the population into several clusters, DNEA [17] adopts the fitness sharing strategy [10], MOEA/D-MM [20] uses the clearing strategy [22].

### 2.2 Dynamic Multi-objective Optimization

A basic dMOP can be defined by introducing a time variable $t$ into a standard multi-objective optimization problem as follows:

$$\begin{aligned} \min \boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x},t), f_2(\boldsymbol{x},t), \ldots, f_M(\boldsymbol{x},t))^T, \\ \text{s. t. } \boldsymbol{g}(\boldsymbol{x},t) \leq 0, \boldsymbol{h}(\boldsymbol{x},t) = 0, \end{aligned} \tag{1}$$

where $\boldsymbol{F}$, $\boldsymbol{g}$, and $\boldsymbol{h}$ are dynamic (i.e., time-dependent) objective functions, inequality constraints, and equality constraints, respectively.

As shown in (1), both the objective functions and the constraints may change over time, which can lead to certain Pareto optimal solutions becoming suboptimal or infeasible. Therefore, the key to efficiently solving dMOPs is to sensitively detect changes in the environment and quickly converge to the new Pareto set.

As suggested by Raquel et al. [23], dynamic multi-objective optimization evolutionary algorithms (dMOEAs) for solving dMOPs can be broadly classified into the following categories:

1. **Diversity-based dMOEAs**. This approach attempts to maintain and/or enhance the diversity of the population in order to quickly detect and react to environmental changes. In [6], Deb et al. proposed two NSGA-II [5] variants called DNSGA-II-A and DNSGA-II-B based on this approach.
2. **Memory-based dMOEAs**. This approach attempts to store (i.e., memorize) historical Pareto optimal solutions for reusing them in the future. This type of algorithms is particularly efficient for dMOPs with periodical changes. Representatives in this category include the algorithms proposed in [1,14].
3. **Prediction-based dMOEAs**. This approach aims to train a model to predict the movement of the Pareto set of a dMOP based on historical data. The main advantage of this approach is that the algorithm can react to the change proactively (i.e., it can take action in advance). This enables the algorithm to swiftly converge to the new Pareto set. However, the performance of prediction-based algorithms largely depends on the accuracy of prediction models. State-of-the-art prediction-based algorithms include MOEA/D-SVR [2] and PPS [15].
4. **Multi-population dMOEAs**. Multi-population-based algorithms explore the search space with multiple subpopulations. Ideally, multiple subpopulations can explore different regions of the search space simultaneously. In this manner, the algorithm can be more sensitive to environmental changes and locate new promising regions in the search space quickly. dCOEA [9] and VEPSO [11] are two well-known multiple-population algorithms for solving dMOPs.

## 3 Dynamic Multi-modal Multi-objective Optimization

In order to define dMMOPs, we first need to introduce the multi-modal property. Suppose that an objective function vector $\boldsymbol{F}$ defines a multi-objective optimization problem whose Pareto set is $S$, we say that $\boldsymbol{F}$ is a multi-modal function if and only if the following condition is met:

$$\exists \boldsymbol{x}_1^*, \boldsymbol{x}_2^* \in S, \text{ s. t. } \boldsymbol{x}_1^* \neq \boldsymbol{x}_2^* \text{ and } \boldsymbol{F}(\boldsymbol{x}_1^*) = \boldsymbol{F}(\boldsymbol{x}_2^*), \tag{2}$$

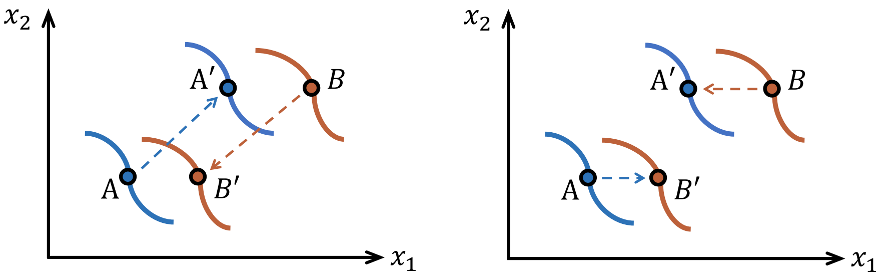
where $\boldsymbol{x}_1^*$ and $\boldsymbol{x}_2^*$ are called equivalent Pareto optimal solutions [20].

Now we can formally define dMMOPs as follows:

**Definition 1 (dMMOP).** *A dMMOP is a dMOP with multi-modal objective functions.*

Due to the multi-modal property, the optimization goal of dMMOPs is to track all the equivalent Pareto sets in the decision space. Compared to dMOPs, which require the optimizer to track only one Pareto set, dMMOPs pose more difficult challenges to algorithm designers.

First, dMMOPs require the optimizer to manage multiple "subpopulations" for tracking multiple equivalent Pareto sets which may locate in different regions in the decision space. This strategy improves the diversification ability of the optimizer at the cost of reducing its convergence ability. Thus, balancing this trade-off is essential for solving dMMOPs. Second, the time series of multiple equivalent Pareto sets may interfere with each other, making it very difficult for the optimizer to identify them correctly. Fig. 1 gives an example showing a dMMOP with two equivalent Pareto sets whose centers are denoted by $A$ and $B$, respectively. Suppose that at time $t$, $A$ and $B$ move to $A'$ and $B'$, respectively. As shown in Fig. 1 (a), the actual time series from $t-1$ to $t$ are $A \rightarrow A'$ and $B \rightarrow B'$. However, as shown in Fig. 1 (b), an optimizer (e.g., a prediction-based dMOEA) may obtain incorrect time series, i.e., $A \rightarrow B'$ and $B \rightarrow A'$. In this case, the algorithm may have a very poor performance.



(a) The actual time series. (b) The incorrect time series.

Fig. 1: Illustration of the difficulty of time series identification when handling dMMOPs.

Furthermore, detecting environmental changes when handling dMMOPs can also be a difficult task. Most existing dynamic multi-objective optimization algorithms detect changes by re-evaluating solutions. However, this strategy may fail when handling dMMOPs. For example, suppose that there is a dMMOP with three equivalent Pareto sets. If a population obtained by an algorithm is only distributed on two of the three Pareto sets, then the algorithm cannot detect the change of the remaining Pareto set by simply re-evaluating the population.

Based on the above discussions, we can see that dMMOPs are very different from standard the dMOPs. Aside from addressing the above-mentioned issues, we also need novel test problems with various characteristics to facilitate the development of efficient algorithms for solving dMMOPs. Thus, in the next section, we suggest a systematic approach for constructing benchmark dMMOPs.

## 4 A Systematic Approach for Constructing dMMOPs

In this section, we propose a general approach for constructing dMMOPs. Our proposed approach is capable of constructing scalable and flexible test problems with various dynamics.

To construct a dMMOP, we first define a basic dMOP denoted by $G$ as follows:

$$\text{Minimize}\, \boldsymbol{G}(\boldsymbol{x},t) = \{g_1(\boldsymbol{x},t), g_2(\boldsymbol{x},t), \ldots, g_M(\boldsymbol{x},t)\}, \tag{3}$$

where $M$ is the number of objectives, $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)^T$ is the decision variable vector with $p$ dimensions, and $g_j \geq 0$ $(j = 1, 2, \ldots, M)$ are $M$ objective functions to be minimized.

By carefully specifying the objective functions, we can ensure that the Pareto front of the problem $G$ changes dynamically whereas its Pareto set is stationary. The reason for this is explained later. We denote the Pareto front and Pareto set $PF(G,t)$ and $PS(G)$, respectively.

Now we construct the desired dMMOP denoted by $F$ as follows:

$$\begin{gathered}\text{Minimize}\, \boldsymbol{F}(\boldsymbol{x},\boldsymbol{y},t) = (f_1, f_2, \ldots, f_M)^T, \\ f_j = g_j(\boldsymbol{x},t)\cdot[1 + h(\boldsymbol{x},\boldsymbol{y},t)], j = 1, 2, \ldots, M,\end{gathered} \tag{4}$$

where $\boldsymbol{y} = (y_1, y_2, \ldots, y_q)^T$ is a decision variable vector with $q$ dimensions, and $h(\boldsymbol{x},\boldsymbol{y},t) \geq 0$ is a dynamic scalar function regarding two decision variable vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ which satisfy the following constraint at any time $t$:

$$\begin{gathered}\forall \boldsymbol{x}^* \in PS(G), \exists \boldsymbol{y} = \boldsymbol{y}^*, \\ \text{s. t. } h(\boldsymbol{x}^*, \boldsymbol{y}^*, t) = 0.\end{gathered} \tag{5}$$

With the above formulations, we can describe the Pareto set of the problem $F$ as follows:

$$PS(F,t) = \{\boldsymbol{x} = \boldsymbol{x}^*, \boldsymbol{y} = \boldsymbol{y}^* \mid \boldsymbol{x}^* \in PS(G), h(\boldsymbol{x}^*, \boldsymbol{y}^*, t) = 0\}. \tag{6}$$

Notice that when $h = 0$, the problem $F$ is equivalent to $G$, i.e., their Pareto fronts are the same. This means that the geometry and dynamics of the Pareto front of the problem $F$ only depend on $g_j$ $(j = 1, 2, \ldots, M)$. Recall that when we constructed $G$, we purposefully made its Pareto set stationary over time. From (6), we can see that the dynamics of the Pareto set can only be controlled by $h$.

Thus, the dynamics for the Pareto front and the Pareto set of the constructed dMMOP can be controlled independently (i.e., by specifying $g_j$ and $h$ functions, respectively). This enables researchers to construct various new test problems by composing different dynamics for the Pareto front and Pareto set. Furthermore, according to (6), $F$ has multiple equivalent Pareto sets when $h$ is a multi-modal function at time $t$ (i.e., $h(\boldsymbol{x}, \boldsymbol{y}, t) = 0$ holds for different values of $\boldsymbol{x}$ and $\boldsymbol{y}$). Thus, by altering the number and positions of the global and local optima of $h$, we can easily specify the number and distribution of the global and local Pareto sets of $F$, respectively.

To conclude, the proposed approach can construct scalable dMMOPs with an arbitrary number of decision variables and objective functions. The number of equivalent Pareto sets is also scalable. The proposed approach also allows us to set different dynamics for the Pareto front and Pareto set, thus making it possible to build flexible and sophisticated test problems according to the needs of algorithm designers.

### 4.1 Case Study on An Example Test Problem

In this section, we use a simple example with only two decision variables $x$ and $y$ to demonstrate how to create a novel dMMOP using our proposed framework. The first step is to define the base dMOP denoted by $G_e$. We use the

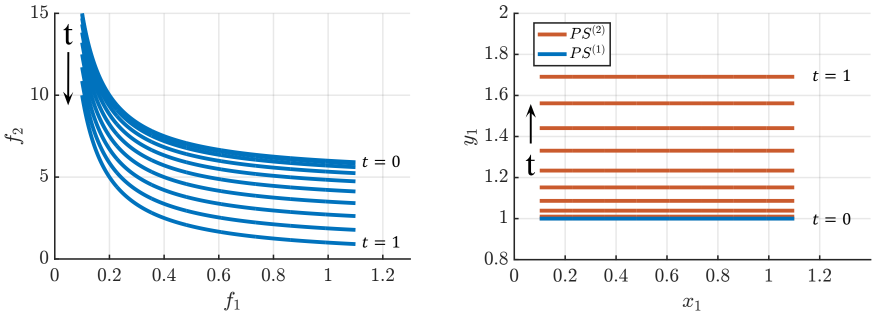
$$\begin{cases} \boldsymbol{x} = (x), x \in [0.1, 1], \\ g_1(\boldsymbol{x}, t) = x, \\ g_2(\boldsymbol{x}, t) = \dfrac{1}{x} + 5\cos^2(0.5\pi t). \end{cases} \tag{7}$$

Then we can construct a dMMOP denoted by dMMOP1 based on $G$. In our current example, we use the following function $h$ to control the dynamics and geometry of the Pareto set:

$$\begin{aligned} \boldsymbol{y} &= (y), y \in [0, 10], \\ h(\boldsymbol{x}, \boldsymbol{y}, t) &= \sqrt{|y-1| \cdot |y - D(t)|}, \\ D(t) &= 1 + 2\sin^2(0.2\pi t). \end{aligned} \tag{8}$$

Since $h$ has two optima (i.e., $y = 1$ and $y = D(t)$), dMMOP1 has two equivalent Pareto sets, one of which varies dynamically over time while the other remains stationary. Notice that when $D(t) = 1$, the two equivalent Pareto sets are overlapping. The Pareto set and Pareto front of dMMOP1 are described in (9) and illustrated in Fig. 2.

$$\begin{aligned} PS &: x \in [0.1, 1.1], y \in \{1, D(t)\}, \\ PF &: g_2 = \frac{1}{g_1} + 5\cos^2(0.5\pi t), g_1 \in [0.1, 1.1]. \end{aligned} \tag{9}$$



(a) The Pareto front.

(b) The Pareto set.

Fig. 2: The Pareto front and Pareto set of dMMOP1. (a) shows the Pareto front sampled from ten $t$ values vary from 0 to 1. (b) shows the corresponding Pareto sets, where $PS^{(1)}$ and $PS^{(2)}$ denote the first and second equivalent Pareto sets, respectively.

To the best of our knowledge, there are no test problems similar to dMMOP1 in the literature. We further investigate the performance of two algorithms, namely, DNSGA-II-A [6] and MMO-MOES [29] on this test problem. DNSGA-II-A is a diversity-based algorithm for solving dMOPs. It randomly selects and re-evaluates 10% of the population in each generation. If the objective values of any of the solutions have changed, 30% of the population are randomly re-initialized. However, since DNSGA-II-A does not take into account the multimodal property, we expect it to obtain only one of the two equivalent Pareto sets of dMMOP1. In contrast, MMO-MOES is an algorithm designed for solving MMOPs in static environments. To make it possible to handle dMMOPs, we incorporate the same change detection and change response mechanisms used in DNSGA-II-A into MMO-MOES. We call the resulting algorithm dMMO-MOES.

We use the mean values of the IGD [3] and IGDX [30] indicators (denoted by MIGD and MIGDX, respectively) to measure the performance in tracking the moving Pareto front and Pareto set, respectively. Smaller IGD and IGDX values mean that the obtained solutions can better approximate the Pareto front and the Pareto set, respectively.

The time unit $t$ for dMMOP1 can be calculated with (10), which is modified from [19].

$$t = \max\left\{\frac{1}{n_t}\lfloor 1+\frac{\tau-\tau_0}{\tau_t}\rfloor, 0\right\}, \tag{10}$$

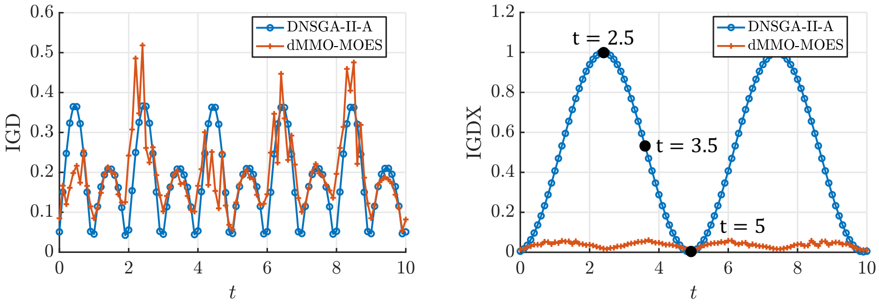
where:

- $\tau$ is the current generation counter,
- $\tau_0$ is the number of generations that the optimization problem remains stationary before the first change,
- $n_t$ is the number of distinct time steps in one time unit, which controls the severity of the dynamic change, and

- $\tau_t$ is the number of generations where $t$ remains unchanged, which controls the frequency of the dynamic change.

For each algorithm, the population size is 200, and other parameters are set as the suggested values in the corresponding papers [6,29]. For dMMOP1, $\tau_0$, $\tau_t$, and $n_t$ are specified as 50, 20, and 10, respectively. Each algorithm is tested on dMMOP1 for 31 runs with the maximum number of generations being set to $\tau_0 + 100\tau_t$ (i.e., each run comprises 100 environmental changes).

Fig. 3 and Fig. 4 report the results obtained from a single run with the median MIGD value among 31 runs of each algorithm. From Fig. 3 (a), we can observe that both DNSGA-II-A and dMMO-MOES can track the moving Pareto front. Although these two algorithms use exactly the same change detection and change response mechanisms, DNSGA-II-A clearly outperforms dMMO-MOES regarding the IGD indicator (i.e., the IGD values obtained by DNSGA-II are smaller). DNSGA-II-A not only has more stable performance but also converges faster to the new Pareto front when environmental changes occur.



(a) IGD.

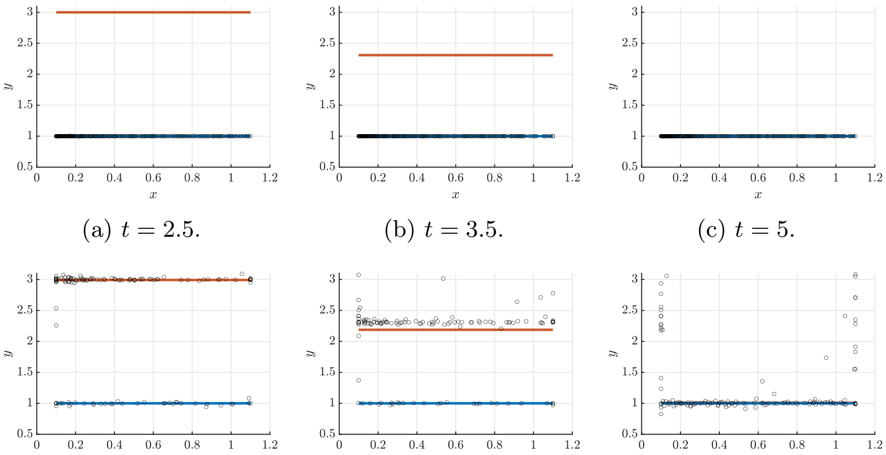
(b) IGDX.

Fig. 3: The change of IGD and IGDX values obtained by DNSGA-II-A and dMMO-MOES on dMMOP1 over 100 environmental changes.

However, in Fig. 3 (b), dMMO-MOES significantly outperforms DNSGA-II-A in terms of IGDX. This is because DNSGA-II can obtain solutions only in one of the two equivalent Pareto subsets. Therefore, as the distance between the two equivalent Pareto sets increases (e.g., $t$ increases from 0 to 2.5), the IGDX value obtained by DNSGA-II-A also increases. Similarly, when $t = 0, 5, 10$, the IGDX values obtained by DNSGA-II-A are the best since the two equivalent Pareto sets of dMMOP1 are overlapping. As shown in Fig. 3 (b), the IGDX values of dMMO-MOES are much smaller than that of DNSGA-II-A over the 100 environmental changes. These observations indicate that dMMO-MOES is more capable of tracking multiple Pareto sets than DNSGA-II-A.

Fig. 4 shows the populations obtained by DNSGA-II-A and dMMO-MOES when $t$ equals to 2.5, 3.5 and 5 in the decision space. From this figure, we can verify that DNSGA-II-A obtained solutions only in one of the two equivalent

Pareto sets, whereas dMMO-MOES can track both of them. It is worth noting that Fig. 4 (e) and Fig. 4 (f) also reveal that the convergence ability of dMMO-MOES is noticeably weaker than DNSGA-II-A since many solutions are not on the Pareto sets.



(a) $t = 2.5$. (b) $t = 3.5$. (c) $t = 5$.

(d) $t = 2.5$. (e) $t = 3.5$. (f) $t = 5$.

Fig. 4: The populations obtained by DNSGA-II-A (i.e., (a)-(c)) and dMMO-MOES (i.e., (d)-(f)) on dMMOP1 when $t$ equals to 2.5, 3.5 and 5.

From these experimental results, we can see that existing algorithms do not perform well on dMMOPs. Novel algorithms are needed in order to efficiently solve this novel type of optimization problems.

## 5 A Suggested Test Suite

In this section, we provide a novel test suite 10 dMMOPs with various dynamics. All test problems are constructed based on the proposed approach presented in Section 4. Definitions of the proposed dMMOPs are summarized in Table 1. Notice that we omit the definition of dMMOP1 in Table 1 since it has already been defined in Section 4.1.

Here we briefly describe some main characteristics of each test problem. Details of each test problem are shown in the supplementary file [1]. First, dMMOP2 features a Pareto front whose shape changes periodically from convex to linear and then to concave. It has two stationary equivalent Pareto sets in the decision space. dMMOP3 is the same as dMMOP2 except that its $h$ function is

[1] The supplementary file can be found at `https://github.com/Yiming-Peng-Official/dMMOP`

Table 1: The proposed dMMOP test suite.

| Problem | Definition | Pareto Set |
|---|---|---|
| dMMOP1 | - | - |
| dMMOP2 | $\boldsymbol{x} = (x_1, x_2)^T, x_{1:2} \in [0, 1]$,<br>$\begin{cases} g_1(\boldsymbol{x}, t) = [\cos(\pi/2x_1)\cos(\pi/2x_2)]^{1/D(t)}, \\ g_2(\boldsymbol{x}, t) = [\cos(\pi/2x_1)\sin(\pi/2x_2)]^{1/D(t)}, \\ g_3(\boldsymbol{x}, t) = [\sin(\pi/2x_1)]^{1/D(t)}, \end{cases}$<br>$\boldsymbol{y} = (y), y_1 \in [0, 1]$,<br>$h(\boldsymbol{x}, \boldsymbol{y}, t) = 1 - \sin^2(2\pi y)$,<br>$D(t) = 0.25 + 0.75\sin^2(\pi/12t)$. | $x_{1:2} \in [0, 1]$,<br>$y = \{\frac{1}{4}, \frac{3}{4}\}$. |
| dMMOP3 | Same as dMMOP2, except that $h$ is defined as follows:<br>$h(\boldsymbol{x}, \boldsymbol{y}, t) = 1 - \exp\left[-(\frac{y-0.2}{0.03})^2\right] - 0.8\exp\left[-(\frac{y-0.6}{0.4})^2\right]$. | $x_{1:2} \in [0, 1]$,<br>$y = 0.2$. |
| dMMOP4 | Add a minus sign to all objectives of dMMOP2. | Same as dMMOP2. |
| dMMOP5 | $\boldsymbol{x} = (x), x \in [0, 1]$,<br>$\begin{cases} g_1(\boldsymbol{x}, t) = x, \\ g_2(\boldsymbol{x}, t) = 1 - x\cos^2(D(t)x\pi), \end{cases}$<br>$\boldsymbol{y} = (y), y \in [-1, 1]$,<br>$h(\boldsymbol{x}, \boldsymbol{y}, t) = 1 - \exp\left[-\left(\frac{(y + x) \cdot (y - x)}{0.4}\right)^2\right]$,<br>$D(t) = 5\sin^2(0.2\pi t)$. | All non-dominated solutions satisfying:<br>$g_2 = 1 - g_1\cos^2(D(t)g_1\pi)$,<br>$g_1 \in [0, 1]$. |
| dMMOP6 | $\boldsymbol{x} = (x), x \in [0, 1]$,<br>$\begin{cases} g_1(\boldsymbol{x}, t) = x, \\ g_2(\boldsymbol{x}, t) = 1 - x - \frac{\cos(2D(t)\pi x+\pi/2)}{2D(t)\pi}, \end{cases}$<br>$\boldsymbol{y} = (y), y \in [0, 9]$,<br>$h(\boldsymbol{x}, \boldsymbol{y}, t) = \begin{cases} 2(y - \sin\lvert 2\pi x - \pi\rvert + \lvert 2\pi x - \pi\rvert)^2, & y \in [0, 4] \\ 2(y - 4 - \sin\lvert 2\pi x - \pi\rvert + \lvert 2\pi x - \pi\rvert), & y \in (4, 9] \end{cases}$<br>$D(t) = 0.1 + 5\sin^2(0.2\pi t)$. | $x \in [0, 1], y \in [0, 4]$,<br>$y = \sin\lvert 2\pi x - \pi\rvert + \lvert 2\pi x - \pi\rvert$.<br>and<br>$x \in [0, 1], y \in (4, 9]$,<br>$y = \sin\lvert 2\pi x - \pi\rvert + \lvert 2\pi x - \pi\rvert + 4$. |
| dMMOP7 | $\boldsymbol{x} = (x_1, x_2)^T, x_{1:2} \in [0, 1]$,<br>$\begin{cases} g_1 = 0.5 + D(t) \cdot (x_1 - 0.5), \\ g_2 = (1 - x_2)(1 - g_1), \\ g_3 = x_2(1 - g_1), \end{cases}$<br>$D(t) = \cos^2(0.2\pi t)$,<br>$\boldsymbol{y} = (y), y \in [0, 1]$,<br>$h(\boldsymbol{x}, \boldsymbol{y}, t) = 1 - \sin^2\left(2\pi(y - S(t)\sin(\pi x_1) + \frac{1}{4})\right)$,<br>$S(t) = 0.5\cos^2(0.2\pi t)$. | $y = S(t)\sin(\pi x_1), x_{1:2} \in [0, 1]$.<br>and<br>$y = S(t)\sin(\pi x_1) + \frac{1}{2}, x_{1:2} \in [0, 1]$. |
| dMMOP8 | $\boldsymbol{x} = (x), x \in [0, 1]$,<br>$\begin{cases} g_1(\boldsymbol{x}, t) = x, \\ g_2(\boldsymbol{x}, t) = 1 - \sqrt{x}, \end{cases}$<br>$\boldsymbol{y} = (y), y \in [0, 1]$,<br>$h(\boldsymbol{x}, \boldsymbol{y}, t) = 1 - \sin(D(t)\pi y)$,<br>$D(t) = 1 + 9\sin^2(0.2\pi t)$. | $x \in [0, 1], y = \frac{0.5 + 2i}{D(t)}$,<br>$i = 0, 1, 2, \ldots$ |
| dMMOP9 | $\boldsymbol{x} = (x), x \in [1, 3]$,<br>$\begin{cases} g_1(\boldsymbol{x}, t) = \lvert x - 2\rvert, \\ g_2(\boldsymbol{x}, t) = 1 - \sqrt{\lvert x - 2\rvert}, \end{cases}$<br>$\boldsymbol{y} = (y), y \in [-1, 1]$.<br>$h(\boldsymbol{x}, \boldsymbol{y}, t) = \begin{cases} 2(y - \sin(2D(t)\pi\lvert y - 2\rvert + \pi))^2, & y \in [1, 2) \\ 2(y - \sin(2\pi\lvert y - 2\rvert + \pi))^2, & y \in [2, 3] \end{cases}$<br>$D(t) = 1 + 4\sin^2(\pi/2t)$ | $y \in [1, 2)$,<br>$y = \sin(2D(t)\pi\lvert x - 2\rvert + \pi)$<br>and<br>$y \in [2, 3]$,<br>$y = \sin(2\pi\lvert x - 2\rvert + \pi)$. |
| dMMOP10 | $\boldsymbol{x} = (x), x \in [-0.5, 0.5]$,<br>$\boldsymbol{y} = (y), y \in [-0.5, 0.5]$,<br>$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} \cos\theta(t) & -\sin\theta(t) \\ \sin\theta(t) & \cos\theta(t) \end{bmatrix}$<br>$\begin{cases} g_1 = x_r, \\ g_2 = 1/x_r, \end{cases}$<br>$h(\boldsymbol{x}, \boldsymbol{y}, t) = 1 - \cos^6(2\pi y_r)$,<br>$\theta(t) = 2\pi\sin^2(0.2\pi t)$. | $x_r \in [-0.5, 0.5]$,<br>$y_r = \{-\frac{1}{2}, \frac{1}{2}\}$. |

modified to have a locally optimal Pareto front. dMMOP4 is another variant of dMMOP2 which is constructed based on the idea proposed in [12]. By adding a minus sign to all objective functions of dMMOP2, the Pareto front of dMMOP4 changes from a triangular shape to an inverted triangular shape. As pointed out in [12], such a Pareto front shape is difficult for decomposition-based multi-objective optimization algorithms. dMMOP5 is a test problem whose Pareto set and Pareto front change from a continuous curve to multiple disconnected segments over time. Furthermore, the number of disconnected Pareto set and Pareto front segments also change dynamically. The dMMOP6 test problem has two equivalent Pareto sets which are the same as the MMF8 test problem in [28]. However, its Pareto front changes dynamically from a regular curve to a mixed convex/concave curve. The number of knee points on the Pareto front also changes dynamically over time. dMMOP7 has two equivalent Pareto sets, each of which is a time-varying manifold with two dimensions. Its Pareto front is a 2-dimensional plane that can degenerate into a line over time. The dMMOP8 test problem has a time-varying number of equivalent Pareto sets. dMMOP9 has two equivalent Pareto sets, one of which changes its geometry over time, while the other is always stationary. In dMMOP10, the equivalent Pareto sets rotate clockwise around the origin as time changes. Since the centroid of the Pareto set is always the origin, dMMOP10 is challenging for some dMOEAs that rely on centroid-based prediction models (e.g., [15]).

In conclusion, the proposed dMMOP test suite provides test problems with various characteristics, thus allowing researchers to evaluate the performance of an algorithm with respect to a wide variety of aspects.

## 6 Concluding Remarks

In this paper, we introduced a novel type of optimization problem, namely, dMMOPs by extending MMOPs into dynamic environments. Furthermore, we gave a formal definition for dMMOPs and analyzed some key challenges in solving them. Since test problems are essential for algorithm development, we proposed a general approach for constructing dMMOPs for benchmarks. In addition, we provided a novel test suite containing 10 novel dMMOPs. We believe that these test problems can help researchers to develop more efficient algorithms for solving real-world dMMOPs.

This paper only provides a preliminary study on dynamic multi-modal multi-objective optimization, many potential research topics are left for future work. For example, experimental results in Section 4.1 show that simply incorporating an existing change response mechanism to multi-modal multi-objective optimization algorithms (e.g., MMO-MOES in our experiments) does not yield satisfactory results, and we still need more efficient algorithms to handle dMMOPs in the future.

## Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant No. 61876075), Guangdong Provincial Key Laboratory (Grant No. 2020 B121201001), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No. 2017ZT07X386), The Stable Support Plan Program of Shenzhen Natural Science Fund (Grant No. 20200925174447003), Shenzhen Science and Technology Program (Grant No. KQTD2016112514355531).

## References

1. Azzouz, R., Bechikh, S., Said, L.B.: A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy. Soft Comput. **21**, 885–906 (2017)
2. Cao, L., Xu, L., Goodman, E.D., Bao, C., Zhu, S.: Evolutionary dynamic multiobjective optimization assisted by a support vector regression predictor. IEEE Trans. Evol. Comput. **24**(2), 305–309 (2020)
3. Coello, C.A.C., Sierra, M.R.: A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In: Proc. of MICAI 2004: Advances in Artificial Intelligence. pp. 688–697 (2004)
4. Deb, K.: Multi-objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Inc. (2001)
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
6. Deb, K., N., U.B.R., Karthik, S.: Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling. In: Proc. of Evolutionary Multi-Criterion Optimization. pp. 803–817 (2007)
7. Deb, K., Tiwari, S.: Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization. Eur. J. Oper. Res. **185**(3), 1062–1087 (2008)
8. Farina, M., Deb, K., Amato, P.: Dynamic multiobjective optimization problems: Test cases, approximations, and applications. IEEE Trans. Evol. Comput. **5**(8), 425–442 (2004)
9. Goh, C., Tan, K.C.: A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. IEEE Trans. Evol. Comput. **13**(1), 103–127 (2008)
10. Goldberg, D.E., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: Proc. of the Second International Conference on Genetic Algorithms and Their Application. pp. 41–49 (1987)
11. Greeff, M., Engelbrecht, A.P.: Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation. In: IEEE Congress on Evolutionary Computation. pp. 2917–2924 (2008)
12. Ishibuchi, H., Matsumoto, T., Masuyama, N., Nojima, Y.: Many-objective problems are not always difficult for Pareto dominance-based evolutionary algorithms. In: Proc. 24th European Conference on Artificial Intelligence (2020)
13. Jaszkiewicz, A.: On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - a comparative experiment. IEEE Trans. Evol. Comput. **6**(4), 402–412 (2002)

14. Jiang, S., Yang, S.: A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization. IEEE Trans. Evol. Comput. **21**(1), 65–82 (2016)
15. Li, Q., Zou, J., Yang, S., Zheng, J., Ruan, G.: A predictive strategy based on special points for evolutionary dynamic multi-objective optimization. Soft Comput. **23**, 3723–3739 (2019)
16. Lin, Q., Lin, W., Zhu, Z., Gong, M., Li, J., Coello, C.A.C.: Multimodal multi-objective evolutionary optimization with dual clustering in decision and objective spaces. IEEE Trans. Evol. Comput. **25**(1), 130–144 (2021)
17. Liu, Y., Ishibuchi, H., Nojima, Y., Masuyama, N., Shang, K.: A double-niched evolutionary algorithm and its behavior on polygon-based problems. In: Proc. of Parallel Problem Solving from Nature - PPSN XV. pp. 262–273 (2018)
18. Liu, Y., Yen, G.G., Gong, D.: A multimodal multiobjective evolutionary algorithm using two-archive and recombination strategies. IEEE Trans. Evol. Comput. **23**(4), 660–674 (2019)
19. Nguyen, T.T.: Continuous Dynamic Optimization Using Evolutionary Algorithms. Ph.D. thesis, The University of Birmingham (2010)
20. Peng, Y., Ishibuchi, H.: A decomposition-based multi-modal multi-objective optimization algorithm. In: Proc. of the 2020 IEEE Congress on Evolutionary Computation. pp. 1–8 (2020)
21. Peng, Y., Ishibuchi, H., Shang, K.: Multi-modal multi-objective optimization: Problem analysis and case studies. In: Proc. of IEEE Symposium Series on Computational Intelligence. pp. 1865–1872 (2019)
22. Petrowski, A.: A clearing procedure as a niching method for genetic algorithms. In: Proc. of IEEE International Conference on Evolutionary Computation. pp. 798–803 (1996)
23. Raquel, C., Yao, X.: Dynamic multi-objective optimization: A survey of the state-of-the-art. In: Evolutionary Computation for Dynamic Optimization Problems, pp. 85–106. Springer (2013)
24. Schütze, O., Vasile, M., Coello, C.A.C.: Computing the set of epsilon-efficient solutions in multiobjective space mission design. J. Aerosp. Comput. Inf. Commun. **8**(3), 53–70 (2011)
25. Shir, O.M.: Niching in evolutionary algorithms. In: Handbook of Natural Computing, pp. 1035–1069. Springer (2012)
26. Tian, Y., Liu, R., Zhang, X., Ma, H., Tan, K.C., Jin, Y.: A multi-population evolutionary algorithm for solving large-scale multi-modal multi-objective optimization problems. IEEE Tran. Evol. Comput. **25**(3), 405–418 (2020)
27. Yue, C., Qu, B., Liang, J.: A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems. IEEE Trans. Evol. Comput. **22**(5), 805–817 (2018)
28. Yue, C., Qu, B., Yu, K., Liang, J., Li, X.: A novel scalable test problem suite for multimodal multiobjective optimization. Swarm Evol. Comput. **48**, 62–71 (2019)
29. Zhang, K., Chen, M., Xu, X., Yen, G.G.: Multi-objective evolution strategy for multi-modal multi-objective optimization. Appl. Soft Comput. **101**, 107004 (2021)
30. Zhou, A., Zhang, Q., Jin, Y.: Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. IEEE Trans. Evol. Comput. **13**(5), 1167–1189 (2009)