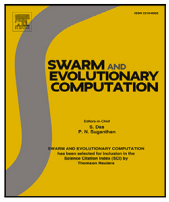
Swarm and Evolutionary Computation 89 (2024) 101621

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo



A dynamic multi-objective optimization evolutionary algorithm with adaptive boosting

Hu Peng [a,b,*], Jianpeng Xiong [a], Chen Pi [a], Xinyu Zhou [c], Zhijian Wu [d]

[a] School of Computer and Big Data Science, Jiujiang University, Jiujiang 332005, PR China
[b] Jiujiang Key Laboratory of Digital Technology, Jiujiang 332005, PR China
[c] School of Computer and Information Engineering, Jiangxi Normal University, Nanchang 330022, PR China
[d] School of Computer Science, Wuhan University, Wuhan 430072, PR China

ARTICLE INFO

Keywords:
Dynamic multi-objective evolutionary algorithm
Dynamic multi-objective optimization problem
Adaptive boosting mechanism

ABSTRACT

Dynamic multi-objective optimization problems (DMOPs) are prevalent in the real world, where the challenge in solving DMOPs is how to track the time-varying Pareto-optimal front (PF) and Pareto-optimal set (PS) quickly and accurately. However, balancing convergence and diversity is challenging as a single strategy can only address a particular type of DMOP. To solve this issue, a dynamic multi-objective optimization evolutionary algorithm with adaptive boosting (AB-DMOEA) is proposed in this paper. In the AB-DMOEA, an adaptive boosting response mechanism will increase the weights of high-performing strategies, including those based on prediction, memory, and diversity, which have been improved and integrated into the mechanism to tackle various problems. Additionally, the dominated solutions reinforcement strategy optimizes the population to ensure the effective operation of the above mechanism. In static optimization, the static optimization boosting mechanism selects the appropriate static multi-objective optimizer for the current problem. AB-DMOEA is compared with the other seven state-of-the-art DMOEAs on 35 benchmark DMOPs. The comprehensive experimental results demonstrate that the overall performance of the AB-DMOEA is superior or comparable to that of the compared algorithms. The proposed AB-DMOEA is also successfully applied to the smart greenhouses problem.

1. Introduction

Dynamic multi-objective optimization problems (DMOPs) involve multiple conflicting objective functions for optimization, in which environmental parameters, constraints, decision spaces, and the number of objectives might vary over time. DMOPs are widely used in real-world applications, such as power plant scheduling [1], vehicle routing [2], production scheduling [3], multi-period portfolio selection [4], control optimization of greenhouse [5], etc. DMOPs focus on solving MOPs within a time constraint and responding to environmental changes. In general, a DMOP can be considered as a series of consecutive static MOPs, and it can be solved by extracting relevant information from the various but interconnected MOPs. In recent years, more and more efficient multi-objective evolutionary algorithms (MOEAs) have been proposed and applied in various optimizations like micro population optimization [6], constrained optimization [7], complex engineering optimization [8], and so on. Technically, all MOEAs can be applied in dynamic multi-objective evolutionary algorithms (DMOEAs) to solve DMOPs. However, the Pareto-optimal set (PS) and the Pareto-optimal front (PF) will change over time (PS_t/PF_t), and they cannot track the constantly changing PS_t/PF_t when the environment changes. Hence, dynamic response strategies are required to address changes. A DMOEA comprises three main components, environmental change detection, a change response mechanism, and a static MOEA. In recent years, numerous DMOEAs [9] have been proposed to solve the DMOPs by developing efficient change response mechanisms and utilizing the latest static MOEA.

The environmental change response mechanism is a key part of a DMOEA and is commonly divided into the following types, diversity-based strategy, memory-based strategy, prediction-based mechanism, etc. The diversity-based strategy includes diversity increase and maintenance. The former replaces a certain proportion of the population with random or mutation solutions [1], while the latter maintains population diversity through special methods like two-archive structure [10]. The memory strategy fully utilizes historical information. The stored elite solutions be applied in a similar new environment to greatly improve convergence speed, particularly in periodic problems [11,12].

* Corresponding author at: School of Computer and Big Data Science, Jiujiang University, Jiujiang 332005, PR China.
E-mail address: hu_peng@whu.edu.cn (H. Peng).

https://doi.org/10.1016/j.swevo.2024.101621
Received 31 October 2023; Received in revised form 18 May 2024; Accepted 28 May 2024
Available online 9 June 2024

Prediction-based mechanism is widely used due to its adequacy and efficiency in utilizing information, and can mainly be divided into linear and nonlinear prediction methods. Linear prediction usually uses special mathematical models, such as autoregressive models [13], Kalman filters [14], while nonlinear prediction method includes transfer learning [15], fuzzy inference [16], reinforcement learning [17] and so on. Although they achieve positive outcomes when handling DOMPs, they are only equipped to deal with specific problems, such as periodicity and predictability. A sole mechanism is not robust enough and cannot fully address the various types of DMOPs.

As for the static multi-objective optimizer, most DMOEAs directly adopt or slightly improve the existing MOEAs, such as using the framework based on NSGA-II [18], MOEA/D [19], and a regularity model-based multi-objective estimation of distribution algorithm (RM-MEDA) [20], etc. Similar to dynamic response strategies, most MOEAs are targeted and difficult to generalize in most problems. The better convergence in MOEA/D, NSGA-II is more evenly distributed, and the performance of RM-MEDA is balanced. Man-Fai Leung and Jun Wang proposed a collaborative neurodynamic approach to multiobjective optimization [21], which combined with particle swarm optimization (PSO) has a good balance between convergence and diversity. Given the computational cost of high-dimensional problems and the convergence effect of high-frequency problems, a single strategy is hard to handle. Therefore, it is crucial to select an appropriate static multi-objective optimizer reasonably for the current problem environment.

Based on the above, a dynamic multi-objective optimization evolutionary algorithm with adaptive boosting, named AB-DMOEA, is proposed. When the environment changes, the adaptive boosting response mechanism is used and the three common methods are integrated into the mechanism, enabling the adaptive boosting of appropriate strategies to solve various problems. Otherwise, a novel adaptive selection mechanism is needed to select the appropriate static multi-objective optimizer. The main contributions of this work are summarized as follows.

(1) The adaptive boosting response mechanism calculates the real-time static optimization distance of various strategies and enhances the weights of the strategy optimization population corresponding to shorter values.
(2) To ensure the effectiveness of the adaptive boosting response mechanism, the dominated solution reinforcement strategy detects the proportion of non-dominated solutions beforehand. If the proportion is less than that of the dominated solution, the strategy moves the dominated solution and performs mutation for optimization.
(3) The static optimization boosting mechanism selects the optimal static multi-objective optimizer based on the weight of non-dominated solutions and Maximum Extension Distance (MED) values of the total population during the first change response.

Compared with current state-of-the-art DMOEAs, AB-DMOEA is more effective in dealing with different types of problems, while computational cost is lower. The adaptive boosting response mechanism integrates multiple strategies and adaptively enhances the optimization weights of appropriate strategies based on their real-time performance. The effectiveness of the mechanism can be ensured and the population can be strengthened in discontinuous, nonlinear, and other environments by implementing the dominated solution reinforcement strategy. In static optimization, the optimal static multi-objective optimizer is selected during the first environmental change, which ensures effective optimization throughout the entire process.

The remainder of this paper is organized as follows. Section 2 describes the related work of dynamic multi-objective optimization and the motivation of this paper. Section 3 describes the specific details of the proposed AB-DMOEA. The study of the experiment is given in Section 4 and more experiments and discussions are given in Section 5. Section 6 concludes the paper and prospects for future work.

2. Background

This section provides an overview of dynamic multi-objective optimization, including the concepts of DMOP and the basics of existing DMOEAs. Additionally, it presents related works and the motivation behind the proposed algorithm are presented.

2.1. Dynamic multi-objective optimization problem

There are multiple conflicting objective functions in a DMOP, which can be defined as follows in [22]:

$$\begin{aligned} \min \quad & F(x,t) = \left(f_1(x,t), \ldots, f_{M_t}(x,t)\right)^T, \\ \text{s.t.} \quad & \begin{cases} h_i(x,t) = 0, & i = 1, \ldots, n_h(t), \\ g_i(x,t) \geq 0, & i = 1, \ldots, n_g(t), \\ x \in \Omega_x, & t \in \Omega_t, \end{cases} \end{aligned} \tag{1}$$

where $x = \left(x_1, x_2, \ldots, x_{n_t}\right)$ denotes a vector with n_t-dimensional decision variables, M_t is the number of conflicting objectives. $F(x,t) : \Omega_x \times \Omega_t \rightarrow R^{M_t}$ is the vector of objective function that will evaluate the solution x at the time t. $h_i(x,t)$ denotes the equation constraint and $g_i(x,t)$ denotes the inequality constraint, where $n_h(t)$ and $n_g(t)$ are their quantities. Ω_x and Ω_t represent the space feasible decision and the space of time variables, respectively. The discrete-time variable t can be formulated as:

$$t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor \tag{2}$$

where n_t is the changing severity of the environment, τ_t is the changing frequency of the environment, τ is the iteration counter, $\lfloor \cdot \rfloor$ is the floor function, and t controls the dynamic evolution process. Here are some definitions and four possible types of time-varying change in the DMOP:

Definition 1. Dynamic Pareto dominance. At discrete time instant t, any two feasible solutions $\mathbf{x}, \mathbf{y} \in \Omega_t$ satisfy $\mathbf{x}$ dominates $\mathbf{y}$, denoted as $\mathbf{x} \prec_t \mathbf{y}$, if and only if:

$$\begin{cases} \forall i \in \{1, \ldots, M\}, f_i(x,t) \leq f_i(y,t) \\ \exists j \in \{1, \ldots, M\}, f_j(x,t) < f_j(y,t) \end{cases} \tag{3}$$

Definition 2. Dynamic Pareto-optimal solutions set (PS_t). A set of all Pareto-optimal solutions at time t is called the Pareto optimal solution set, denoted as PS_t, that is :

$$PS_t = \left\{ x \in \Omega_t \mid \nexists y \in \Omega_t, y \prec_t x \right\} \tag{4}$$

Definition 3. Dynamic Pareto-optimal front (PF_t). The set of Pareto-optimal solution sets mapped in the target space at time t constitutes the Pareto front PF_t, that is:

$$PF_t = \left\{ F(x,t) \mid x \in PS_t \right\} \tag{5}$$

- Type I : PS_t changes with time but PF_t stays the same.
- Type II : Both PS_t and PF_t changes with time.
- Type III : PS_t stays the same but PF_t changes with time.
- Type IV : Both PS_t and PF_t remain unchanged.

Research in dynamic multi-objective optimization concentrates on the first three types of DMOPs. An efficient way to deal with DMOPs is to combine MOEAs with dynamic response strategies, referred to as dynamic multi-objective optimization algorithms (DMOEAs).

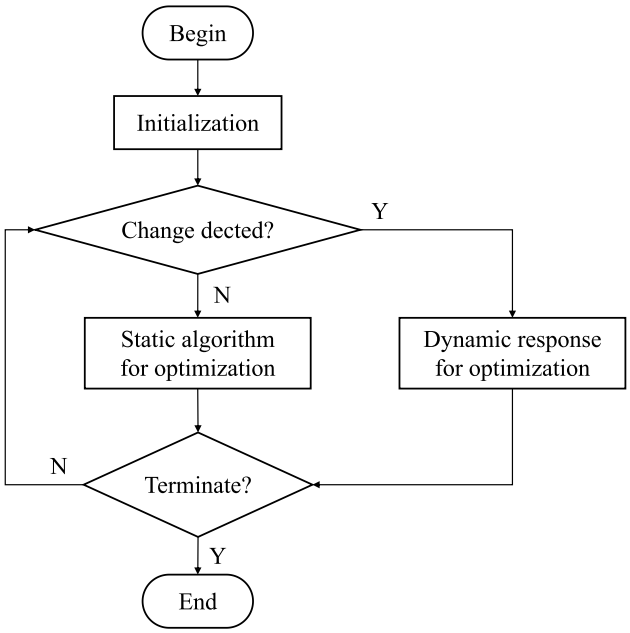


Fig. 1. The flowchart of DMOEA. Initialize a population and set related parameters, then detect whether the environment has changed. When a change in the environment is detected, appropriate dynamic response strategies are adopted to respond to environmental changes. Otherwise, a static algorithm is used to optimize the population. If the termination condition is met, the algorithm process ends, otherwise it proceeds to change detection.

2.2. Dynamic multi-objective optimization algorithm

The flowchart and illustration of dynamic multi-objective optimization algorithms are shown in Fig. 1, and the basic steps of the dynamic multi-objective optimization algorithm are the following:

Step 1 . Initialization. Initialize the population and set the relevant parameters for the algorithm.

Step 2 . Change detection. When a change in the environment is detected, go to **Step 3**. Otherwise, it proceeds to **Step 4**.

Step 3 . Dynamic response mechanism. Respond to environmental changes with appropriate dynamic strategies when the environment changes, then it proceeds to **Step 5**.

Step 4 . Static multi-objective optimization. Optimize the population by using a static multi-objective optimizer when the environment remains unchanged.

Step 5 . Termination. The algorithm process terminates if the termination conditions are met. Otherwise, it proceeds to **Step 2**.

2.3. Related work

Environment change detection, environment change response, and static multi-objective optimizer are three components of DMOEAs. In this subsection, we will provide a detailed introduction to each of these three aspects.

2.3.1. Environmental change detection

Environmental change detection determines whether the environment has changed to determine whether the dynamic response is necessary. To summarize the change detection process, there are three mechanisms for detecting these changes from [22].

(1) Random re-evaluation

Randomly re-evaluating [23] a population proportion at the start of each generation is an early and widely used method for detecting changes in DMOPs [24]. It entails selecting specific or random individuals in each generation of the population for testing and to determine whether the environment has changed through pre and post-objective function values. However, this method may lead to delays in responding to changes. Therefore, researchers have developed improved methods, such as steady-state change detection [25], to address these limitations. Liu et al. proposed an interaction-based prediction for dynamic multiobjective optimization (IP-DMOEA) [26], which samples multiple detectors from the search space, rounded up to 20% of the population size, to detect environmental changes.

(2) Sensor-based detection

Sensor-based detection [27] is a more complex and precise method that uses some special detectors, which are just like well-configured sensors, to monitor changes in the environment. Sensor-based detection is a more complex and precise method than random re-evaluation, but the detectors have a monitoring coverage limit or may overreact. Wang et al. [28] proposed an improved method that divides the population into clusters and installs sensors in each cluster to monitor changes in the environment.

(3) Population-based detection

Population-based detection [29] is a method that uses statistical methods to test for significant differences between the population of the current generation and that of the preceding generation to detect environmental changes. Sahmound et al. [30] hybridized sensor-based schemes and statistical approaches to improve detection performance.

2.3.2. Environmental change response

Upon detection of a change in the environment, algorithms must respond timely and effectively. Over the past decades, considerable effort has been dedicated to developing efficient dynamic response strategies. The following is a summary of the currently commonly used strategies for responding.

(1) Diversity-focused approaches

Diversity-focused approaches are early and effective methods, including diversity increase and maintenance. The method proposed by Deb et al. [1] to combine non-dominated sorting genetic algorithm II (NSGA-II) [18] with introducing some randomly generated solutions (DNSGA-II-A) and hyper-mutating some existing solutions (DNSGA-II-B). Peng et al. [31] proposed an exploration strategy based on the population centroids of the previous and current environments to increase diversity. Generally, the method based on diversity is easy to comprehend and significantly improves the diversity of a population, but responding to environmental changes randomly may mislead the evolutionary direction of the population and the convergence speed is usually slow. Zhang et al. proposed a novel hybrid DMOEA of elitism-based transfer learning and diversity maintenance (HETD-DMOEA) [32], which combines elite-based transfer learning with diversity maintenance to respond to environmental changes.

(2) Multi-population approaches

Multi-population approaches divide the population into multiple sub-populations dispersed in the search space, which is particularly effective in maintaining diversity. The dCOEA [12] divides the population into two sub-populations based on the sensitivity of decision variables to the environment: high-sensitive and low-sensitive sub-populations. The former uses a linear predictive model, while the latter uses Cauchy mutation to populate decision variables. In CGLP [33], Yu et al. propose a correlation analysis that classifies the population into

three sub-populations and applies prediction and diversity preservation to them, respectively.

(3) Prediction-based approaches

Prediction-based approaches, which fully utilize historical information and more accurately determine the location of new PS_t/PF_t, have received widespread attention in solving DMOPs. The linear prediction uses the linear prediction model based on the historical location of the population and the center point combined. Zhou et al. [13] proposed a population prediction strategy (PPS), which divided the population into a center point and a manifold. When a change occurs, the center point and manifold of the new environment are forecasted by the autoregressive (AR) model. Yan et al. [34] proposed a manifold clustering-based prediction strategy (MCP), extracting local linear manifolds from the historical population structure and dividing the population into multiple clusters based on the distinct linear manifolds to which they are attached. This method fully utilizes information such as the neighborhood distribution of individuals and the local correlations in the decision space.

(4) Memory-based approaches

Memory-based approaches effectively cope with the DMOPs with periodic changes, which use memory pools to store the optimal solutions in the historical environment and will be used in new environments similar to the old solution. Jiang et al. [25] proposed a steady-state and generational evolutionary algorithm (SGEA) that keeps half of the best old solutions in the new environment, which not only helps with fast convergence but also improves population diversity. In PMS [31], non-dominated solutions for each past environment were stored in a fixed-size memory pool, which is used to replace the worst individuals of the current population. When the memory pool reaches capacity, the first-in-first-out principle (FIFO) is utilized. Memory-based approaches have limited effectiveness in non-periodic problems, therefore it is essential to hybrid with other strategies [35].

(5) Dynamics-based approaches

Dynamics-based approaches respond to changes by evaluating the impact of changes on the algorithms. This approach involves two main branches: estimates of the severity of change and change-based variable grouping. Azzouz et al. [36] proposed the Dy-NSGA-II belongs to the former, which adjusted the number of memories and random individuals based on the severity of the change. The PBDMO [37] belongs to another type, which groups decision variables into primary and non-principal types and adopts different strategies respectively. The disadvantage of this method is that the results of the classification process may not be accurate.

(6) Special-point-based approaches

Special points contain key information regarding PF and have been widely used in optimizing stationary environments [38] for a considerable period. The combination of special points with other strategies or models can effectively predict the population in new environments. The CKPS [39] proposed by Zou et al. combined the center points with knee points and used the AR model for the prediction. Li et al. proposed a special-point-based hybrid prediction strategy (SHPS), which integrates special points with PPS [13] and utilizes various strategies to make predictions, based on the historical information available. Special points are receiving increasing attention, which makes their use in combination with other methods a very promising approach, such as Zheng et al. proposed center and boundary points-based strategy (PCPB) [40], and pivot points-based strategy (HPPDS) [41].

(7) Robustness-based approaches

When environmental changes are severe, the cost of producing excellent solutions increases, and robustness-based approaches can effectively solve such problems. Guo et al. [42] utilize robust optimization over time (ROOT), which is combined with multiple predictive models to construct a strong robustness model to track the PS_t/PF_t. He also introduces grid-based clustering and a hybrid mutation operator [43] to improve population diversity. Robustness-based approaches provide a low-cost environmental response but still have some limitations in terms of changes.

In addition to the seven types of methods approaches above, many new methods have been proposed in the last few years. Peng et al. proposed a multi-strategy dynamic multi-objective evolutionary algorithm with hybrid change response (MDMEA-HCR) [9], which uses multiple response strategies based on the old population to generate multiple response sub-populations, then they are combined and optimized. Yan et al. [44] proposed decomposition-based inter-individual correlation transfer learning (DICTL) and dimension-wise learning (DL) strategies, which make the algorithm better and faster to adapt to dynamic environments. A decision variable classification strategy based on the degree of environmental change (DVCEC) [45] is proposed by Sun et al. DVCEC has developed an adaptive change detection method based on multiple regions, which divides the decision variables and uses different prediction methods.

2.3.3. Static multi-objective optimizer

After dynamic response or when no changes are detected in the environment, the current environment is a static MOP. In this case, any existing MOEA with minor modifications can be used as a static multi-objective optimizer (SMOA) for population evolution during this period, so proposing or selecting an excellent MOEA is crucial. Traditional MOEAs, such as NSGA-II [18], MOEA/D [19], RM-MEDA [20], etc., can quickly track PS_t/PF_t before the next environmental change and are widely used as static multi-objective optimizers. The improved version of simulated binary crossover (SBX) replaced by the DE operator named NSGA-II-DE [46] and MOEA/D-DE [46] are also popular. What a static multi-objective optimizer is required to do is quickly converge and maintain good diversity between every two changes.

2.4. Motivation

DMOP is known for its complexity and variability. PS_t/PF_t can be subject to drastic, non-periodic, high frequency, and other environmental changes that present great challenges for DMOEAs. As introduced in Section 2.3.2, a single response mechanism only solves a specific type of DMOP. For example, prediction strategies effectively handle conventional predictable problems, while memory-based strategies perform significantly on periodic problems, and diversity-based strategies ensure population diversity to some extent. Moreover, a single response mechanism might mislead the evolutionary direction of the population. Therefore, a proposed adaptive boosting response mechanism is proposed, which combines the three commonly used strategies mentioned above to handle DMOPs. The concepts of boosting in evolutionary optimization, such as transfer Adaboost (TrAdaboost) [47,48], are mainly derived from machine learning. The concept of adaptive boosting (AdaBoost) [49] in machine learning focuses on misclassified samples, to create a strong classifier. However, the proposed adaptive boosting method primarily rewards and adjusts the weights of strategies based on their performance at each time.

Similarly, the idea of dynamic response can be applied to static optimization. Different static multi-objective optimizers have different abilities and characteristics in optimizing problems, and a suitable one should be selected before the occurrence of environmental changes. Therefore, a static optimization boosting mechanism is proposed, which includes three classic MOEAs (NSGA-II-DE [46], MOEA/D-DE [46], RM-MEDA [20]). The three MOEAs iterate through the τ_t-generation optimization population and select the MOEA with the best performance as the static multi-objective optimizer when the environment changes for the first time. Besides, a dominated solutions reinforcement strategy is proposed to ensure the effectiveness of the adaptive boosting response mechanism in solving problems such as non-linearity, difficulty in tracking PS_t/PF_t, or significant changes in severity. The details of the proposed algorithm are described in detail in Section 3.

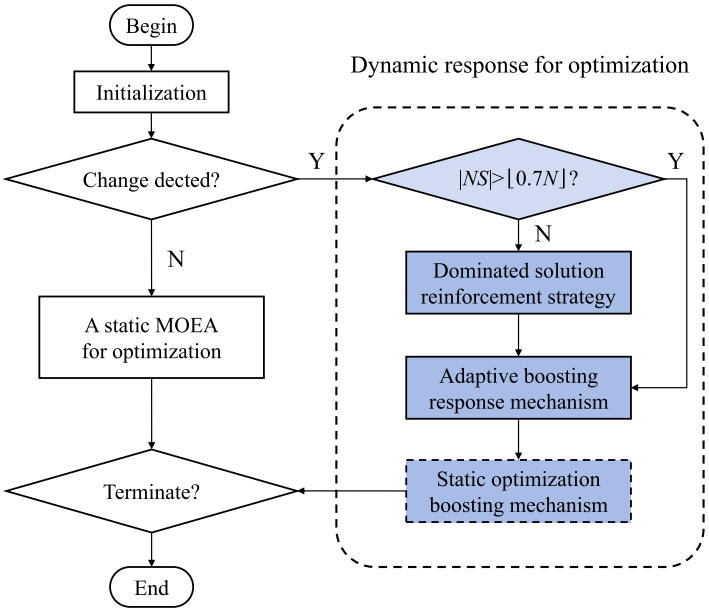


Fig. 2. The flowchart of AB-DMOEA. If an environmental change is detected, enter the dynamic response phase. When the number of non-dominated solutions is less than $\lfloor 0.7N \rfloor$, the dominated solution reinforcement strategy is used to strengthen population convergence; otherwise, the adaptive boosting response mechanism is directly for the response. The static optimization boosting mechanism only selects the appropriate SMOA when the environment changes for the first time. If the environmental change is not detected, the corresponding SMOA is used for static optimization.

3. The proposed AB-DMOEA

The proposed AB-DMOEA is presented in this section. First, the general framework of AB-DMOEA is outlined. Then, the methods proposed for adaptive boosting mechanisms in both dynamic and static environments are illustrated. Finally, the computational complexity of the proposed AB-DMOEA is analyzed.

3.1. The framework of AB-DMOEA

The dynamic response mechanism and static multi-objective optimizer are two parts of the optimization process. Most existing DMOEAs only focus on the environmental response mechanism, but the selection of static optimizers is also worth considering. We have proposed AB-DMOEA to address this issue, as illustrated in Fig. 2. The main contribution is to propose adaptive boosting mechanisms in both dynamic and static environments, which can effectively select strategies that are more suitable for the current environment. The framework of AB-DMOEA is presented in Algorithm 1.

The first step is to initialize the population and related parameters, the initial static multi-objective optimizer is MOEA/D-DE [46]. When a change in the environment is detected, the process will be transferred to dynamic processing. It consists of three parts, the dominated solution reinforcement strategy, the adaptive boosting response mechanism, and the static optimization boosting mechanism. If the number of non-dominated solutions is less than $\lfloor 0.7N \rfloor$, the dominated solutions boosting strategy would be used to optimize the population. When the environment changes for the first time, the static optimization boosting mechanism selects a static multi-objective optimizer that adapts to the current environment. The adaptive boosting response mechanism is the key response mechanism, which updates the population and the probability of each strategy being selected. If the environment has not changed, the population will be optimized by the corresponding SMOA.

Algorithm 1 AB-DMOEA

Input: Population size N, Maximum iterations $MaxGen$;
Output: A series of approximate PS_t and PF_t;
1: $t = 0$; Initialize parameter settings and population P_t;
2: SMOA = MOEA/D-DE;
3: **while** the termination condition is not satisfied **do**
4: **if** Environmental change detected **then**
5: $(P^t_{ND}, P^t_{DS}) = NondominatedSort(P_t)$;
6: **if** $\left|P^t_{ND}\right| < \lfloor 0.7 * N \rfloor$ **then**
7: P'_t = Dominated solution reinforcement strategy$(P_t, P^t_{ND}, P^t_{DS})$;
8: **end if**
9: (P_{t+1}, ρ_{t+1}) = Adaptive boosting response mechanism(P'_t, ρ_t);
10: **if** $t == 1$ **then**
11: SMOA = Static optimization boosting mechanism(P_t, τ_t);
12: **end if**
13: $t = t + 1$;
14: **else**
15: Update the population by SMOA;
16: **end if**
17: **end while**

3.2. Dominated solution reinforcement strategy

When AB-DMOEA encounters unpredictable and non-periodic issues, the convergence of the population is challenged and the performance of the adaptive boosting response mechanism becomes uncertain. Therefore, AB-DMOEA needs first to detect the quality of the current population, and if there are too many dominated solutions, using the dominated solution reinforcement strategy as shown in Fig. 3.

Foremost, the population is subjected to fast non-dominated sorting [18] and divided the population into non-dominated solutions ND and dominated solutions DS. Let C be the centroid in the dominated solution or non-dominated solution P, then C_{ND} and C_{DS} can be computed as:

$$C = \frac{1}{|P|} \sum_{x \in P} x \tag{6}$$

where $|P|$ is the cardinality of P, and $x = (x_1, \ldots, x_D)$ is a solution in P. The update method of the dominant solution is as follows:

$$x^* = x + C_{ND} - C_{DS} \tag{7}$$

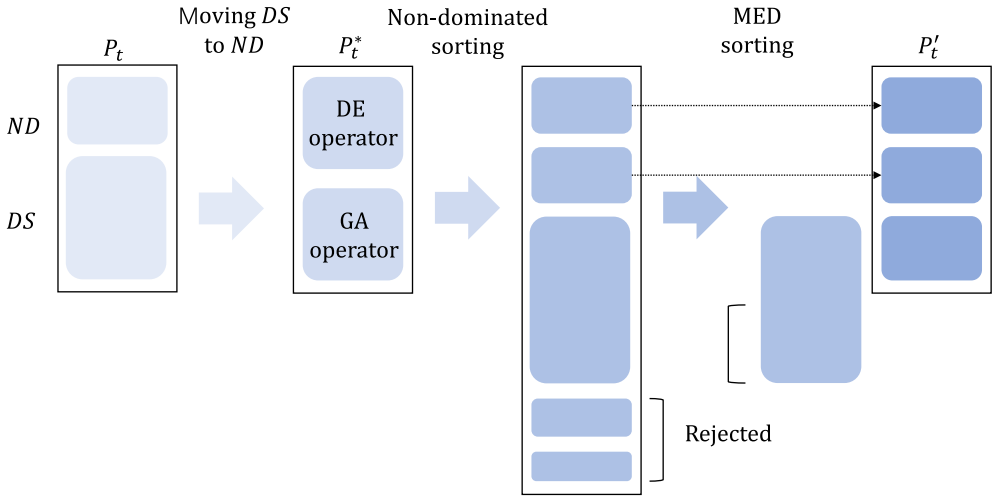


Fig. 3. The process of dominated solution reinforcement strategy in AB-DMOEA. The dominated solutions move to the dominated solutions, then perform the DE operator on half and the GA operator on the other half. Non-dominated sorting and selects the ones with lower non-dominated rankings. If they are the same, compare the MED values and choose higher ones.

Then, the population is evenly divided into two parts, using DE and GA operators to increase diversity, respectively. A large population consists of two mutated populations for fast non-dominated sorting and selects the lower ranking as the new population. The non-domination rank of an individual x_i denotes rank(x_i). When the non-dominated ranks are the same, calculate the Maximum Extension Distance MED [50] of each individual and select the individual with a larger MED. It is implemented as follows:

$$MED\left(P_t^i\right) = NearDist\left(P_t^i\right) \times TotalDist\left(P_t^i\right) \tag{8}$$

where $NearDist\left(P_t^i\right) = \min_{j,j\neq i}\left(\sqrt{\sum_{m=1}^{M}\left(f_m^j - f_m^i\right)^2}\right)$, $TotalDist\left(P_t^i\right) = \sum_{i=1}^{p}\left(\sqrt{\sum_{m=1}^{M}\left(f_m^j - f_m^i\right)^2}\right)$. Here, P_t^i denotes the ith individual of population P_t at the tth time, $NearDist(P_t^i)$ represents the minimum Euclidean distance from individual P_i^t to other individual P_t^j in the population P_t, and $TotalDist(P_t^i)$ is the sum of the Euclidean distances from individual P_t^i to others in the population P_t. The product of these two values is MED, and a larger $MED(x_i)$ indicates that the individual x_i is far away from other individuals.

Finally, the dominated solutions undergo movement, mutation, and selection to generate the reinforced population P_t'. The pseudo-code of the dominated solution reinforcement strategy is shown in Algorithm 2.

Algorithm 2 Dominated solution reinforcement strategy

Input: Population P_t, non-dominated set P_{ND} and dominated set P_{DS} of P_t;
Output: Reinforced population P_t';
1: $P_t' = \oslash$, $Archive = P_{ND}$;
2: Calculate the centroids of P_{DS} and P_{ND} by Eq. (6);
3: Generate a new individual x^* by Eq. (7);
4: Boundary check and put x^* into P^*;
5: Divide P^* into two parts and perform DE and GA operations to obtain $Archive_{DE}$ and $Archive_{GA}$;
6: $Archive = Archive \cup Archive_{DE} \cup Archive_{GA}$;
7: **for** i=1 : N **do**
8: **for** x_i and $x_{i+N} \in Archive$ **do**
9: Compare the rank(x_i) and the rank(x_{i+N}) and select the smaller value to place in P_t';
10: **if** rank(x_i)=x_{i+N} **then**
11: Compare the $MED(x_i)$ and the $MED(x_{i+N})$ and select the larger value to place in P_t';
12: **end if**
13: **end for**
14: **end for**

3.3. Adaptive boosting response mechanism

A single response strategy may only be effective for a certain type of problem, while the utilization of multiple strategies may occasionally lead to the wrong evolutionary direction of the population due to the failure of one strategy. To track time-varying PS_t/PF_t, this paper designs an adaptive boosting response mechanism that can continuously increase the weight of the most effective strategy based on its historical performance.

If the PS_t/PF_t of the population before each environmental change can approach the real PS_t/PF_t, then as shown in Fig. 4 is an example of PF_t change, the definitions as following: $O = \{O_1, \ldots, O_J\}$ is a collection of J different strategies, the set of black points $P_{t-1} = \{x_{t-1}^1, \ldots, x_{t-1}^N\}$ is the approximate PF_{t-1} of the last environmental change, which is close to the true PF_{t-1}. The set of green, blue, and yellow points $P^* = \{x_1^*, \ldots, x_N^*\}$ is the population generated by the J strategies when the environment changes. The set of orange points $P_t = \{x_t^1, \ldots, x_t^N\}$ is the population optimized through static optimization in time t. The solid red line represents the static optimization distance $d_t = \{d_t^1, \ldots, d_t^N\}$, which is the distance between the individual x_i^* in P_i^* and the nearest individual x_t^i in P_t^i.

The static optimization distance is the standard for evaluating dynamic response strategies. The shorter the distance, the more effective the corresponding strategy is generally in the environment. The average static optimization distance calculation formula is as follows:

$$\bar{d} = \frac{\sum_{i=1}^{N} d\left(x_t^{i,*}, P_t\right)}{N} \tag{9}$$

where d is the minimum Euclidean distance between $x_t^{i,*}$ and the individuals in P_t. If $d_i < \bar{d}$, it indicates that the strategy employed by the individual is advantageous. Therefore, the excellence rate of a strategy γ can be expressed by the number of outstanding individuals υ and the

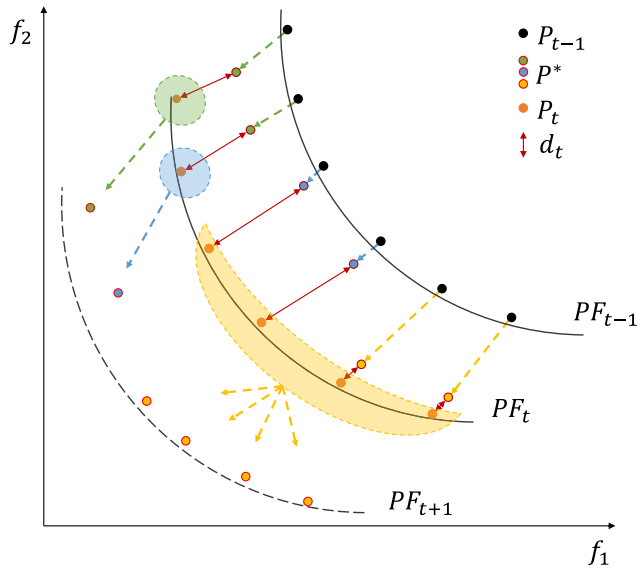


Fig. 4. An example of *PF* change process. The green, blue, and yellow points are sub-populations updated by three different strategies, while the orange points are the population by static optimization. The solid red line shows the static optimization distance.

total number of individuals κ in the strategy:

$$\gamma_j = \frac{\upsilon_j}{\kappa_j},\ j = \{1, \ldots, J\} \tag{10}$$

Each dynamic response strategy updates the weight by:

$$\omega_{t+1}^j = (1-\alpha)\omega_t^j + \alpha\gamma,\ j = \{1, \ldots, J\} \tag{11}$$

where J is the number of dynamic response strategies, $\alpha \in [0, 1]$, which is a rate used to balance between current and historical performance. Finally, the selection probability is updated by:

$$\rho_{t+1}^j = \rho_m + (1 - J * \rho_m) * \frac{\omega_{t+1}^j}{\sum_{j=1}^{J} \omega_{t+1}^j} \tag{12}$$

where $\rho_m \in [1, 1/J]$ is the lower bound of the selection probability, which prevents a certain strategy from being eliminated, and both parameters are empirical values as same as in [51]. The population is divided into multiple sub-populations by probability ρ_{t+1} and roulette wheel selection. The sub-populations are updated by different types of dynamic response strategies.

The adaptive boosting response mechanism calculates and evaluates the real-time static optimization distance of various strategies and their current environmental impact. For example, the blue individuals in Fig. 4 have the longest static optimization distances. Therefore, the corresponding strategies are not appropriate for this environment. In this case, the mechanism will reduce the weight of the strategy and substitute it corresponding strategy of outstanding yellow individuals. In this way, excellent strategies gradually eliminate inappropriate strategies making the algorithm adapt to different environments.

The mechanism has the potential to integrate numerous strategies to solve more types of problems. However, during initialization, the population is evenly distributed to each strategy, which excessive strategies reduce the number of optimized populations for each strategy, resulting in an unstable performance of the strategy. Therefore, the adaptive boosting response mechanism implements three effective strategies based on prediction, memory, and diversity. The three strategies are abbreviated as A, B, and C, and will be further elaborated upon in the subsection later. The adaptive boosting response mechanism is shown in Algorithm 3.

Algorithm 3 Adaptive boosting response mechanism

Input: Population P_t, centroid archive A_C, average objective value archive $A_{\bar{F}}$;
Output: Updated Population P_{t+1}, centroid archive A_C and average objective value archive $A_{\bar{F}}$;

```
Calculate the centroid C_t and average objective value F̄_t of P_t;
A_C = (A_C ∪ C_t), A_F̄ = (A_F̄ ∪ F̄_t);
P_ND=NonDominationSelection(P_t);
(P_t^A, P_t^B, P_t^C) = RouletteWheelSelection(P_t);
Calculate γ, ω_{t+1}, and ρ_{t+1} by Eq. (10) (11) (12);
P_{t+1}^A=Prediction boosting strategy(P_t^A);
(flag, index)=isSimilar(A_F̄, F̄_t);
Update A_F̄ and A_C using the FIFO principle;
P_{t+1}^B=Memory boosting strategy(P_t^B, A_C, flag, index);
P_{t+1}^C=Diversity boosting strategy(P_{t+1}^C, P_ND);
P_{t+1}=P_{t+1}^A ∪ P_{t+1}^B ∪ P_{t+1}^C;
```

3.3.1. Prediction boosting strategy

Prediction-based approach is one of the effective ways to deal with environmental changes. Linear prediction is a widely used method that predicts the location of new environmental populations based on changes in the centroid of the population before and after environmental changes [52]. Although the approach approximates well in linear problems, it performs poorly for more complex problems. To improve population uniformity and diversity, an improved prediction boosting strategy has been proposed.

Let C_t be the centroid in the current population P_t, then C_t can be computed as:

$$C_t = \frac{1}{|P_t|} \sum_{x \in P_t} x \tag{13}$$

where $|P_t|$ is the cardinality of P_t, and $x = (x_1, \ldots, x_D)$ is a solution in P_t. The moving direction of the centroid of the population in adjacent environments is calculated as follows:

$$D_t = C_t - C_{t-1} \tag{14}$$

The approximate position of the next environmental population can be predicted through D_t, and the proposed improved prediction method is as follows:

$$x_{t+1} = x_t + r_1 \times D_t + r_2 \times Gaussian(0, d) \tag{15}$$

where $x_t = (x_t^1, \ldots, x_t^D)$ is an individual in the population at time t, and x_{t+1} is a new individual generated through the proposed prediction strategy. Gaussian (0, d) represents a Gaussian perturbation with an average of zero and a standard deviation of $d = \frac{\|D_t\|}{|P_t|}$. The moving step $r_1 \in \{0.5, 1, 1.5\}$ divides the population into different parts, and $r_2 \in \{0, 1\}$ controls the presence or absence of Gaussian perturbation.

Linear prediction methods are generally effective in addressing predictable issues and are straightforward to apply. The proposed prediction boosting strategy improves the convergence and diversity of the predicted population to some extent by incorporating parameters r_1 and r_2. More details of the strategy are shown in Algorithm 4.

3.3.2. Memory boosting strategy

The centroid of the population is C_t, and the average value of each objective function is $\bar{F}_t = \left(\bar{f}_t^1, \ldots, \bar{f}_t^m\right)^T$. These values are stored in archive sets A_C and $A_{\bar{F}}$, respectively. When a change in the environment is detected, the current moment of $\bar{F}_t$ is compared with the historical $\bar{F}_t$ that is stored in $A_{\bar{F}}$. A threshold ε is set to determine the similarity of segment environments. If the difference between all objectives is less than 0.5ε, the storage environment represented by the index and the current environment are strongly similar, while values

Algorithm 4 Prediction boosting strategy.

Input: Sub-population P_t^A;
Output: Response sub-population P_{t+1}^A;
$P_{t+1}^A = \oslash$;
Calculate the centroid C_t of P_t^A by Eq. (13);
Calculate the moving direction D_t of P_t^A by Eq. (14);
for $i = 1 : |P_t^A|$ **do**
  if $i\%3 == 0$ **then**
    $r_1 = 0.5, r_2 = 0$;
  else if $i\%3 == 1$ **then**
    $r_1 = 1, r_2 = 1$;
  else
    $r_1 = 1.5, r_2 = 0$;
  end if
  Calculate x_{t+1} by Eq. (15);
  $P_{t+1}^A = P_{t+1}^A \cup x_{t+1}$;
end for

greater than 1.5ε indicate dissimilarity, and between the two values are considered weakly similar. Based on experience, the threshold ε has the same value in [35].

Algorithm 5 isSimilar

Input: The historical archive $A_{\bar{F}}$ and current value $\bar{F}_t$ of mean objective function;
Output: Environmental similarity indicator $flag$, similar position index $index$;
for $i = 1 : |A_{\bar{F}}|$ **do**
  $flag = 0, index = 0, count_w = 0, count_s = 0$;
  for $j = 1 : m$ **do**
    if $|A_{\bar{F}}^j - \bar{F}_t^j| < 0.5\varepsilon$ **then**
      $count_s + +$;
    end if
    if $0.5\varepsilon \le |A_{\bar{F}}^j - \bar{F}_t^j| < 1.5\varepsilon$ **then**
      $count_w + +$;
    end if
  end for
  if $count_w == m$ **then**
    $index = i, flag = 1$;
    Break;
  end if
  if $count_s == m$ **then**
    $index = i, flag = 2$;
    Break;
  end if
end for

The specific steps to determine environmental similarity are outlined in Algorithm 5, where $count_w$ and $count_s$ count the weak and strong similarity of each objective dimension, respectively. The environmental similarity is represented by $flag \in \{0, 1, 2\}$, with a higher value indicating greater similarity. The size of the memory archive is the same as the population size for effectiveness. When the memory archive is not full, the centroid of the population and the mean objective value are stored in archives A_C and $A_{\bar{F}}$, respectively. Otherwise, the memory archive is updated using the first-in-first-out strategy (FIFO) [35].

A strongly similar new solution will be generated as follows:

$$x_{t+1} = x_t + C_{index} - C_t \tag{16}$$

Algorithm 6 Memory boosting strategy

Input: Sub-population P_t^B, an archive of the centroid A_C, environmental similarity indicator $flag$, similar position index $index$;
Output: Response sub-population P_{t+1}^B;
$P_{t+1}^B = \oslash$;
for $i = 1 : |P_t^B|$ **do**
  if $flag == 2$ **then**
    Calculate x_{t+1}^i by Eq. (16);
  else if $flag == 1$ **then**
    $r_1 = 1, r_2 = 0$;
    Calculate x_{t+1}^i by Eq. (15);
  else
    $r_1 = 1, r_2 = 1$;
    Calculate x_{t+1}^i by Eq. (15);
  end if
  $P_{t+1}^B = P_{t+1}^B \cup x_{t+1}$;
end for

If there is no similar environment or weak similarity, a new solution can be generated by using Eq. (15) in the prediction boosting strategy. The specific implementation process of the memory boosting strategy can be found in Algorithm 6.

3.3.3. Diversity boosting strategy

To improve population convergence and diversity, prediction-based and memory-based approaches have limitations and may cause the algorithm to fall into the local optimum. To address this issue, most existing DMOEAs employ various diversity preservation strategies, for example, introducing mutation operators. Different mutation operators have different characteristics and impacts at different algorithm stages. To maintain diversity without increasing complexity, three effective mutation operators with different characteristics are introduced, genetic operator [18], polynomial mutation [53], and Gaussian mutation.

The genetic operator strongly associates the current individual with non-dominated solutions, increasing diversity and maintaining convergence. The polynomial mutation explores larger search spaces, avoiding local optimum. Gaussian mutation has better local search ability and is more popular due to its simplicity and effectiveness. For greater diversity, new individuals randomly use one of three mutation methods.

Algorithm 7 Diversity boosting strategy

Input: Sub-population P_t^C, non-dominated individuals P_{ND} of P_t;
Output: Response sub-population P_{t+1}^3;
$P_{t+1}^C = \oslash$;
for $i = 1 : |P_t^C|$ **do**
  R=random(3,1);
  if $R == 1$ **then**
    x_{t+1}^i = RandSelected($GeneticOperator(x_t^i \cup P_{ND})$, 1);
  else if $R == 2$ **then**
    x_{t+1}^i = RandSelected($PolynomialMutation(x_t^i \cup P_{ND})$, 1);
  else
    $r_1 = 0, r_2 = 1$;
    $x_{t+1}^i = ModifiedGaussianMutation$ by Eq. (15);
  end if
  $P_{t+1}^C = P_{t+1}^C \cup x_{t+1}$;
end for

As shown in Algorithm 7, randomly select one of three mutation methods, genetic operator, polynomial mutation, or Gaussian mutation.

If the selected method is a genetic or polynomial mutation, merge the individual with the non-dominated solutions of the current population. Then perform mutation and randomly select one as the new individual. If Gaussian mutation is selected, the new individual is generated by the Eq. (15).

3.4. Static optimization boosting mechanism

Existing DMOEAs pay little attention to static optimization and directly use a certain MOEA as the static optimizer, such as MOEA/D [19], RM-MEDA [20], etc. However the performance of different optimizers varies, for example, MOEA/D has excellent convergence due to its weight vector, but its distribution needs to be strengthened. Therefore, the idea of adaptive boosting can be introduced into static optimization as well, thus proposing the static optimization boosting mechanism.

AB-DMOEA selects the SMOA that performs the best when the first environmental change. The current population is randomly divided into three sub-populations of equal size and optimized independently by three commonly and useful SMOAs, MOEA/D-DE, RM-MEDA, and NSGA-II-DE. The static optimization time in DMOPs is τ_t, thus the maximum number of iterations here is also set to τ_t. Then, a scoring mechanism is proposed to evaluate the convergence and diversity of the static optimizer. The scoring mechanism includes the MED proportion score *MEDScore* and the non-dominated solution proportion score *NDScore*.

The *MEDScore* refers to the proportion of a single SMOA's MED to all MEDs. To calculate the *MEDScore*, the MED of each individual is calculated first by Eq. (8) to obtain the MED of the population. The calculation of *MEDScore* is as follows:

$$MEDScore = \frac{\mathrm{MED}(P_i)}{\sum_{i=1}^{3}\mathrm{MED}(P_i)} \tag{17}$$

The *NDScore* refers to the proportion of non-dominated solutions for each SMOA in all merged populations. It is necessary to record the SMOA used by each individual, then merge the three populations, and perform a fast non-dominated sorting to obtain the non-dominated solutions *SND*. The formula of *NDScore* is as follows:

$$NDScore = \frac{|SND_i|}{|SND|},\ i = \{1, 2, 3\} \tag{18}$$

Algorithm 8 Static optimization boosting mechanism

Input: Population P,change frequency τ_t,static optimizers SMOAs;
Output: a SMOA;

```
Randomly divide P into |SMOAs| sub-populations;
for t=1:τ_t do
    for i=1:N do
        P^i_{t+1} = SMOA_i(P^i_t);
    end for
end for
Calculate MEDScore for each SMOA;
Merge all sub-populations and calculate NDScore for each SMOA;
score = MEDScore + NDScore;
SMOA = Maxscore (SMOAs);
```

Finally, the *MEDScore* and *NDScore* are added to obtain the total score, with the highest score used as the following static optimizer. They represent the convergence and diversity of SMOA, respectively, and through this mechanism, the SMOA that performs well in both can be quickly selected in a short period. The specific implementation process of the static optimization boosting mechanism is shown in Algorithm 8.

3.5. Computational complexity analysis

The overall computational complexity of AB-DMOEA is mainly determined by the change response mechanism and the static multi-objective optimizer. The change response mechanism includes three parts. The time complexity of calculating the static optimization boosting mechanism in Algorithm 8 is $O(\tau_t MN^2)$, which can be ignored since it is executed once and the number of iterations is much smaller than the total number of iterations. The dominated solution reinforcement strategy takes $O(N^2)$ computational complexity. The complexity of the adaptive boosting response mechanism is $O(MN^2)$, due to the mutation operation was performed on each individual in the diversity boosting strategy, where M represents the number of objective functions and N is the population size. In reality, it is less than this value because the population is almost impossible to be completely updated by the strategy.

The cost of the static optimization process is also important. The computational complexity for NSGA-II is $O(MN^2)$, which was detailed by Deb et al. [18]. A deep comparison is conducted between MOEA/D and NSGA-II in [19], and the result showed that the computational complexity of MOEA/D is $O(MNT)$ better than that of NSGA-II, where T is the size of the applied neighborhood. The computational complexity of RM-MEDA includes modeling $O(DN)$, reproduction $O(DK)$, and the selection $O(MN^2)$, where D is the number of the decision space and K is the number of clusters. The overall computational complexity of RM-MEDA is $O(MN^2)$. Therefore, the worst-case computational complexity of AB-DMOEA is $O(MN^2)$.

4. Experimental study

This section introduces the benchmark suites, performance metrics, seven DMOEAs for comparison, related parameters of all algorithms, and the result of the experiment. The computational cost of all the algorithms is executed on a PC with Intel(R) Core(TM) i7-4790 CPU @3.60 GHz 8 GB of RAM, the operating system is Windows 7, and all experiments are conducted on Matlab R2021a.

4.1. Benchmark suites

To comprehensively evaluate the performance of the proposed algorithm, five sets of dynamic multi-objective benchmark suites were used, including the FDA [24], dMOP [54], F [13], UDF [55] and DF [56], with a total of 35 test instances. FDA and dMOP are the classic test suites, where most of the cases in them are periodically variable and predictable. F and UDF are challenging test suites that include various types of cases such as discontinuity and non-convexity. DF is the IEEE CEC 2018 dynamic multi-objective benchmark suite, which covers three types of DMOPs and various features in real-world scenarios.

4.2. Performance indicators

Many indicators are used to evaluate the performance of static MOEAs. The Inverted Generational Distance (IGD) [57] and Hypervolume (HV) [25] are particularly effective in evaluating distribution, convergence, and diversity, making them stand out. However, they are not applicable in dynamic environments, and the modified versions of IGD (MIGD) [13] and HV (MHV) [37] are applied for this purpose. The definitions and explanations of MIGD and MHV are as follows.

(1) IGD: Inverted Generational Distance is used to compute the approximation degree between the approximate PF and the true PF. At time t, the IGD is in the following form:

$$IGD(PF_t, PF_t^*) = \frac{\sum_{v \in PF_t^*} d(v, PF_t)}{|PF_t^*|} \tag{19}$$

where PF_t is an approximation of PF in environment t, PF_t^* is the set of sampled points on the true PF, $|PF_t^*|$ is the cardinality of PF_t^*, $d(v, PF_t)$ represents the minimum Euclidean distance between the solution v in PF_t^* and the solution in PF_t.

(2) MIGD: The MIGD indicator calculates the average IGD values over several time steps during a run, serving as a performance indicator for DMOPs. The MIGD indicator is defined as:

$$MIGD = \frac{\sum_{t\in T} IGD\left(PF_t, PF_t^*\right)}{|T|} \tag{20}$$

where T is a set of discrete time series in one run, and $|T|$ is the number of T. The lower MIGD value represents better optimization performance.

(3) HV: Hypervolume is also a composite evaluation metric that computes the volume of the region enclosed by a reference point in the approximate PF_t and objective space, and HV can be defined as:

$$HV = volume(PF_t, \overline{Z}^*) \tag{21}$$

where PF_t is a set of the approximate solution on PF in environment t, and $\overline{Z}^*$ is the reference point in the objective space.

(4) MHV: Mean Hypervolume (MHV) is an improved version of Hypervolume (HV) and provides a more accurate evaluation of the performance of DMOEAs. MHV is calculated as the mean HV value in a run, which is defined as:

$$MHV = \frac{\sum_{t\in T} HV(PF_t, \overline{Z}^*)}{|T|} \tag{22}$$

where T is a set of discrete time series in one run, and $|T|$ is the number of T. The higher value of MHV indicates the superior performance of solutions.

4.3. Compared algorithms

To confirm the superior performance of the proposed AB-DMOEA demonstrates superior, it is compared with seven state-of-the-art algorithms. These algorithms include SGEA [25], PBDMO [37], MOEA/D-HMPS [35], MOEA/D-ARMS [51], HPPCM [58], PCPB [40] and RVCP [59] all of which cover multiple different types of dynamic response strategies. This section will briefly describe each comparison algorithm:

(1) SGEA [25]: The proposed algorithm combines memory and prediction strategies, which exploits useful information extracted from the new environment, such as the direction of movement, to relocate the remaining portion of the population to regions near the time-varying PF_t.

(2) PBDMO [37]: Novel prediction strategies generate three subpopulations by using three strategies: linear prediction mode, sampling strategy, and shrinking strategy. The new population will be composed of three sub-populations.

(3) MOEA/D-HMPS [35]: Hybrid memory and prediction strategies can effectively cope with similar or dissimilar environmental changes. It first determines the similarity of environmental changes, then adopts the differential prediction method and the new memory-driven prediction method to deal with similar and dissimilar environments.

(4) MOEA/D-ARMS [51]: The algorithm combines an adaptive response mechanism selection strategy that adaptively selects effective response mechanisms from the response mechanism pool based on the recent and historical performance of each response mechanism. It addresses the shortcoming that a single response mechanism is suitable only for solving a certain type of problem.

(5) HPPCM [58]: Hybrid prediction strategy and a precision controllable mutation strategy handle problems more comprehensively. The former focuses on convergence and responds quickly to predictable changes, the latter increases diversity and responds to unpredictable changes well.

(6) PCPB [40]: The algorithm proposes a niche prediction strategy, which divides the radius of ecological niches by finding boundary points and center points in the objective space, and divides different niche populations in the decision space for prediction.

(7) RVCP [59]: The algorithm proposes adaptive reference vectors and linear prediction, then applies a noise-based individual expansion strategy to maintain the population in good diversity.

4.4. Parameter setting

(1) $t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor$, where τ is the current number of iterations, n_t and τ_t refer to the severity of the change and frequency of change. Following most algorithms based on historical information [13, 35,51], these two dynamic parameters were reasonably set to $\tau_t = 30$ and $n_t = 10$.

(2) The population size N was set to 100 for bi-objective problems and 105 for tri-objective cases. The dimension D of the decision variable was set to 10 for all the test instances.

(3) The total number of generations for all algorithms was set to $3n_t\tau_t + 50$, which means executing 50 generations of the static environment, and $3n_t$ environment changes are guaranteed.

(4) For change detection, a maximum number of 20% population members are re-evaluated.

(5) The number of sampling points set on the bi-objectives and tri-objectives real PF are 800 and 1200, respectively.

(6) The other parameter settings of the comparison algorithms were taken from the original papers to ensure the fairness of the experiment.

(7) The test results of all the algorithms in this paper were obtained by running them 30 times independently on the test instances. Additionally, the Wilcoxon rank-sum test [60] and Friedman test [61] at a significance level of 0.05 are conducted on the results obtained by all algorithms to provide statistically reasonable conclusions. "+", "−", and "≈" represent the compared algorithm is statistically significantly better, worse, and similar to the proposed AB-DMOEA.

4.5. Experimental result and analysis

Tables 1 and 2 show the mean and standard deviation of MIGD and MHV on 35 test instances. The optimal performance of each test instance is highlighted to make the experimental results more apparent when comparing these algorithms. It can be seen that AB-DMOEA achieved a total of 21 best MIGD and 19 best MHV in all cases, while the other algorithms obtained better MIGD values of $\{1, 3, 3, 1, 4, 6\}$ and better MHV values of $\{3, 7, 5, 1, 3, 7\}$. In the statistical results of the Friedman test, AB-DMOEA achieved rankings of 1.80 and 1.86 on MIGD and MHV, respectively. Obviously, AB-DMOEA has achieved better performance compared to the other seven advanced algorithms.

DF test suite covers three types of DMOPs with changes, 2 and 3 objective numbers, as well as various types of changes in PS and PF. The results indicate that the performance of AB-DMOEA in terms of MIGD and MHV is similar, the algorithm has significant advantages to comparative algorithms in DF1, DF2, etc. Instances like DF4 with small change ranges of the PF make it easier for AB-DMOEA to track the changes in the PF manifold. For both PS and PF with special types of changes and some high-dimensional cases, such as DF13 and DF14, AB-DMOEA performs generally.

FDA, dMOP, F, and UDF are classic test suites, that are widely used. AB-DMOEA exhibits excellent performance in dMOP1, FDA1, etc, due to improved prediction and memory-based strategies. In three-dimensional cases like FDA5, the three strategies used perform slightly worse than SGEA. In most of these cases, AB-DMOEA performs much better than other comparative algorithms, with only a few cases slightly inferior to the optimal algorithm, such as UDF4 and UDF7. The results

Table 1
Mean and standard deviation values of MIGD were obtained by seven algorithms on 35 test instances run independently 30 times. Wilcoxon rank sum test and Friedman test results are shown in the last two rows, with the best results highlighted.

Problem	M	D	SGEA [25]	PBDMO [37]	MOEA/D-HMPS [35]	MOEA/D-ARMS [51]	HPPCM [58]	PCPB [40]	RVCP [59]	AB-DMOEA
DF1	2	10	1.5793e−2 (1.14e−3) −	1.5208e−2 (3.28e−3) −	1.0931e−2 (3.96e−4) −	1.0595e−2 (6.84e−4) −	5.9485e−3 (6.87e−5) −	1.1664e−2 (7.05e−3) −	6.7005e−3 (1.91e−3) −	5.6372e−3 (1.91e−3)
DF2	2	10	1.2174e−1 (1.06e−2) −	2.9422e−2 (1.15e−2) −	4.3405e−2 (3.92e−3) −	3.4115e−2 (4.99e−3) −	6.7235e−3 (9.22e−5) −	8.6168e−3 (9.36e−3) ≈	9.4505e−3 (9.42e−3) ≈	6.5142e−3 (4.90e−4)
DF3	2	10	2.5918e−1 (3.71e−2) −	8.8393e−3 (1.18e−3) −	1.6529e−2 (5.97e−4) −	1.6973e−2 (6.36e−4) −	7.2272e−3 (1.46e−4) −	6.6330e−3 (2.52e−3) −	6.9757e−3 (2.59e−3) −	5.0240e−3 (4.69e−5)
DF4	2	10	7.9247e−2 (5.57e−3) −	6.9914e−2 (4.22e−4) −	8.6899e−2 (9.52e−4) −	8.5745e−2 (9.27e−4) −	7.1518e−2 (5.73e−4) −	7.9546e−2 (2.74e−3) −	8.0161e−2 (2.35e−3) −	6.7600e−2 (2.20e−4)
DF5	2	10	1.4580e−2 (4.19e−4) −	2.0169e−2 (4.45e−3) −	1.2413e−2 (5.27e−4) −	1.1875e−2 (7.83e−4) −	7.0551e−3 (3.59e−4) −	1.3682e−2 (8.90e−3) −	1.1084e−2 (6.00e−3) −	4.7283e−3 (1.11e−4)
DF6	2	10	1.6686e+0 (6.48e−1) −	4.9302e−1 (2.85e−1) ≈	2.3399e+0 (9.78e−1) −	2.5321e+0 (8.94e−1) −	1.9516e+0 (4.24e−1) −	7.0588e−1 (6.61e−1) ≈	1.5980e+0 (7.03e−1) −	6.1862e−1 (5.27e−1)
DF7	2	10	3.0269e−1 (7.55e−2) −	9.6942e−3 (3.07e−4) +	3.5888e−2 (2.54e−2) −	4.6973e−2 (3.39e−2) −	1.0239e−2 (1.72e−4) +	2.0305e−2 (1.34e−2) ≈	2.4492e−2 (2.14e−2) ≈	2.4692e−2 (2.15e−2)
DF8	2	10	1.6235e−1 (9.87e−3) −	1.2150e−1 (3.14e−4) +	1.2969e−1 (1.80e−3) −	1.2722e−1 (1.26e−3) −	1.2711e−1 (1.01e−3) −	1.2658e−1 (6.19e−3) ≈	1.2560e−1 (6.09e−3) ≈	1.2345e−1 (1.21e−3)
DF9	2	10	3.4986e−1 (4.22e−2) +	2.0888e+0 (6.55e−3) −	8.1832e−2 (2.01e−2) +	6.6515e−2 (7.05e−3) +	2.0823e+0 (1.52e−3) +	2.0753e+0 (1.05e−2) +	2.0749e+0 (7.56e−3) +	2.0843e+0 (2.54e−2)
DF10	3	10	6.8537e−2 (7.09e−3) +	7.1926e−2 (1.31e−3) +	1.5615e−1 (1.06e−2) −	1.6655e−1 (1.02e−2) −	9.2654e−2 (2.78e−3) +	1.3878e−1 (1.14e−2) −	1.1427e−1 (3.11e−2) ≈	1.0794e−1 (1.22e−2)
DF11	3	10	8.1150e−2 (9.76e−3) −	7.0989e−2 (8.16e−4) −	7.0937e−2 (2.96e−4) −	7.3250e−2 (3.68e−4) −	6.9985e−2 (5.60e−4) ≈	6.9450e−2 (2.65e−3) ≈	7.0240e−2 (3.47e−3) ≈	6.9849e−2 (4.85e−4)
DF12	3	10	3.3361e−1 (8.47e−2) −	2.5588e−1 (6.91e−4) −	2.7193e−1 (6.27e−4) −	2.7413e−1 (4.54e−4) −	2.5507e−1 (1.67e−3) ≈	2.6231e−1 (6.92e−3) −	2.6221e−1 (7.46e−3) −	2.5492e−1 (1.15e−3)
DF13	3	10	1.6798e−1 (2.88e−3) −	1.5326e−1 (2.26e−3) +	3.0935e−1 (5.27e−3) −	3.0132e−1 (5.69e−3) −	1.5219e−1 (1.25e−3) +	2.4035e−1 (8.91e−2) ≈	2.4043e−1 (8.68e−2) ≈	1.5522e−1 (1.43e−3)
DF14	3	10	7.0286e−1 (1.15e−1) −	1.5628e−1 (4.21e−2) ≈	5.3609e−2 (2.88e−4) +	5.4578e−2 (4.18e−4) +	1.9161e−1 (2.67e−2) −	9.8910e−2 (5.03e−2) +	9.8778e−2 (5.07e−2) +	1.4738e−1 (2.51e−2)
FDA1	2	10	7.7283e−3 (1.54e−3) −	3.9228e−2 (2.36e−2) −	5.7687e−3 (7.43e−5) −	6.4055e−3 (9.88e−5) −	6.3463e−3 (1.83e−4) −	7.2112e−3 (2.61e−3) −	5.7010e−3 (7.24e−4) −	4.5323e−3 (1.52e−4)
FDA2	2	10	6.4456e−3 (2.68e−4) −	6.1213e−3 (3.18e−4) −	5.3276e−3 (4.93e−5) −	5.8155e−3 (6.66e−5) −	6.5609e−3 (1.06e−4) −	4.6976e−3 (8.47e−5) −	5.1637e−3 (6.33e−4) −	4.4585e−3 (1.80e−5)
FDA3	2	10	3.6177e−2 (4.49e−3) +	6.4383e−2 (3.81e−3) ≈	1.2759e−2 (9.36e−4) +	1.1477e−2 (7.12e−4) +	6.3839e−2 (4.11e−3) ≈	3.6322e−2 (2.72e−2) +	3.5035e−2 (2.73e−2) +	6.0709e−2 (9.15e−3)
FDA4	3	10	7.9810e−2 (1.20e−3) −	7.7198e−2 (2.47e−3) −	7.3809e−2 (3.73e−4) −	7.5728e−2 (4.25e−4) −	7.0978e−2 (5.13e−4) ≈	7.5007e−2 (2.57e−3) −	7.4616e−2 (3.19e−3) −	6.9327e−2 (2.66e−3)
FDA5	3	10	1.2970e−1 (2.29e−3) +	1.3236e−1 (5.62e−3) +	1.8829e−1 (4.59e−2) −	1.9580e−1 (3.61e−2) −	1.6228e−1 (5.44e−3) −	1.7393e−1 (5.19e−2) ≈	1.4987e−1 (1.44e−2) ≈	1.4323e−1 (1.34e−2)
dMOP1	2	10	3.6802e−2 (2.43e−2) −	5.0240e−3 (3.49e−5) −	6.2150e−3 (1.31e−4) −	6.6168e−3 (1.26e−4) −	5.1898e−3 (5.41e−5) −	4.3272e−3 (9.77e−5) −	4.4582e−3 (1.74e−4) −	4.2384e−3 (3.84e−5)
dMOP2	2	10	8.6874e−3 (1.99e−3) −	5.6529e−2 (1.81e−2) −	6.0155e−3 (9.57e−5) −	6.6236e−3 (9.44e−5) −	6.7389e−3 (2.29e−4) −	9.9210e−3 (6.04e−3) −	6.0996e−3 (9.48e−4) −	4.6587e−3 (1.82e−4)
dMOP3	2	10	7.2642e−3 (8.86e−4) −	1.0120e−2 (1.45e−3) −	5.3464e−3 (4.61e−5) −	5.7736e−3 (7.52e−5) −	5.7379e−3 (7.96e−5) −	8.0889e−3 (3.83e−3) −	5.5473e−3 (9.02e−4) −	5.3008e−3 (2.73e−3)
F5	2	10	9.5593e−1 (1.48e−2) −	2.9774e+0 (1.51e+0) −	8.8625e−1 (3.47e−3) ≈	8.8775e−1 (2.65e−3) ≈	1.0037e+0 (1.92e−2) −	9.7456e−1 (1.15e−1) ≈	9.2823e−1 (5.09e−2) ≈	8.8912e−1 (6.98e−3)
F6	2	10	6.5839e−1 (5.27e−3) −	1.9043e+0 (5.33e−1) −	6.1448e−1 (4.56e−3) ≈	6.1917e−1 (2.74e−3) −	6.5686e−1 (1.23e−2) −	6.3628e−1 (4.70e−2) −	6.2733e−1 (2.05e−2) ≈	6.1372e−1 (3.52e−3)
F7	2	10	7.1174e−1 (3.09e−3) −	1.4616e+0 (3.71e−1) −	7.0373e−1 (2.04e−3) ≈	7.0479e−1 (3.03e−3) ≈	7.3302e−1 (5.76e−3) −	7.1103e−1 (1.11e−2) ≈	7.1412e−1 (1.34e−2) −	7.0314e−1 (2.77e−3)
F8	3	10	1.6830e−1 (8.99e−3) −	7.5439e−2 (1.50e−3) −	7.8560e−2 (4.70e−4) −	8.2309e−2 (5.95e−4) −	7.6391e−2 (5.42e−4) −	7.2842e−2 (1.34e−3) −	7.3006e−2 (1.28e−3) −	7.0913e−2 (6.21e−4)
F9	2	10	8.3449e−1 (2.18e−1) −	6.6807e−1 (7.64e−2) −	3.1694e−1 (4.84e−3) ≈	3.1662e−1 (3.35e−3) ≈	4.4729e−1 (2.00e−2) −	4.1940e−1 (1.36e−1) ≈	3.5434e−1 (4.59e−2) ≈	3.2356e−1 (2.08e−2)
F10	2	10	1.8023e+0 (3.27e−2) −	4.8840e+0 (3.33e+0) −	1.7199e+0 (1.10e−2) +	1.7149e+0 (6.05e−3) +	1.8368e+0 (1.68e−2) −	1.9025e+0 (2.70e−1) ≈	1.7592e+0 (5.57e−2) ≈	1.7386e+0 (2.43e−2)
UDF1	2	10	9.3171e−2 (2.51e−2) −	1.0023e−1 (1.20e−2) −	4.6586e−2 (3.48e−3) −	5.5469e−2 (2.25e−2) −	1.1373e−1 (1.12e−2) −	8.2118e−2 (4.42e−2) −	6.7908e−2 (2.94e−2) −	3.8828e−2 (4.96e−3)
UDF2	2	10	3.6964e−2 (7.09e−3) −	7.2477e−3 (7.28e−4) −	6.1713e−3 (5.53e−5) −	6.9695e−3 (8.63e−5) −	6.5686e−3 (7.97e−5) −	8.7626e−3 (3.80e−3) −	9.0270e−3 (4.00e−3) −	4.8384e−3 (4.32e−5)
UDF3	2	10	9.8721e−1 (7.33e−2) −	6.3013e−1 (9.78e−2) +	9.3160e−1 (1.37e−1) −	9.6901e−1 (9.57e−2) −	7.6546e−1 (1.44e−1) ≈	1.0051e+0 (1.54e−1) −	7.9728e−1 (1.57e−1) ≈	7.7964e−1 (1.76e−1)
UDF4	2	10	1.5918e−1 (4.46e−2) −	6.2952e−2 (3.40e−3) +	7.5824e−2 (2.47e−3) ≈	7.5887e−2 (3.66e−3) ≈	9.3242e−2 (2.79e−3) −	1.2932e−1 (6.23e−2) ≈	9.0344e−2 (2.50e−2) ≈	8.1010e−2 (1.27e−2)
UDF5	2	10	4.6342e−2 (2.92e−2) −	5.7112e−3 (3.50e−4) −	6.7329e−3 (8.27e−5) −	7.4115e−3 (1.50e−4) −	6.3945e−3 (2.99e−4) −	5.6220e−3 (1.90e−4) −	5.5820e−3 (1.75e−4) −	4.9300e−3 (1.60e−4)
UDF6	2	10	6.3895e−1 (1.44e−1) +	6.9486e−1 (6.46e−2) ≈	6.9097e−1 (1.47e−1) ≈	7.1390e−1 (1.93e−1) ≈	7.8008e−1 (6.24e−2) −	8.2827e−1 (2.08e−1) ≈	7.5587e−1 (1.41e−1) ≈	7.2161e−1 (8.38e−2)
UDF7	3	10	9.6396e−1 (1.43e−1) −	7.2964e−1 (1.72e−2) +	8.1108e−1 (4.66e−2) −	8.0457e−1 (2.98e−2) −	7.5561e−1 (1.40e−2) ≈	8.1870e−1 (4.12e−2) −	7.5367e−1 (2.98e−2) ≈	7.4696e−1 (2.02e−2)
+/−/≈			5/30/0	8/23/4	4/25/6	4/26/5	4/25/6	3/19/13	3/17/15	
Average ranking			6.26	4.94	4.26	4.89	4.57	4.89	3.91	2.29

"+", "−", and "≈" in the Wilcoxon test represent the performance of the algorithm is significantly better, worse, and similarly compared to AB-DMOEA, respectively. M is the number of objectives, and D is the dimensions of decision variables.

Table 2
Mean and standard deviation values of MHV were obtained by seven algorithms on 35 test instances run independently 30 times. Wilcoxon rank sum test and Friedman test results are shown in the last two rows, with the best results highlighted.

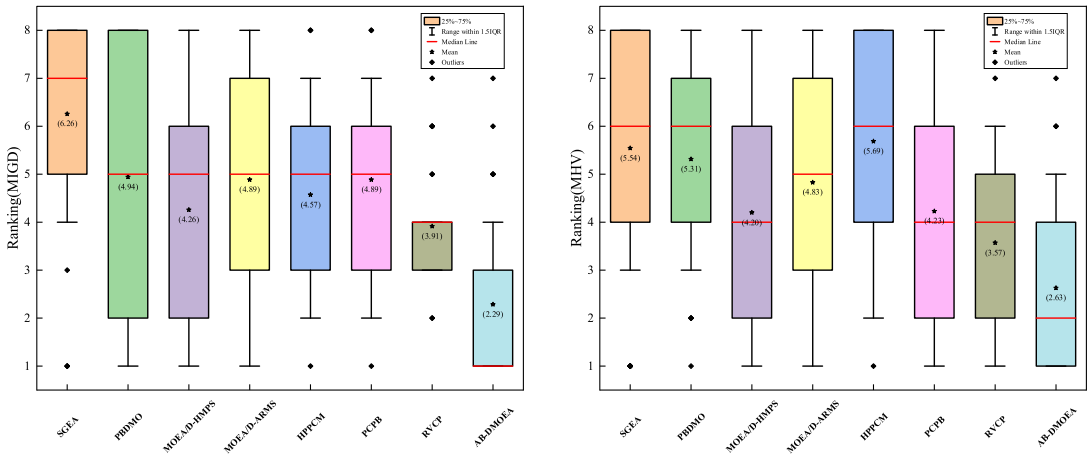
Problem	M	D	SGEA [25]	PBDMO [37]	MOEA/D-HMPS [35]	MOEA/D-ARMS [51]	HPPCM [58]	PCPB [40]	RVCP [59]	AB-DMOEA
DF1	2	10	5.0831e−1 (1.48e−3) −	5.1006e−1 (4.16e−3) −	5.1456e−1 (6.84e−4) −	5.1506e−1 (9.85e−4) −	5.2352e−1 (1.33e−4) −	5.1631e−1 (8.37e−3) −	5.2229e−1 (3.00e−3) −	5.2405e−1 (2.41e−3)
DF2	2	10	5.9159e−1 (9.51e−3) −	6.8249e−1 (1.47e−2) −	6.6911e−1 (4.43e−3) −	6.7795e−1 (4.37e−3) −	7.1486e−1 (1.51e−4) −	7.1222e−1 (1.23e−2) ≈	7.1101e−1 (1.26e−2) ≈	7.1493e−1 (7.81e−4)
DF3	2	10	2.7359e−1 (2.20e−2) −	4.8783e−1 (1.07e−3) −	4.7277e−1 (8.44e−4) −	4.7163e−1 (1.16e−3) −	4.8871e−1 (2.55e−4) −	4.8945e−1 (4.40e−3) −	4.8883e−1 (4.54e−3) −	4.9204e−1 (7.82e−5)
DF4	2	10	2.8344e−1 (3.39e−4) −	2.8502e−1 (6.87e−5) −	2.8327e−1 (1.13e−4) −	2.8317e−1 (1.06e−4) −	2.8453e−1 (2.18e−4) −	2.8656e−1 (2.12e−3) ≈	2.8648e−1 (2.07e−3) ≈	2.8531e−1 (2.25e−5)
DF5	2	10	1.2597e−1 (2.44e−4) −	1.2618e−1 (9.37e−4) −	1.2582e−1 (4.12e−4) −	1.2596e−1 (5.86e−4) −	1.2774e−1 (3.13e−4) −	1.3262e−1 (4.72e−3) ≈	1.3279e−1 (4.83e−3) ≈	1.2873e−1 (6.66e−5)
DF6	2	10	4.0128e−1 (1.12e−1) −	4.7060e−1 (9.36e−2) ≈	2.8810e−1 (1.41e−1) −	2.4659e−1 (1.27e−1) −	2.1812e−1 (1.11e−1) −	4.5855e−1 (1.76e−1) ≈	2.9736e−1 (1.43e−1) −	4.9921e−1 (1.44e−1)
DF7	2	10	1.2710e−1 (7.59e−3) −	1.5335e−1 (4.85e−5) +	1.5027e−1 (2.51e−3) −	1.4899e−1 (3.86e−3) −	1.5324e−1 (3.48e−5) ≈	1.5463e−1 (3.56e−3) ≈	1.5422e−1 (4.27e−3) +	1.5162e−1 (2.41e−3)
DF8	2	10	7.9409e−2 (2.33e−4) −	8.1898e−2 (8.38e−5) +	8.0769e−2 (1.95e−4) −	8.0986e−2 (1.57e−4) −	8.1109e−2 (2.35e−4) −	8.8672e−2 (7.53e−3) +	8.8664e−2 (7.44e−3) +	8.1769e−2 (8.97e−5)
DF9	2	10	9.0411e−2 (3.36e−3) −	7.9003e−2 (2.96e−2) −	3.5084e−2 (1.36e−3) −	3.5059e−2 (2.74e−4) −	3.4495e−2 (1.09e−3) −	4.1787e−2 (1.36e−3) −	9.7100e−2 (2.92e−2) −	1.0480e−1 (6.89e−4)
DF10	3	10	6.5958e−1 (2.20e−3) +	6.3651e−1 (1.68e−3) +	6.3012e−1 (5.76e−3) +	6.2827e−1 (5.06e−3) +	6.0733e−1 (3.97e−3) +	5.9381e−1 (5.16e−2) ≈	6.3480e−1 (9.30e−3) +	5.8032e−1 (1.41e−2)
DF11	3	10	2.8469e−1 (4.48e−3) −	2.8992e−1 (1.23e−3) −	2.9204e−1 (2.70e−4) −	2.9069e−1 (2.91e−4) −	2.9322e−1 (4.07e−4) −	2.9495e−1 (1.62e−3) ≈	2.9428e−1 (1.57e−3) ≈	2.9403e−1 (5.45e−4)
DF12	3	10	8.7897e−1 (2.08e−2) −	9.0661e−1 (8.73e−4) −	9.0818e−1 (3.31e−4) −	9.0842e−1 (7.42e−4) ≈	9.0920e−1 (4.11e−4) +	9.0648e−1 (4.89e−4) −	9.0765e−1 (3.07e−4) −	9.0859e−1 (7.42e−4)
DF13	3	10	6.5496e−1 (2.39e−3) +	6.2411e−1 (3.57e−3) −	5.7935e−1 (3.52e−3) −	5.6974e−1 (4.56e−3) −	6.2597e−1 (1.63e−3) −	6.0378e−1 (2.68e−2) −	6.0022e−1 (2.77e−2) −	6.3038e−1 (2.28e−3)
DF14	3	10	9.1999e−2 (2.94e−2) −	4.3547e−1 (9.95e−3) −	5.8661e−1 (6.92e−4) +	5.8754e−1 (3.54e−4) +	3.9959e−1 (9.20e−3) −	4.3133e−1 (8.07e−3) −	4.2906e−1 (7.44e−3) −	5.2959e−1 (7.14e−2)
FDA1	2	10	7.1455e−1 (1.05e−3) −	6.8307e−1 (1.71e−2) −	7.1624e−1 (1.24e−4) −	7.1525e−1 (1.79e−4) −	7.1570e−1 (2.69e−4) −	7.1469e−1 (3.16e−3) −	7.1639e−1 (1.04e−3) −	7.1838e−1 (3.22e−4)
FDA2	2	10	5.2860e−1 (2.47e−4) −	5.2833e−1 (5.05e−4) −	5.2903e−1 (7.68e−5) −	5.2836e−1 (9.04e−5) −	5.2763e−1 (1.25e−4) −	5.2999e−1 (9.97e−5) −	5.2928e−1 (8.26e−4) −	5.3035e−1 (3.95e−5)
FDA3	2	10	5.6210e−1 (3.83e−3) +	5.3011e−1 (3.45e−3) −	5.8295e−1 (9.95e−4) +	5.8415e−1 (7.89e−4) +	5.3186e−1 (3.16e−3) −	5.6414e−1 (2.23e−2) +	5.6353e−1 (2.44e−2) +	5.3840e−1 (8.00e−3)
FDA4	3	10	5.4040e−1 (1.31e−3) +	5.0371e−1 (4.54e−3) −	5.2659e−1 (7.00e−4) +	5.2369e−1 (7.67e−4) ≈	5.1900e−1 (1.47e−3) ≈	5.1831e−1 (5.05e−3) ≈	5.1823e−1 (3.61e−3) ≈	5.1996e−1 (5.38e−3)
FDA5	3	10	5.4130e−1 (1.33e−3) +	5.1821e−1 (2.36e−3) −	5.2142e−1 (8.78e−3) ≈	5.1891e−1 (6.99e−3) ≈	5.1455e−1 (1.67e−3) −	5.1810e−1 (6.50e−3) ≈	5.2410e−1 (5.19e−3) ≈	5.1991e−1 (5.28e−3)
dMOP1	2	10	4.9470e−1 (2.26e−2) −	5.2563e−1 (5.11e−5) −	5.2242e−1 (2.71e−4) −	5.2186e−1 (2.46e−4) −	5.2507e−1 (1.07e−4) −	5.2649e−1 (4.03e−4) ≈	5.2612e−1 (2.28e−4) ≈	5.2627e−1 (6.48e−5)
dMOP2	2	10	5.2055e−1 (1.96e−3) −	4.7038e−1 (1.48e−2) −	5.2280e−1 (1.93e−4) −	5.2182e−1 (1.65e−4) −	5.2217e−1 (3.74e−4) −	5.1909e−1 (6.24e−3) −	5.2326e−1 (9.09e−4) −	5.2539e−1 (4.48e−4)
dMOP3	2	10	7.1518e−1 (5.85e−4) −	7.1023e−1 (1.71e−3) −	7.1701e−1 (9.22e−5) −	7.1638e−1 (1.27e−4) −	7.1674e−1 (1.11e−4) −	7.1363e−1 (4.83e−3) −	7.1672e−1 (1.31e−3) −	7.1733e−1 (3.44e−3)
F5	2	10	2.7624e−1 (7.83e−3) −	2.8450e−1 (1.75e−2) −	3.2708e−1 (1.46e−3) −	3.2761e−1 (7.52e−4) +	2.2493e−1 (1.21e−2) −	2.9924e−1 (3.61e−2) −	2.8972e−1 (4.37e−2) ≈	3.2742e−1 (7.03e−3)
F6	2	10	2.4146e−1 (1.67e−3) −	2.2092e−1 (9.69e−3) −	2.6354e−1 (7.81e−4) −	2.6359e−1 (6.77e−4) −	2.0779e−1 (8.32e−3) −	2.4848e−1 (2.20e−2) −	2.3909e−1 (2.84e−2) −	2.6386e−1 (4.03e−3)
F7	2	10	2.3187e−1 (1.65e−3) −	1.6040e−1 (3.00e−2) −	2.3768e−1 (3.16e−4) ≈	2.3717e−1 (2.59e−4) ≈	2.1543e−1 (4.47e−3) −	2.3326e−1 (6.70e−3) ≈	2.3346e−1 (6.32e−3) ≈	2.3734e−1 (1.45e−3)
F8	3	10	4.3024e−1 (9.17e−3) −	5.0720e−1 (2.19e−3) −	5.0918e−1 (1.22e−3) −	5.0324e−1 (1.34e−3) −	5.0266e−1 (1.45e−3) −	5.1326e−1 (3.41e−3) ≈	5.1260e−1 (3.17e−3) ≈	5.1199e−1 (1.11e−3)
F9	2	10	3.0315e−1 (2.04e−2) −	3.2801e−1 (1.94e−2) −	3.9259e−1 (1.10e−3) −	3.9229e−1 (9.02e−4) −	2.5958e−1 (1.55e−2) −	3.6022e−1 (3.74e−2) −	3.4582e−1 (5.35e−2) −	3.9475e−1 (3.64e−3)
F10	2	10	2.6454e−1 (1.05e−2) −	1.5238e−1 (4.74e−2) −	3.0646e−1 (2.91e−3) ≈	3.0779e−1 (1.52e−3) ≈	2.3467e−1 (6.72e−3) −	2.6688e−1 (5.07e−2) −	2.7902e−1 (3.50e−2) ≈	3.0342e−1 (1.42e−2)
UDF1	2	10	2.3651e−1 (1.08e−2) −	2.2621e−1 (7.82e−3) −	2.5990e−1 (2.87e−3) −	2.5500e−1 (8.58e−3) −	2.1618e−1 (7.05e−3) −	2.4244e−1 (2.30e−2) −	2.5051e−1 (1.48e−2) −	2.6636e−1 (3.82e−3)
UDF2	2	10	2.7208e−1 (3.44e−3) −	2.9168e−1 (4.63e−4) −	2.9219e−1 (5.15e−5) −	2.9157e−1 (9.81e−5) −	2.9200e−1 (7.90e−5) −	2.9634e−1 (3.79e−3) ≈	2.9615e−1 (3.65e−3) ≈	2.9338e−1 (7.36e−5)
UDF3	2	10	4.9394e−1 (5.95e−2) +	3.5165e−1 (9.52e−2) ≈	4.8509e−1 (7.24e−2) +	4.4592e−1 (5.94e−2) +	1.2110e−1 (8.39e−2) −	2.5319e−1 (2.20e−1) ≈	3.6308e−1 (1.24e−1) ≈	3.9967e−1 (7.51e−2)
UDF4	2	10	4.0175e−1 (2.18e−2) −	4.4846e−1 (1.02e−2) +	4.3500e−1 (1.14e−2) ≈	4.3220e−1 (1.05e−2) ≈	3.9663e−1 (7.92e−3) −	3.8162e−1 (6.57e−2) ≈	4.1263e−1 (3.44e−2) ≈	4.2768e−1 (1.30e−2)
UDF5	2	10	5.1505e−1 (1.49e−2) ≈	5.1286e−1 (2.56e−4) −	5.1127e−1 (1.80e−4) −	5.1057e−1 (2.18e−4) −	5.1180e−1 (4.75e−4) −	5.1269e−1 (2.48e−4) −	5.1273e−1 (2.19e−4) −	5.1374e−1 (2.71e−4)
UDF6	2	10	1.0814e−1 (2.81e−2) +	1.0011e−1 (1.24e−2) +	9.8055e−2 (2.16e−2) ≈	1.0206e−1 (2.84e−2) ≈	8.8014e−2 (1.17e−2) ≈	7.1663e−2 (3.95e−2) ≈	9.0151e−2 (2.04e−2) ≈	8.7691e−2 (1.91e−2)
UDF7	3	10	1.4960e−1 (1.50e−2) −	1.8083e−1 (1.79e−3) +	1.7440e−1 (1.24e−3) −	1.7175e−1 (1.80e−3) −	1.7380e−1 (1.95e−3) −	1.7630e−1 (2.72e−3) ≈	1.8378e−1 (6.52e−3) +	1.7694e−1 (2.48e−3)
+/−/≈			7/27/1	6/27/2	5/25/5	5/23/7	2/30/3	2/16/17	5/15/15	
Average ranking			5.54	5.31	4.20	4.83	5.69	4.23	3.57	2.63

"+", "−", and "≈" in the Wilcoxon test represent the performance of the algorithm is significantly better, worse, and similarly compared to AB-DMOEA, respectively. M is the number of objectives, and D is the dimensions of decision variables.

indicate that the proposed adaptive boosting response mechanism can effectively address sophisticated issues. Overall, AB-DMOEA achieved good performance in all test suites.

For a more intuitive representation of the Friedman test ranking results of MIGD and MHV for seven algorithms in Tables 1 and 2, the box plots are shown in Fig. 5. Schematically, AB-DMOEA shows the best performance of all the algorithms according to the mean and median values of both MIGD and MHV rankings. The coverage range of AB-DMOEA is mostly in the top three, indicating that the performance of AB-DMOEA is very stable. The ranking distribution of other comparison algorithms is generally relatively wide, which can be analyzed as AB-DMOEA consistently performs well on problems with diverse features, indicating a high level of competitiveness.

To better compare the performance of the seven algorithms on each test instance, we selected representative cases for each suite, with a



(a) MIGD (b) MHV

Fig. 5. The box plots of MIGD and MHV Friedman rankings of all the algorithms for all test functions in Table 1 and 2. The vertical coordinates represent the Friedman ranking values of two metrics and the data in parentheses is the average ranking of the algorithm.

total of 12 as shown in Fig. 6. In this figure, each point of the curve is the logarithm of the IGD value obtained by convergence of the population in each environment, the lower and smoother the curve, the better and more stable the performance. The proposed algorithm generates lower and smoother curves in most test instances, such as dMOP1 and DF1, indicating that AB-DMOEA can quickly and stably respond to environmental changes. In DF13, the performance of AB-DMOEA is similar to that of PBDMO, HPPCM, PCPB, and RVCP, with SGEA having excellent values in a few moments. The reason is that the size of PFs varies, and hybrid multiple strategies can better address such issues.

5. Further discussion

To verify the validity of the dynamic response strategies and static optimization mechanism in this paper, further discussion includes dynamic and static variants experiments, analysis and comparison of computational cost, performance comparison under 100 environmental changes, parameter analysis, and real-world application. The 35 test instances are consistent with those in Section 4, and the statistical results are run independently 30 times. To draw statistically sound conclusions, the Wilcoxon rank-sum test [60] and Friedman test [61] performed at the significance level of 0.05 to indicate the significance of the difference between the metric values of the two algorithms. "+", "−", and "≈" represent the performance of each variant is significantly better, worse, and similarly compared to AB-DMOEA, respectively

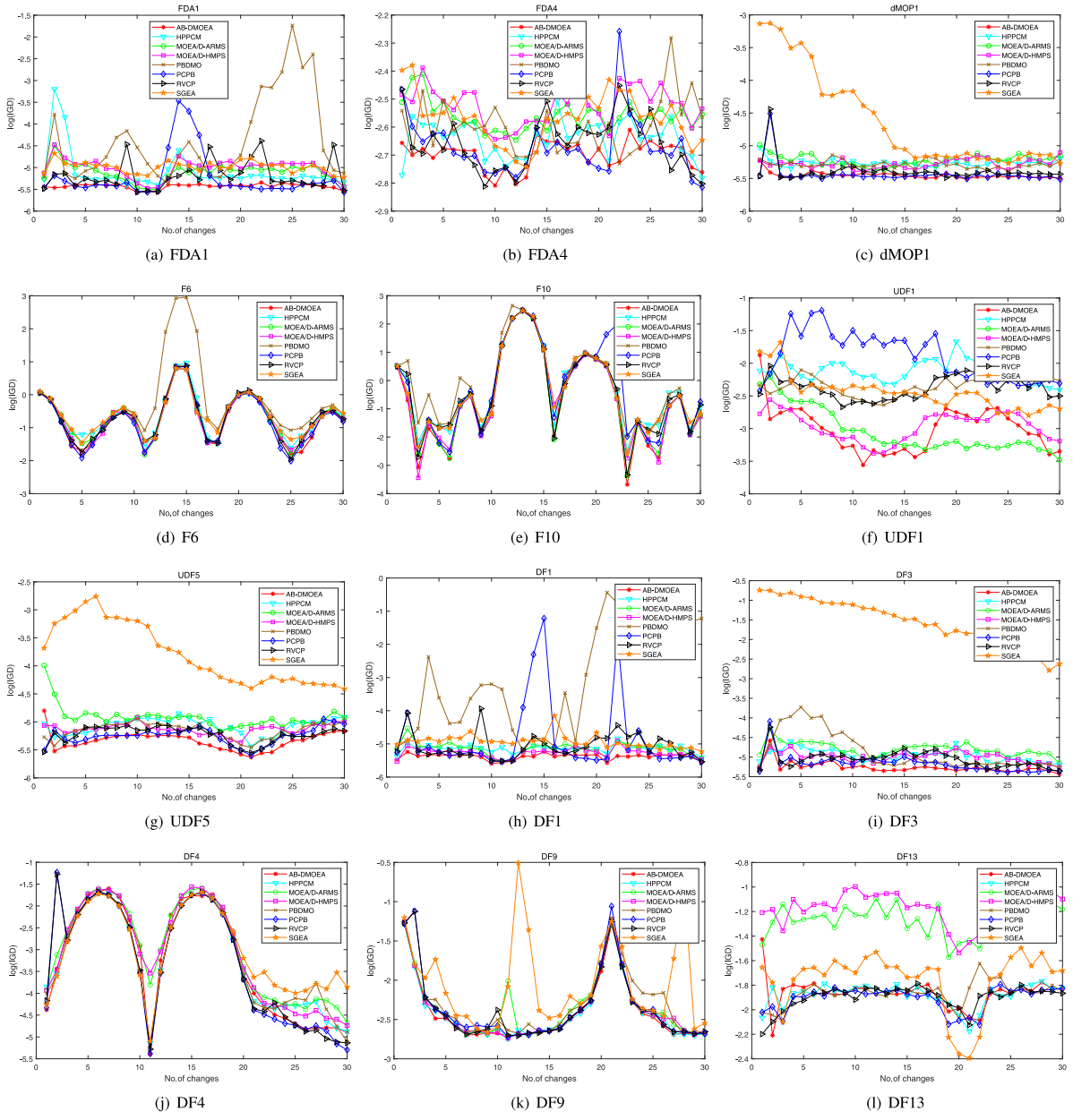
5.1. Analysis of algorithm static components

The static multi-objective optimizer is essential for solving DMOPs, and researchers have extensively studied the experimental effects of different SMOAs [13,62]. To verify the effectiveness of the static optimization boosting mechanism in static optimization, the following excellent SMOAs were placed in AB-DMOEA and compared separately as the SMOA for comparative experiments.

(1) AB-MOEA/D: MOEA/D is widely used as a classical algorithm in dynamic multi-objective evolutionary as a static multi-objective optimizer. The improved version with the addition of the DE operator has a stronger effect [14,34,51], so it is used for this variant.
(2) AB-RM-MEDA: RM-MEDA is very common as a static multi-objective optimizer [13,37,58], and its excellent performance is superior to its convenience in tracking PS_t/PF_t.
(3) AB-NSGA-II: NSGA-II has received widespread attention and application in dynamic multi-objective optimization [1] since its inception. Currently, NSGA-II is still one of the best choices for static optimization [63–65], and this variant is named AB-NSGA-II, which simulated binary crossover (SBX) is replaced by the DE operator.
(4) AB-DSS2: DSS2 is a strategy of DSS [63] that quickly converges in a short period and has received increasing attention in recent years [66–68]. Therefore, DSS2 was added as a static multi-objective optimizer in the experiment for comparison.

Tables 3 and 4 show the average of MIGD and MHV obtained by AB-DMOEA and four variants (AB-MOEA/D, AB-RM-MEDA, AB-NSGA-II, and AB-DSS2), with the best performance represented by highlights. Wilcoxon rank sum test and Friedman test results are shown in the last two rows, and the ranking results are shown in 2 decimal places. For MIGD, AB-DMOEA achieved 20 optimal results, with an average ranking of 1.66. The number of times other variants achieved better results than AB-DMOEA is $\{9, 4, 6, 0\}$, and the average rankings are $\{2.49, 2.97, 2.89, 5.00\}$. For MHV, AB-DMOEA achieved 21 optimal results, with an average ranking of 1.60. The number of times other variants achieved better results than AB-DMOEA is $\{6, 3, 3, 0\}$, and the average rankings are $\{2.34, 2.83, 3.26, 4.97\}$. Overall, the performance of AB-DMOEA is superior to the other four variants based on different SMOAs, indicating that the static optimization boosting mechanism is stable and robust. For a more intuitive comparison, the ranking of the two indicators is shown in Fig. 7.

The performance of AB-DSS2 is very poor, with almost all results in the last location. Indicating that DSS2 may need to combine special dynamic response strategies [66] to be more effective, and is not introduced in the formal AB-DMOEA. Simultaneously, the results suggest that AB-DMOEA rarely selects DSS2, implying the dependability of the static optimization boosting mechanism. It is worth noting that, the framework based on MOEA/D performs better than other static multi-objective optimizers in most cases. The average ranking results of variants AB-RM-MEDA and AB-NSGA-II are slightly worse than those of AB-MOEA/D, and they have few optimal times. The performance difference between these three static multi-objective optimizers is not significant, and each of them has a different focus. It can be seen that the average ranking of AB-DMOEA is the best and has more advantages, and the static optimization boosting mechanism accurately selects a static multi-objective optimizer that is suitable for the current environment.



(a) FDA1 (b) FDA4 (c) dMOP1

(d) F6 (e) F10 (f) UDF1

(g) UDF5 (h) DF1 (i) DF3

(j) DF4 (k) DF9 (l) DF13

Fig. 6. The average IGD convergence curves of all algorithms on representative test instances of FDA, dMOP, F, UDF, and DF.

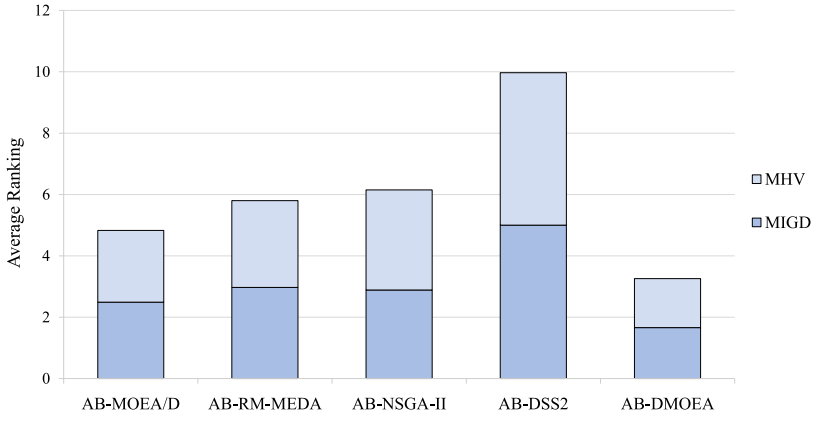


Fig. 7. Average ranking of AB-MOEA/D, AB-RM-MEDA, AB-NSGA-II, AB-DSS2, and AB-DMOEA by the Friedman test for the 35 test instances in terms of the mean value.

Table 3
Mean and standard deviation values of MIGD were obtained by AB-DMOEA and its four static variants on 35 test instances run independently 30 times. Wilcoxon rank sum test and Friedman test results are shown in the last two rows, with the best results highlighted.

Problem	M	D	AB-MOEA/D [19]	AB-RM-MEDA [20]	AB-NSGA-II [18]	AB-DSS2 [63]	AB-DMOEA
DF1	2	10	5.0508e−3 (1.03e−4) +	1.2959e−2 (8.77e−3) −	6.3708e−3 (9.07e−5) −	2.7143e−1 (2.05e−2) −	5.2933e−3 (1.39e−3)
DF2	2	10	9.8468e−3 (8.99e−4) −	5.8061e−2 (2.05e−2) −	6.8651e−3 (2.64e−4) −	2.2578e−1 (1.31e−2) −	6.4935e−3 (4.47e−4)
DF3	2	10	7.2718e−3 (1.71e−4) −	5.3371e−3 (2.31e−4) −	1.1230e−2 (2.41e−4) −	3.2491e−1 (2.80e−2) −	5.0625e−3 (1.56e−4)
DF4	2	10	8.1835e−2 (8.23e−4) −	7.3610e−2 (2.38e−2) −	7.1702e−2 (1.81e−4) −	2.4267e+0 (3.38e−1) −	6.7463e−2 (1.21e−4)
DF5	2	10	5.5424e−3 (1.80e−4) −	6.2593e−3 (4.46e−3) =	6.7046e−3 (5.75e−5) −	2.3035e−1 (2.40e−2) −	4.7704e−3 (2.75e−4)
DF6	2	10	1.9293e−1 (1.15e−1) +	1.4466e−1 (1.28e−1) +	2.1214e−1 (1.50e−1) +	2.6532e+0 (4.82e−1) −	7.8933e−1 (7.97e−1)
DF7	2	10	8.3123e−2 (3.17e−2) −	5.6566e−2 (5.18e−2) −	2.9202e−2 (3.40e−2) =	1.7853e−1 (2.31e−2) −	1.4283e−2 (1.19e−2)
DF8	2	10	1.2059e−1 (7.57e−4) +	1.3252e−1 (7.83e−3) −	1.3078e−1 (1.81e−3) −	4.0140e−1 (2.60e−2) −	1.2362e−1 (6.16e−4)
DF9	2	10	2.0800e+0 (1.85e−3) =	2.0815e+0 (3.12e−3) +	2.0802e+0 (8.08e−4) =	2.2833e+0 (2.26e−2) −	2.0817e+0 (1.39e−2)
DF10	3	10	1.5549e−1 (1.06e−2) −	1.0682e−1 (7.35e−3) =	8.2579e−2 (2.26e−3) +	4.1114e−1 (3.18e−2) −	1.0513e−1 (8.64e−3)
DF11	3	10	6.9684e−2 (3.13e−4) +	7.0593e−2 (1.15e−3) =	7.1741e−2 (4.49e−4) −	6.5751e−1 (2.71e−2) −	6.9913e−2 (9.87e−4)
DF12	3	10	2.7311e−1 (1.54e−3) −	2.5546e−1 (2.44e−3) =	2.5591e−1 (1.35e−3) =	5.4328e−1 (1.36e−2) −	2.5529e−1 (1.11e−3)
DF13	3	10	3.1165e−1 (3.86e−3) −	1.5234e−1 (3.64e−3) +	1.5226e−1 (9.95e−4) +	6.4776e−1 (3.51e−2) −	1.5602e−1 (1.91e−3)
DF14	3	10	5.3558e−2 (3.69e−4) +	1.5036e−1 (1.30e−2) =	1.3036e−1 (2.05e−2) +	3.3504e−1 (2.66e−2) −	1.5438e−1 (2.54e−2)
FDA1	2	10	5.4855e−3 (1.30e−4) −	6.1073e−3 (3.23e−3) =	6.4688e−3 (7.38e−5) −	1.7763e−1 (1.90e−2) −	4.5994e−3 (2.56e−4)
FDA2	2	10	4.8278e−3 (8.95e−5) −	7.9930e−3 (9.14e−3) −	6.3186e−3 (5.27e−5) −	2.3121e−1 (2.01e−2) −	4.4607e−3 (1.97e−5)
FDA3	2	10	3.4388e−2 (1.24e−3) +	6.7017e−2 (3.91e−3) −	6.8227e−2 (1.42e−3) −	3.4407e−1 (2.21e−2) −	5.8738e−2 (1.12e−2)
FDA4	3	10	7.4676e−2 (8.21e−4) −	6.9067e−2 (3.33e−3) =	7.2628e−2 (5.82e−4) −	6.5121e−1 (3.35e−2) −	6.8645e−2 (2.72e−3)
FDA5	3	10	1.7695e−1 (1.62e−2) −	1.5522e−1 (5.71e−3) −	1.1967e−1 (1.34e−3) +	9.3365e−1 (3.49e−2) −	1.4541e−1 (1.55e−2)
F5	2	10	8.8570e−1 (3.07e−3) =	8.9928e−1 (5.22e−3) −	9.0383e−1 (3.74e−3) −	1.6064e+0 (8.03e−2) −	8.8501e−1 (5.03e−3)
F6	2	10	6.1804e−1 (6.46e−3) −	6.2030e−1 (5.18e−3) −	6.2546e−1 (6.37e−3) −	1.1797e+0 (1.18e−1) −	6.1269e−1 (3.87e−3)
F7	2	10	7.0690e−1 (3.76e−3) −	7.1004e−1 (7.32e−3) −	7.1782e−1 (3.14e−3) −	1.7022e+0 (1.92e−1) −	7.0402e−1 (2.58e−3)
F8	3	10	9.3029e−2 (3.06e−3) −	7.2649e−2 (2.02e−3) −	8.4845e−2 (6.17e−4) −	5.4523e−1 (2.81e−2) −	7.1268e−2 (4.01e−4)
F9	2	10	3.2321e−1 (2.22e−2) =	4.0580e−1 (9.93e−2) −	3.3379e−1 (1.20e−2) −	8.4798e−1 (9.03e−2) −	3.2175e−1 (1.21e−2)
F10	2	10	1.7192e+0 (1.18e−2) +	1.7675e+0 (3.85e−2) −	1.7704e+0 (9.10e−3) −	3.3428e+0 (1.31e−1) −	1.7356e+0 (1.93e−2)
dMOP1	2	10	5.4680e−3 (6.72e−4) −	6.3934e−3 (6.30e−3) −	6.4655e−3 (1.44e−4) −	8.7398e−1 (1.80e−1) −	4.2535e−3 (2.89e−5)
dMOP2	2	10	5.6048e−3 (1.10e−4) −	6.5195e−3 (4.92e−3) −	6.7373e−3 (6.27e−5) −	2.1988e−1 (2.35e−2) −	4.5960e−3 (1.55e−4)
dMOP3	2	10	5.1638e−3 (3.55e−4) +	8.1586e−3 (4.15e−3) −	6.1122e−3 (4.97e−5) −	2.1314e−1 (1.70e−2) −	5.7038e−3 (2.22e−3)
UDF1	2	10	9.5887e−2 (1.14e−2) −	1.2321e−1 (4.26e−3) −	8.8975e−2 (4.32e−3) −	5.2328e−1 (2.25e−2) −	3.9462e−2 (6.35e−3)
UDF2	2	10	6.1670e−3 (9.37e−5) −	5.1558e−3 (2.22e−4) −	8.9103e−3 (9.24e−5) −	1.9688e−1 (1.47e−2) −	4.8163e−3 (3.06e−5)
UDF3	2	10	6.3288e−1 (4.95e−2) +	1.0226e+0 (2.41e−1) =	5.3793e−1 (5.66e−2) +	3.7346e+0 (2.47e+0) −	8.2702e−1 (2.11e−1)
UDF4	2	10	1.5299e−1 (7.08e−3) −	1.6925e−1 (4.91e−3) −	1.2816e−1 (7.21e−3) −	3.8281e−1 (1.02e−2) −	9.0298e−2 (2.02e−2)
UDF5	2	10	6.4567e−3 (2.28e−4) −	5.1944e−3 (9.19e−4) +	7.5608e−3 (8.36e−5) −	1.4073e−1 (1.06e−2) −	5.2295e−3 (8.44e−4)
UDF6	2	10	8.3795e−1 (4.92e−2) −	9.8945e−1 (9.76e−2) −	7.1799e−1 (3.07e−2) =	1.9393e+0 (7.39e−1) −	7.2722e−1 (1.06e−1)
UDF7	3	10	8.4316e−1 (1.15e−2) −	7.4911e−1 (1.11e−2) =	7.8889e−1 (1.96e−2) −	1.3428e+0 (2.40e−2) −	7.4474e−1 (1.52e−2)
+/−/=			9/23/3	4/22/9	6/25/4	0/35/0	
Average ranking			2.49	2.97	2.89	5.00	1.66

Each of AB-MOEA/D, AB-RM-MEDA, AB-NSGA-II, and AB-DSS2 is a variant with different static multi-objective optimizers. "+", "−", and "≈" in the Wilcoxon test represent the performance of each variant is significantly better, worse, and similarly compared to AB-DMOEA, respectively.

5.2. Analysis of algorithm dynamic components

Designing a robust environmental change response mechanism is the main task for a DMOEA. To examine the effectiveness of this part of AB-DMOEA, this subsection will conduct the following variant experiments.

(1) AB-V1: This variant replaces the adaptive boosting response mechanism with the prediction boosting strategy in Algorithm 4, mainly verifying its effectiveness.
(2) AB-V2: The version of the proposed memory boosting strategy in Algorithm 6, paired with the dominated solution reinforcement strategy to respond to environmental changes.
(3) AB-V3: In the adaptive boosting response mechanism, only the diversity boosting strategy proposed in Algorithm 7 is retained to measure its effectiveness.
(4) AB-V4: Compared to AB-DMOEA, this variant does not use the dominated solution reinforcement strategy in Algorithm 2 to validate the effectiveness of this part.

Tables 5 and 6 show the average of MIGD and MHV obtained by AB-DMOEA and four variants (AB-V1, AB-V2, AB-V3, and AB-V4), with the best performance represented by highlights. For MIGD, AB-DMOEA achieved 21 optimal results, with an average ranking of 2.26. The number of times other variants achieved better results than AB-DMOEA is $\{8, 5, 4, 3\}$, and the average ranking is $\{2.96, 3.67, 3.66, 2.46\}$. For MHV, AB-DMOEA achieved 19 optimal results, with an average ranking of 2.20. The number of times other variants achieved better results than AB-DMOEA is $\{6, 7, 2, 4\}$, and the average rankings are $\{3.17, 3.24, 3.79, 2.60\}$. For a more intuitive comparison, the ranking of the two indicators is shown in Fig. 8.

The results indicate that the ranking of the main components of the dynamic response strategy (V1-V3) is close, and their performance tends to be similar. In the overall ranking of Fig. 8, The optimal frequency of AB-V1 is significantly higher than that of AB-V2 and AB-V3, and the average ranking is also higher. The performance of AB-V2 and AB-V3 is similar in terms of MIGD, MHV, and average rankings. The average ranking between AB-V4 and AB-DMOEA is close and there are many similar results, due to the dominated individual boosting mechanism mainly playing a role in complex environmental changes and a large number of dominated solutions. These types of problems are relatively few, the first three strategies are sufficient to address most of them. AB-DMOEA has much more optimal rankings than this variant indicating that the strategy is effective.

From the above analysis, it is clear that the three main components of the AB-DMOEA significantly improve the overall performance of the algorithm. Three effective dynamic response strategies have been strengthened through the adaptive boosting response mechanism to

Table 4
Mean and standard deviation values of MHV were obtained by AB-DMOEA and its four static variants on 35 test instances run independently 30 times. Wilcoxon rank sum test and Friedman test results are shown in the last two rows, with the best results highlighted.

Problem	M	D	AB-MOEA/D [19]	AB-RM-MEDA [20]	AB-NSGA-II [18]	AB-DSS2 [63]	AB-DMOEA
DF1	2	10	5.2432e−1 (1.99e−4) −	5.1632e−1 (7.78e−3) −	5.2279e−1 (1.64e−4) −	2.8461e−1 (1.35e−2) −	5.2441e−1 (1.73e−3)
DF2	2	10	7.0928e−1 (1.51e−3) −	6.5297e−1 (2.25e−2) −	7.1443e−1 (5.35e−4) −	4.6408e−1 (1.70e−2) −	7.1497e−1 (6.95e−4)
DF3	2	10	4.8737e−1 (3.47e−4) −	4.9159e−1 (3.56e−4) −	4.8284e−1 (3.88e−4) −	2.2044e−1 (1.18e−2) −	4.9194e−1 (4.03e−4)
DF4	2	10	2.8437e−1 (1.06e−4) −	2.8356e−1 (6.87e−3) −	2.8432e−1 (6.61e−5) −	3.8136e−2 (1.35e−2) −	2.8532e−1 (2.25e−5)
DF5	2	10	1.2842e−1 (1.80e−4) −	1.2768e−1 (2.66e−3) =	1.2824e−1 (3.11e−5) −	7.5020e−2 (3.28e−3) −	1.2870e−1 (1.85e−4)
DF6	2	10	6.0369e−1 (7.08e−2) +	6.4356e−1 (7.04e−2) +	5.9716e−1 (4.28e−2) +	1.4543e−1 (3.68e−2) −	4.6962e−1 (1.44e−1)
DF7	2	10	1.4561e−1 (3.31e−3) −	1.4735e−1 (6.17e−3) −	1.5127e−1 (3.57e−3) −	1.2849e−1 (5.60e−3) −	1.5287e−1 (1.30e−3)
DF8	2	10	8.1788e−2 (1.21e−4) =	8.0422e−2 (1.42e−3) −	8.1092e−2 (1.56e−4) −	3.9651e−2 (2.60e−3) −	8.1752e−2 (9.33e−5)
DF9	2	10	3.4689e−2 (7.99e−4) −	3.4473e−2 (1.11e−3) −	3.4458e−2 (5.09e−4) −	6.6892e−3 (9.61e−4) −	3.5232e−2 (1.37e−3)
DF10	3	10	5.9853e−1 (8.42e−3) +	5.8615e−1 (1.09e−2) =	6.1043e−1 (4.43e−3) +	3.3060e−1 (1.84e−2) −	5.8542e−1 (1.14e−2)
DF11	3	10	2.9228e−1 (6.45e−4) −	2.9230e−1 (1.30e−3) −	2.8977e−1 (7.39e−4) −	7.2220e−2 (5.58e−3) −	2.9393e−1 (8.55e−4)
DF12	3	10	9.0756e−1 (6.73e−4) +	9.0649e−1 (1.21e−3) −	9.0371e−1 (3.70e−4) −	7.4021e−1 (8.28e−3) −	9.0709e−1 (8.79e−4)
DF13	3	10	5.7200e−1 (4.71e−3) −	6.3081e−1 (3.69e−3) +	6.2874e−1 (1.39e−3) =	3.5948e−1 (1.20e−2) −	6.2888e−1 (2.08e−3)
DF14	3	10	5.8773e−1 (5.50e−4) +	4.3135e−1 (1.04e−2) =	4.7686e−1 (6.92e−3) +	2.5256e−1 (1.36e−2) −	4.3596e−1 (2.40e−2)
FDA1	2	10	7.1654e−1 (1.96e−4) −	7.1693e−1 (2.73e−3) =	7.1535e−1 (1.21e−4) −	5.0833e−1 (1.45e−2) −	7.1827e−1 (4.38e−4)
FDA2	2	10	5.2952e−1 (1.33e−4) −	5.2756e−1 (7.06e−3) −	5.2776e−1 (8.14e−5) −	3.0048e−1 (1.59e−2) −	5.3035e−1 (4.99e−5)
FDA3	2	10	5.6201e−1 (1.13e−3) +	5.3346e−1 (3.40e−3) −	5.3202e−1 (1.37e−3) −	2.8370e−1 (1.71e−2) −	5.4028e−1 (1.03e−2)
FDA4	3	10	5.2405e−1 (1.37e−3) =	5.2136e−1 (4.67e−3) =	5.1356e−1 (1.38e−3) −	1.2888e−1 (1.32e−2) −	5.2203e−1 (5.27e−3)
FDA5	3	10	5.2232e−1 (2.67e−3) =	5.1630e−1 (3.95e−3) =	5.0963e−1 (1.97e−3) −	1.5071e−1 (8.23e−3) −	5.1859e−1 (6.86e−3)
F5	2	10	3.2946e−1 (7.55e−4) =	3.3071e−1 (1.37e−3) +	3.2881e−1 (7.16e−4) =	2.2218e−1 (9.15e−3) −	3.2784e−1 (4.30e−3)
F6	2	10	2.6315e−1 (8.25e−4) −	2.6486e−1 (9.53e−4) =	2.6389e−1 (6.38e−4) −	1.7488e−1 (1.17e−2) −	2.6503e−1 (2.06e−3)
F7	2	10	2.3593e−1 (8.35e−4) −	2.3085e−1 (5.45e−3) −	2.2788e−1 (1.46e−3) −	4.5991e−2 (4.08e−3) −	2.3708e−1 (1.47e−3)
F8	3	10	4.8261e−1 (4.54e−3) −	5.0897e−1 (3.14e−3) −	4.8444e−1 (1.18e−3) −	1.3593e−1 (1.10e−2) −	5.1136e−1 (1.05e−3)
F9	2	10	3.9400e−1 (8.16e−4) =	3.9567e−1 (2.21e−3) =	3.9182e−1 (1.14e−3) −	2.5508e−1 (1.14e−2) −	3.9421e−1 (3.63e−3)
F10	2	10	3.0785e−1 (3.15e−3) =	2.8858e−1 (1.12e−2) −	2.8638e−1 (2.80e−3) −	3.8022e−2 (7.17e−3) −	3.0597e−1 (9.51e−3)
dMOP1	2	10	5.2365e−1 (9.54e−4) −	5.2414e−1 (5.37e−3) −	5.2248e−1 (2.35e−4) −	1.1363e−1 (8.54e−3) −	5.2627e−1 (6.37e−5)
dMOP2	2	10	5.2330e−1 (2.03e−4) −	5.2397e−1 (3.87e−3) =	5.2213e−1 (1.11e−4) −	3.1761e−1 (7.61e−3) −	5.2549e−1 (3.76e−4)
dMOP3	2	10	7.1717e−1 (5.63e−4) +	7.1479e−1 (3.14e−3) −	7.1602e−1 (7.75e−5) −	4.7737e−1 (2.13e−2) −	7.1676e−1 (3.03e−3)
UDF1	2	10	2.3265e−1 (4.65e−3) −	2.1324e−1 (3.05e−3) −	2.3432e−1 (3.19e−3) −	4.4073e−2 (6.71e−3) −	2.6588e−1 (4.81e−3)
UDF2	2	10	2.9203e−1 (8.83e−5) −	2.9303e−1 (2.24e−4) −	2.9042e−1 (8.40e−5) −	1.8398e−1 (4.87e−3) −	2.9341e−1 (4.11e−5)
UDF3	2	10	3.3608e−1 (1.37e−2) −	5.6706e−2 (2.66e−2) −	1.5735e−1 (1.74e−2) −	8.0147e−2 (1.31e−1) −	3.8202e−1 (8.60e−2)
UDF4	2	10	3.7602e−1 (7.13e−3) −	3.1642e−1 (4.71e−3) −	3.6478e−1 (4.63e−3) −	1.5284e−1 (7.27e−3) −	4.1584e−1 (2.96e−2)
UDF5	2	10	5.1129e−1 (2.22e−4) −	5.1345e−1 (9.36e−4) =	5.0996e−1 (1.30e−4) −	3.7634e−1 (6.35e−3) −	5.1330e−1 (1.22e−3)
UDF6	2	10	7.8823e−2 (5.31e−3) −	3.6616e−2 (1.93e−2) −	7.4234e−2 (4.37e−3) −	2.9614e−2 (4.25e−2) −	8.8449e−2 (1.36e−2)
UDF7	3	10	1.6934e−1 (3.30e−3) −	1.7667e−1 (1.92e−3) =	1.7474e−1 (1.85e−3) −	4.6820e−2 (2.08e−3) −	1.7696e−1 (2.08e−3)
+/−/≈			6/23/6	3/21/11	3/30/2	0/35/0	
Average ranking			2.34	2.83	3.26	4.97	1.60

Each of AB-MOEA/D, AB-RM-MEDA, AB-NSGA-II, and AB-DSS2 is a variant with different static multi-objective optimizers. "+", "−", and "≈" in the Wilcoxon test represent the performance of each variant is significantly better, worse, and similarly compared to AB-DMOEA, respectively.

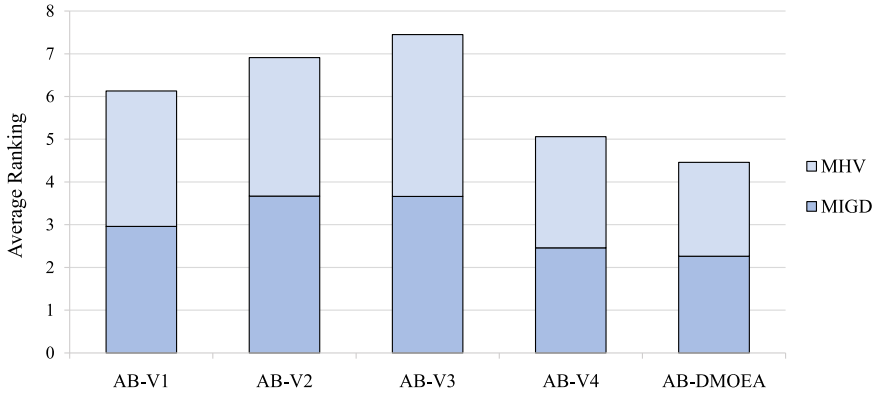


Fig. 8. Average ranking of AB-V1, AB-V2, AB-V3, AB-V4, and AB-DMOEA by the Friedman test for the 35 test instances in terms of the mean value.

track PS_t/PF_t more accurately. The dominated solution reinforcement mechanism generates a reliable population in extreme environments, which helps further optimize and increase the effectiveness of the proposed mechanism. The static optimization boosting mechanism will select the best among the three robust SMOAs to ensure static optimization.

5.3. Analysis and testing of computational cost

To demonstrate the computational resources of the compared algorithms, the computational complexity of each optimization algorithm and AB-DMOEA are analyzed as follows.

Table 5
Mean and standard deviation values of MIGD were obtained by AB-DMOEA and its four dynamic variants on 35 test instances run independently 30 times. Wilcoxon rank sum test and Friedman test results are shown in the last two rows, with the best results highlighted.

Problem	M	D	AB-V1	AB-V2	AB-V3	AB-V4	AB-DMOEA
DF1	2	10	4.8535e−3 (7.04e−5) +	5.1891e−3 (7.20e−4) +	4.8512e−3 (1.40e−4) +	5.2220e−3 (1.28e−3) ≈	5.6372e−3 (1.91e−3)
DF2	2	10	7.0716e−3 (2.84e−4) −	7.3109e−3 (3.21e−3) ≈	7.2359e−3 (2.35e−3) ≈	7.1835e−3 (2.55e−3) −	6.5142e−3 (4.90e−4)
DF3	2	10	6.4192e−3 (1.57e−4) −	6.7302e−3 (7.96e−4) −	7.1109e−3 (1.03e−3) −	5.0950e−3 (1.63e−4) −	5.0240e−3 (4.69e−5)
DF4	2	10	8.1503e−2 (6.40e−4) −	7.1560e−2 (5.55e−3) −	7.0201e−2 (4.18e−3) −	6.9077e−2 (4.28e−3) ≈	6.7600e−2 (2.20e−4)
DF5	2	10	5.2188e−3 (7.47e−5) −	5.5855e−3 (3.07e−4) −	5.3110e−3 (2.76e−4) −	4.8840e−3 (1.52e−4) −	4.7283e−3 (1.11e−4)
DF6	2	10	8.4220e−1 (7.04e−1) ≈	4.6548e−1 (3.91e−1) ≈	4.3673e−1 (2.91e−1) ≈	3.7764e−1 (2.63e−1) ≈	6.1862e−1 (5.27e−1)
DF7	2	10	2.3835e−2 (2.16e−2) ≈	2.2154e−2 (2.25e−2) ≈	6.0882e−2 (3.33e−2) −	2.0981e−2 (1.76e−2) ≈	2.4692e−2 (2.15e−2)
DF8	2	10	1.1991e−1 (3.14e−4) +	1.3049e−1 (6.84e−3) −	1.3188e−1 (7.66e−3) −	1.2262e−1 (2.38e−3) ≈	1.2345e−1 (1.21e−3)
DF9	2	10	2.0775e+0 (1.47e−3) ≈	2.0780e+0 (1.07e−3) ≈	2.0778e+0 (1.07e−3) ≈	2.0782e+0 (1.81e−3) ≈	2.0843e+0 (2.54e−2)
DF10	3	10	1.5359e−1 (9.10e−3) −	1.3618e−1 (2.34e−2) −	1.5277e−1 (2.41e−2) −	1.1899e−1 (2.83e−2) ≈	1.0794e−1 (1.22e−2)
DF11	3	10	6.8138e−2 (2.25e−4) +	6.9632e−2 (8.95e−4) ≈	6.9010e−2 (8.04e−4) +	6.7149e−2 (1.22e−3) +	6.9849e−2 (4.85e−4)
DF12	3	10	2.6969e−1 (3.63e−4) −	2.6245e−1 (7.86e−3) −	2.6549e−1 (9.50e−3) −	2.6024e−1 (6.44e−3) −	2.5492e−1 (1.15e−3)
DF13	3	10	3.2105e−1 (4.69e−3) −	2.9854e−1 (6.33e−2) −	3.0837e−1 (5.40e−2) −	2.7914e−1 (7.70e−2) −	1.5522e−1 (1.43e−3)
DF14	3	10	5.2673e−2 (2.34e−4) +	7.6504e−2 (4.57e−2) +	8.2362e−2 (4.94e−2) +	8.3265e−2 (4.71e−2) +	1.4738e−1 (2.51e−2)
FDA1	2	10	5.1169e−3 (9.29e−5) −	5.4991e−3 (8.45e−5) −	5.0109e−3 (4.00e−4) −	4.6177e−3 (2.11e−4) ≈	4.5323e−3 (1.52e−4)
FDA2	2	10	4.7937e−3 (4.98e−5) −	5.5338e−3 (2.51e−4) −	5.3273e−3 (2.28e−4) −	4.5778e−3 (6.00e−5) −	4.4585e−3 (1.80e−5)
FDA3	2	10	9.7839e−3 (4.98e−4) +	3.0015e−2 (6.13e−3) +	2.8502e−2 (1.18e−3) +	4.0063e−2 (1.77e−2) +	6.0709e−2 (9.15e−3)
FDA4	3	10	7.2511e−2 (3.27e−4) −	7.1426e−2 (6.84e−4) −	7.0316e−2 (1.86e−3) ≈	7.0105e−2 (1.67e−3) ≈	6.9327e−2 (2.66e−3)
FDA5	3	10	1.8290e−1 (3.69e−2) −	1.4433e−1 (7.68e−3) ≈	1.4622e−1 (8.49e−3) ≈	1.5749e−1 (1.88e−2) −	1.4323e−1 (1.34e−2)
F5	2	10	8.8158e−1 (2.80e−3) +	8.9290e−1 (2.74e−2) −	8.9146e−1 (6.44e−3) ≈	8.8802e−1 (6.35e−3) ≈	8.8912e−1 (6.98e−3)
F6	2	10	6.1573e−1 (4.37e−3) ≈	6.1550e−1 (9.16e−3) ≈	6.1453e−1 (4.78e−3) ≈	6.1600e−1 (7.64e−3) ≈	6.1372e−1 (3.52e−3)
F7	2	10	7.0293e−1 (2.43e−3) ≈	7.0427e−1 (2.66e−3) ≈	7.0473e−1 (3.06e−3) ≈	7.0341e−1 (2.74e−3) ≈	7.0314e−1 (2.77e−3)
F8	3	10	7.9618e−2 (7.76e−4) −	7.8428e−2 (2.91e−3) −	8.0090e−2 (3.39e−3) −	7.1452e−2 (1.31e−3) ≈	7.0913e−2 (6.21e−4)
F9	2	10	3.1558e−1 (1.02e−2) ≈	3.2024e−1 (2.26e−2) ≈	3.1914e−1 (1.22e−2) ≈	3.1417e−1 (4.96e−3) ≈	3.2356e−1 (2.08e−2)
F10	2	10	1.7072e+0 (9.83e−3) +	1.7242e+0 (2.42e−2) +	1.7276e+0 (1.76e−2) ≈	1.7307e+0 (1.47e−2) ≈	1.7386e+0 (2.43e−2)
dMOP1	2	10	4.5869e−3 (5.62e−5) −	5.5460e−3 (1.35e−3) −	6.0399e−3 (1.59e−3) −	4.2180e−3 (5.93e−5) ≈	4.2384e−3 (3.84e−5)
dMOP2	2	10	5.3111e−3 (8.29e−5) −	5.8194e−3 (2.50e−4) −	5.2599e−3 (4.93e−4) −	4.8261e−3 (3.16e−4) −	4.6587e−3 (1.82e−4)
dMOP3	2	10	4.8385e−3 (5.88e−5) +	4.9912e−3 (2.18e−4) +	4.7660e−3 (1.62e−4) ≈	4.7916e−3 (4.39e−4) ≈	5.3008e−3 (2.73e−3)
UDF1	2	10	4.8119e−2 (2.08e−2) −	9.3491e−2 (1.38e−2) −	9.6153e−2 (1.28e−2) −	5.3480e−2 (2.19e−2) ≈	3.8828e−2 (4.96e−3)
UDF2	2	10	5.4250e−3 (8.58e−5) −	6.1787e−3 (4.09e−4) −	6.1801e−3 (4.64e−4) −	4.8514e−3 (6.52e−5) ≈	4.8384e−3 (4.32e−5)
UDF3	2	10	8.7009e−1 (1.23e−1) ≈	8.8280e−1 (1.69e−1) ≈	9.3406e−1 (1.25e−1) −	8.4241e−1 (1.66e−1) ≈	7.7964e−1 (1.76e−1)
UDF4	2	10	9.9241e−2 (1.50e−2) −	1.3819e−1 (1.00e−2) −	1.3539e−1 (1.03e−2) −	9.9639e−2 (1.68e−2) −	8.1010e−2 (1.27e−2)
UDF5	2	10	5.6889e−3 (6.32e−5) −	6.4637e−3 (7.69e−4) −	6.5242e−3 (6.59e−4) −	5.0597e−3 (3.09e−4) ≈	4.9300e−3 (1.60e−4)
UDF6	2	10	7.7926e−1 (1.43e−1) ≈	7.2786e−1 (1.27e−1) ≈	7.7305e−1 (1.12e−1) ≈	7.5851e−1 (1.30e−1) ≈	7.2161e−1 (8.38e−2)
UDF7	3	10	7.6580e−1 (4.79e−2) ≈	8.1803e−1 (3.01e−2) −	8.4698e−1 (2.41e−2) −	7.7860e−1 (4.71e−2) ≈	7.4696e−1 (2.02e−2)
+/−/≈			8/18/9	5/19/11	4/19/12	3/9/23	
Average ranking			2.96	3.67	3.66	2.46	2.26

Each of AB-V1, AB-V2, AB-V3, and AB-V4 is a variant of AB-DMOEA. "+", "−", and "≈" in the Wilcoxon test represent the performance of each variant is significantly better, worse, and similarly compared to AB-DMOEA, respectively.

Set M, N, and D to be the dimensions of the objective space, population size, and the dimensions of decision variables, respectively.

MOEA/D-HMPS: The computational complexity of MOEA/D-HMPS is mainly determined by three parts: change identification, memory strategy, and change response. The change identification in MOEA/D-HMPS requires $O(MD)$ computational complexity used to determine the similarity of the environment. The memory strategy needs $O(MN^2)$ computational complexity, while the change response takes $O(DN)$ computational complexity. Therefore, the overall computational complexity is $O(MN^2)$.

MOEA/D-ARMS: The computational complexity of MOEA/D-ARMS is mainly determined by three parts, reward assignment, response mechanism selection, and response mechanism pool. The computational complexity of the first two is $O(N)$, and the worst-case response computational complexity of the mechanism pool is $O(MN^2)$, which integrates four strategies introduced. As a result, the overall computational complexity is $O(MN^2)$.

The computational complexity of other comparative algorithms is analyzed in detail in [40,59,69], and AB-DMOEA is analyzed in Section 3.5. The running time of all the compared algorithms is recorded when solving the 35 instances under the parameter and dynamic settings given in Tables 1 and 2. The computational complexity of all algorithms in the experiment, the average time over all the test functions, and the total running time of each algorithm are shown in Table 7. For a more intuitive comparison, the runtime of all algorithms in experiments is shown in Fig. 9.

It can be seen that the average running time of the proposed AB-DMOEA is at a moderately lower level, which indicates that the proposed method will not bring unbearable computational costs to the optimization process. Although PBDMO and HPPCM have advantages in computational cost, their optimization results are not as well as AB-DMOEA. The running time of PCPB and RVCP is slightly longer than that of AB-DMOEA, while MOEA/D-HMPS and MOEA/D-ARMS take more time than AB-DMOEA to solve these functions. At the same time, this is affected by the static optimization process, and the running time of RM-MEDA is less than MOEA/D. SGEA has a significantly higher running time than other comparison algorithms due to its unique processing mechanism.

5.4. Performance comparison under 100 environmental changes

In Section 4, all algorithms optimized only 30 environmental changes for each test instance. Based on the original experiment, this section sets up 100 environmental changes to comprehensively analyze the performance of each algorithm in more environments. The partial results of the experiment are shown in Fig. 10.

For most test instances, AB-DMOEA achieved excellent results and stayed at the leading level for a long time, such as dMOP1, F8, DF1, etc.

Table 6
Mean and standard deviation values of MHV were obtained by AB-DMOEA and its four dynamic variants on 35 test instances run independently 30 times. Wilcoxon rank sum test and Friedman test results are shown in the last two rows, with the best results highlighted.

Problem	M	D	AB-V1	AB-V2	AB-V3	AB-V4	AB-DMOEA
DF1	2	10	5.2446e−1 (1.34e−3) +	5.2421e−1 (8.48e−4) +	5.2484e−1 (4.23e−4) ≈	5.2457e−1 (1.58e−3) ≈	5.2405e−1 (2.41e−3)
DF2	2	10	7.1362e−1 (1.31e−3) −	7.1446e−1 (3.43e−3) ≈	7.1367e−1 (3.44e−3) ≈	7.1392e−1 (3.89e−3) −	7.1493e−1 (7.81e−4)
DF3	2	10	4.8805e−1 (2.77e−3) −	4.8876e−1 (1.71e−3) −	4.8815e−1 (2.08e−3) −	4.9173e−1 (5.93e−4) −	4.9204e−1 (7.82e−5)
DF4	2	10	2.8453e−1 (3.88e−4) −	2.8490e−1 (2.75e−4) −	2.8489e−1 (2.85e−4) −	2.8523e−1 (1.76e−4) ≈	2.8531e−1 (2.25e−5)
DF5	2	10	1.2845e−1 (2.74e−4) −	1.2844e−1 (3.77e−4) −	1.2861e−1 (8.57e−5) −	1.2866e−1 (7.33e−5) −	1.2873e−1 (6.66e−5)
DF6	2	10	4.8051e−1 (1.91e−1) ≈	5.9161e−1 (4.11e−2) ≈	5.6745e−1 (5.58e−2) ≈	5.5246e−1 (7.40e−2) ≈	4.9921e−1 (1.44e−1)
DF7	2	10	1.5144e−1 (2.79e−3) ≈	1.5187e−1 (2.39e−3) ≈	1.4777e−1 (3.85e−3) −	1.5206e−1 (2.03e−3) ≈	1.5162e−1 (2.41e−3)
DF8	2	10	8.1756e−2 (2.99e−4) −	8.1234e−2 (3.96e−4) −	8.1244e−2 (4.30e−4) −	8.1746e−2 (2.23e−4) ≈	8.1769e−2 (8.97e−5)
DF9	2	10	3.4928e−2 (1.36e−3) −	3.5237e−2 (2.92e−4) −	3.5112e−2 (3.82e−4) −	3.5133e−2 (1.37e−3) −	1.0480e−1 (6.89e−4)
DF10	3	10	6.1695e−1 (2.41e−2) +	5.8389e−1 (9.23e−3) ≈	5.7996e−1 (1.15e−2) ≈	6.1785e−1 (1.89e−2) +	5.8032e−1 (1.41e−2)
DF11	3	10	2.9248e−1 (2.13e−3) −	2.9267e−1 (8.78e−4) −	2.9222e−1 (8.00e−4) −	2.9376e−1 (4.41e−4) ≈	2.9403e−1 (5.45e−4)
DF12	3	10	9.0964e−1 (3.07e−3) +	9.0662e−1 (5.20e−4) −	9.0675e−1 (5.37e−4) −	9.0930e−1 (8.68e−4) +	9.0859e−1 (7.42e−4)
DF13	3	10	5.6775e−1 (8.76e−3) −	5.6940e−1 (2.99e−2) −	5.5926e−1 (1.68e−2) −	5.9277e−1 (2.24e−2) −	6.3038e−1 (2.28e−3)
DF14	3	10	5.8577e−1 (1.76e−3) +	5.5645e−1 (7.49e−2) +	5.4746e−1 (8.55e−2) +	5.4511e−1 (8.24e−2) +	5.2959e−1 (7.14e−2)
FDA1	2	10	7.1494e−1 (1.92e−3) −	7.1620e−1 (1.52e−4) −	7.1650e−1 (3.76e−4) −	7.1733e−1 (2.23e−4) −	7.1838e−1 (3.22e−4)
FDA2	2	10	5.2817e−1 (1.27e−3) −	5.2799e−1 (5.75e−4) −	5.2799e−1 (6.64e−4) −	5.2982e−1 (1.24e−4) −	5.3035e−1 (3.95e−5)
FDA3	2	10	5.8276e−1 (4.13e−3) +	5.6407e−1 (1.23e−2) +	5.5599e−1 (1.75e−2) ≈	5.5137e−1 (1.83e−2) ≈	5.3840e−1 (8.00e−3)
FDA4	3	10	5.1841e−1 (7.15e−3) ≈	5.2091e−1 (3.42e−3) ≈	5.2225e−1 (2.75e−3) ≈	5.2645e−1 (3.25e−3) +	5.1996e−1 (5.38e−3)
FDA5	3	10	5.1687e−1 (8.06e−3) ≈	5.2206e−1 (6.58e−3) +	5.2139e−1 (7.84e−3) ≈	5.2295e−1 (5.04e−3) ≈	5.1991e−1 (5.28e−3)
F5	2	10	3.2973e−1 (2.02e−3) ≈	3.2942e−1 (3.62e−3) +	3.2514e−1 (4.24e−3) −	3.2543e−1 (7.97e−3) ≈	3.2742e−1 (7.03e−3)
F6	2	10	2.6384e−1 (1.86e−3) −	2.6278e−1 (3.13e−3) −	2.6304e−1 (2.45e−3) −	2.6136e−1 (6.58e−3) ≈	2.6386e−1 (4.03e−3)
F7	2	10	2.3728e−1 (1.67e−3) ≈	2.3439e−1 (3.92e−3) ≈	2.3351e−1 (4.43e−3) ≈	2.3442e−1 (2.95e−3) −	2.3734e−1 (1.45e−3)
F8	3	10	4.8351e−1 (1.79e−2) −	4.9064e−1 (2.51e−3) −	4.9029e−1 (2.41e−3) −	5.0843e−1 (4.05e−3) −	5.1199e−1 (1.11e−3)
F9	2	10	3.9387e−1 (3.00e−3) ≈	3.9484e−1 (4.88e−3) ≈	3.8902e−1 (5.79e−3) −	3.9009e−1 (1.11e−2) ≈	3.9475e−1 (3.64e−3)
F10	2	10	3.0604e−1 (5.43e−3) ≈	3.0768e−1 (9.98e−3) +	3.0531e−1 (1.26e−2) +	2.9951e−1 (1.53e−2) ≈	3.0342e−1 (1.42e−2)
dMOP1	2	10	5.2476e−1 (2.37e−3) −	5.2350e−1 (2.39e−3) −	5.2259e−1 (2.79e−3) −	5.2610e−1 (5.14e−4) ≈	5.2627e−1 (6.48e−5)
dMOP2	2	10	5.2334e−1 (1.80e−3) −	5.2304e−1 (2.40e−4) −	5.2419e−1 (9.89e−4) −	5.2491e−1 (6.57e−4) −	5.2539e−1 (4.48e−4)
dMOP3	2	10	7.1742e−1 (1.21e−3) +	7.1745e−1 (2.39e−4) +	7.1782e−1 (3.83e−4) ≈	7.1792e−1 (5.07e−4) ≈	7.1733e−1 (3.44e−3)
UDF1	2	10	2.5842e−1 (1.21e−2) −	2.3611e−1 (6.94e−3) −	2.3254e−1 (7.32e−3) −	2.5782e−1 (1.23e−2) ≈	2.6636e−1 (3.82e−3)
UDF2	2	10	2.9245e−1 (9.04e−4) −	2.9234e−1 (4.10e−4) −	2.9240e−1 (4.27e−4) −	2.9329e−1 (1.88e−4) ≈	2.9338e−1 (7.36e−5)
UDF3	2	10	4.2007e−1 (4.30e−2) ≈	3.8081e−1 (8.60e−2) ≈	3.9066e−1 (5.98e−2) ≈	3.2981e−1 (1.24e−1) ≈	3.9967e−1 (7.51e−2)
UDF4	2	10	4.2355e−1 (2.14e−2) ≈	3.7078e−1 (2.60e−2) −	3.8383e−1 (3.21e−2) −	4.0130e−1 (2.69e−2) −	4.2768e−1 (1.30e−2)
UDF5	2	10	5.1213e−1 (1.48e−3) −	5.1120e−1 (1.10e−3) −	5.1117e−1 (1.02e−3) −	5.1349e−1 (5.40e−4) ≈	5.1374e−1 (2.71e−4)
UDF6	2	10	8.8210e−2 (1.76e−2) ≈	9.2343e−2 (1.77e−2) ≈	8.6442e−2 (1.49e−2) ≈	8.6412e−2 (1.86e−2) ≈	8.7691e−2 (1.91e−2)
UDF7	3	10	1.7675e−1 (4.88e−3) ≈	1.7251e−1 (2.03e−3) −	1.7255e−1 (2.41e−3) −	1.7741e−1 (2.48e−3) ≈	1.7694e−1 (2.48e−3)
+/−/≈			6/17/12	7/19/9	2/22/11	4/11/20	
Average ranking			3.17	3.24	3.79	2.60	2.20

Each of AB-V1, AB-V2, AB-V3, and AB-V4 is a variant of AB-DMOEA. "+", "−", and "≈" in the Wilcoxon test represent the performance of each variant is significantly better, worse, and similarly compared to AB-DMOEA, respectively.

Table 7
Computational complexity and the average running time of all algorithms in each benchmark suit.

Computational complexity	Algorithm	FDA [24]	dMOP [54]	UDF [55]	F [13]	DF [56]	Total
$\max\left(O\left(MN^2\right), O(N^2 \log N)\right)$	SGEA	1965.44	1109.06	2478.38	2023.11	3340.43	10916.42
$O\left(MN^2\right)$	PBDMO	39.16	17.96	59.07	61.01	113.89	291.09
$O\left(MN^2\right)$	MOEA/D-HMPS	90.93	47.03	117.76	101.56	434.14	791.42
$O\left(MN^2\right)$	MOEA/D-ARMS	94.56	47.28	107.72	94.01	370.41	713.99
$O\left(DN^2\right)$	HPPCM	43.05	26.99	59.78	55.66	203.10	388.58
$O(NlogN)$	PCPB	73.83	46.23	101.01	86.08	207.28	514.43
$O\left(MN^2\right)$	RVCP	69.85	46.53	115.94	99.30	195.69	539.03
$O\left(MN^2\right)$	AB-DMOEA	92.12	47.03	95.29	69.04	206.76	510.24

M is the number of objectives, D is the dimensions of decision variables, and N is the number of population.

In FDA1, AB-DMOEA achieved significant results in all stages due to its rapid adaptability and ability to process huge amounts of historical information. In the early term of UDF1, MOEA/D-HMPS, and MOEA/D-ARMS perform similarly to AB-DMOEA as their strategies are based on historical information. In the later stage, their performance is not as well as AB-DMOEA, indicating that the proposed memory boosting strategy is superior.

It can be seen that most of the algorithms have similar effects in the environment of FDA4, F6, and DF9. This is due to the higher level of environmental fluctuations or non-linear correlation of decision variables in these test instances, which makes conventional strategies ineffective. In general, the AB-DMOEA performs outstandingly in problems with mild or periodic environmental changes and rarely achieves bad results. The performance of AB-DMOEA under 100 environmental changes is superior to that of compared DMOEAs, demonstrating the efficacy of the proposed strategies.

5.5. AB-DMOEA parameter sensitivity analysis

There are three control parameters in the AB-DMOEA, the switch of dominated solution reinforcement strategy, and r_1 and r_2 values in the prediction boosting strategy. Separate experiments will be conducted

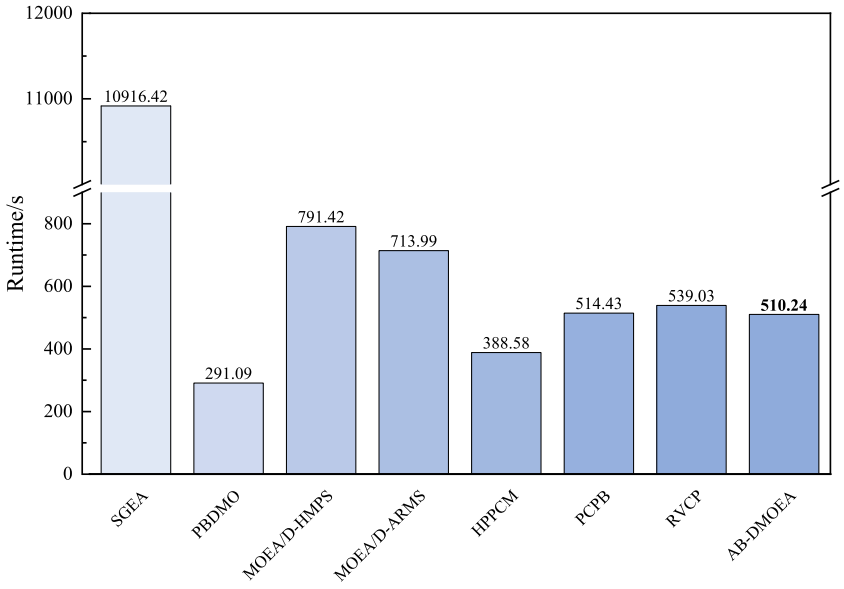


Fig. 9. The average runtime over all the benchmark suites of each algorithm.

Table 8
Mean and standard deviation values of the MIGD metric obtained by the AB-DMOEA with different thresholds.

Problem	0.5N	0.6N	0.7N	0.8N	0.9N
DF1	9.7288e−3 (3.00e−3)	9.0224e−3 (1.57e−3)	9.0014e−3 (1.33e−3)	9.5219e−3 (2.08e−3)	9.1143e−3 (2.32e−3)
DF2	4.9537e−2 (1.68e−2)	4.7810e−2 (1.79e−2)	4.8018e−2 (1.82e−2)	4.9017e−2 (1.87e−2)	6.0180e−2 (2.94e−2)
DF3	1.1765e−2 (2.28e−3)	1.2572e−2 (2.75e−3)	1.1973e−2 (2.32e−3)	2.4984e−2 (6.79e−3)	1.9337e−2 (8.12e−3)
DF4	8.3303e−2 (5.52e−3)	8.5079e−2 (6.54e−3)	8.3153e−2 (7.83e−3)	8.7039e−2 (6.58e−3)	8.7418e−2 (5.83e−3)
DF5	9.6280e−3 (2.09e−3)	1.0005e−2 (2.56e−3)	9.9287e−3 (1.21e−3)	9.8237e−3 (2.16e−3)	9.0202e−3 (9.29e−4)
DF6	6.5519e−1 (1.72e−1)	7.5222e−1 (2.48e−1)	7.1815e−1 (2.13e−1)	6.2329e−1 (1.99e−1)	7.2541e−1 (1.82e−1)
DF7	1.2324e−1 (3.76e−2)	1.1681e−1 (4.80e−2)	8.5748e−2 (4.25e−2)	1.1999e−1 (4.97e−2)	1.0709e−1 (4.67e−2)
DF8	1.6528e−1 (1.24e−2)	1.6933e−1 (1.13e−2)	1.4678e−1 (1.77e−2)	1.7162e−1 (7.49e−3)	1.7517e−1 (7.28e−3)
DF9	5.9048e−2 (1.99e−2)	5.6052e−2 (1.19e−2)	5.4067e−2 (1.17e−2)	5.6155e−2 (1.65e−2)	5.4758e−2 (1.00e−2)
DF10	1.9672e−1 (1.46e−2)	2.0214e−1 (1.24e−2)	1.8947e−1 (1.65e−2)	1.9733e−1 (1.21e−2)	2.0345e−1 (1.37e−2)
DF11	7.2619e−2 (1.18e−3)	7.3979e−2 (2.76e−3)	7.3147e−2 (5.93e−4)	7.4720e−2 (5.36e−3)	7.3359e−2 (1.50e−3)
DF12	2.7503e−1 (8.93e−3)	2.7913e−1 (2.01e−2)	2.6845e−1 (1.23e−2)	2.8714e−1 (2.71e−2)	3.0335e−1 (1.62e−2)
DF13	2.3374e−1 (6.88e−2)	2.5303e−1 (6.30e−2)	2.5195e−1 (6.23e−2)	2.2343e−1 (6.59e−2)	2.4883e−1 (6.27e−2)
DF14	9.7853e−2 (5.22e−2)	9.4509e−2 (5.44e−2)	1.0049e−1 (5.51e−2)	8.5831e−2 (4.34e−2)	9.6759e−2 (4.93e−2)

to investigate the impact of these parameters on the performance of AB-DMOEA.

5.5.1. The switch of dominated solution reinforcement strategy

As discussed in Section 3.2, the threshold controls whether the strategy is used. If it is high, the effectiveness of the adaptive strategy will decrease. On the contrary, it will cause unnecessary computational costs and may affect the initial good convergence. To study the sensitivity of the threshold, five reasonable values are considered: 0.5, 0.6, 0.7, 0.8, and 0.9. The downward rounding value of the threshold multiplied by the population size serves N as a switch for the strategy. For intuitive comparison, experiment in DF benchmark, MIGD as an evaluation indicator, $n_t = 10$, and $\tau_t = 10$.

The results are given in Table 8, and it is clear that there is no single threshold for all problems. While some problems (e.g. DF11) are not very sensitive to the threshold, other problems (e.g. DF5 and DF14) show a wide variation in performance, with the best performance obtained when the threshold is around $0.7N$. Thresholds that are too low or too high can affect the performance of AB-DMOEA. Furthermore, it can be seen that the AB-DMOEA with a threshold of $0.7N$, although not always the top performer, is close to the best results. Consequently, a threshold of $0.7N$ emerges as the preferred choice for the dominated solution reinforcement strategy in AB-DMOEA.

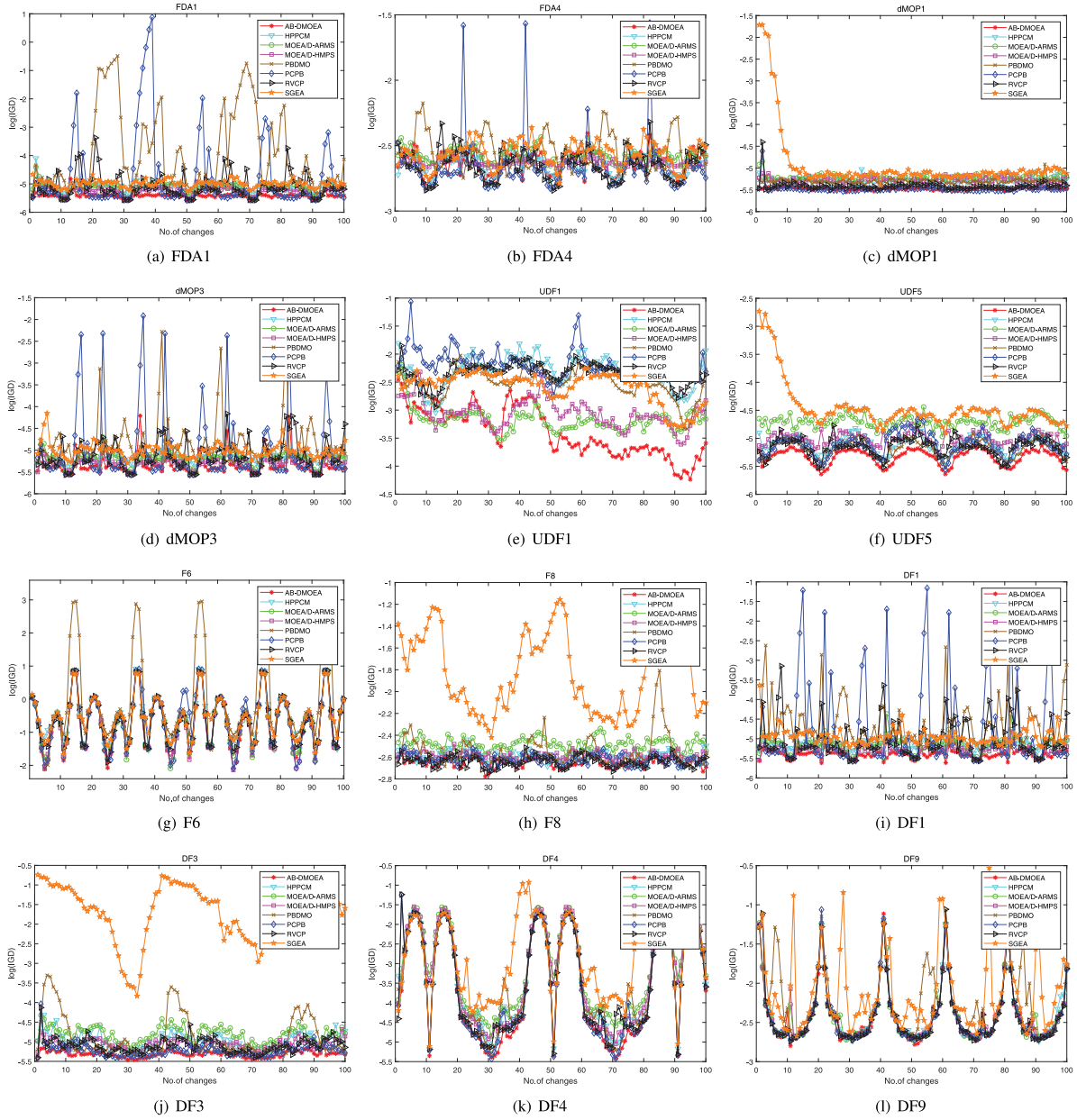
Table 9
AB-DMOEA with different r_1 and r_2 values.

S	r_1 & r_2
AB-S1	r_1=1, r_2=0
AB-S2	$r_1 \in \{0.5, 1, 1.5\}$, r_2=0
AB-S3	r_1=1, r_2=1
AB-S4	$r_1 \in \{0.5, 1, 1.5\}$, r_2=1
AB-DMOEA	$r_1 \in \{0.5, 1, 1.5\}$, $r_2 \in \{0, 1\}$

5.5.2. The r_1 and r_2 values

The moving step $r_1 \in \{0.5, 1, 1.5\}$ divides the population into three parts and three step-sizes have been proven to be effective [37], which are widely used in prediction strategies to enhance convergence [9,58]. $r_2 \in \{0, 1\}$ controls the Gaussian perturbation as one of the commonly effective ways to enhance diversity [35,45,63]. The r_1 and r_2 values are set in different combinations, as detailed in Table 9. To fully compare the effects of two parameters, experiment in DF benchmark, MIGD as an evaluation indicator, and other settings are consistent with Section 4.

The experimental results are shown in Table 10. From the comparison, it can be seen that conventional single-step size prediction (AB-S1) performs poorly compared to other combinations. The variant with three-step sizes (AB-S2) or Gaussian perturbations (AB-S3) has good results in DF6, DF11, and DF14. AB-DMOEA combines the



(a) FDA1 (b) FDA4 (c) dMOP1

(d) dMOP3 (e) UDF1 (f) UDF5

(g) F6 (h) F8 (i) DF1

(j) DF3 (k) DF4 (l) DF9

Fig. 10. The average IGD convergence curves of all algorithms on representative test instances under 100 environmental changes.

characteristics of both and has achieved better performance in terms of convergence and diversity. It is important to combine the two appropriately, otherwise utilizing their advantages can be difficult (AB-S4).

5.6. Real-world application of dynamic multi-objective optimization

The real-world application of the smart greenhouse is a typical dynamic multi-objective optimization problem (DMOP), which has always been a focus of research [5]. Therefore, AB-DMOEA will be applied to the real-world control optimization of smart greenhouses. The mathematical model for the smart greenhouse problem (SG) applied in this section is referenced from [5]. Table 11 displays six environmental variables, including their normalized upper and lower bounds in the experiment.

H_{soil} (%rh) and T_{soil} (°C) represent the humidity and temperature of the soil. H_{air} (%rh) T_{air} (°C) represent the humidity and temperature of the air. CO_2 (ppm) and I (lx) the concentration of CO2 and light intensity in the smart greenhouse. The following are the state equations for the three main environmental parameters of the model.

(1) $Humidity$

The humidity of the smart greenhouse is modeled by the following equations.

$$\dot{x}_{steam} = \frac{1}{3600 \cdot GH} \cdot (Trans + WaterInj - EnvExc - CondEvap) \quad (23)$$

where $Trans$ is the transpiration of the plants, $WaterInj$ is the water injection, $EnvExc$ is the exchange with environment through ventilation, $CondEvap$ is the condensation and evaporation on the greenhouse hull.

(2) $Temperature$

The smart greenhouse temperature is modeled as follows.

$$\dot{x}_{atemp} = \frac{1}{HCap} \cdot (u_{heat} + HSun - HExVent - HExGround - HExHull - HCondEvap - HHum) \quad (24)$$

Table 10
Mean and standard deviation values of the MIGD metric obtained by the AB-DMOEA and its variants.

Problem	AB-S1	AB-S2	AB-S3	AB-S4	AB-DMOEA
DF1	5.2263e−3 (6.21e−4) ≈	5.9881e−3 (3.57e−3) ≈	5.0441e−3 (1.49e−3) ≈	5.0217e−3 (1.11e−3) ≈	5.6372e−3 (1.91e−3)
DF2	1.2947e−2 (1.27e−2) −	1.0545e−2 (1.24e−2) ≈	1.8707e−2 (2.59e−2) ≈	8.2984e−3 (8.43e−3) ≈	6.5142e−3 (4.90e−4)
DF3	7.4734e−3 (1.41e−3) −	5.4160e−3 (3.15e−4) −	5.4270e−3 (3.23e−4) −	5.5172e−3 (3.49e−4) −	5.0240e−3 (4.69e−5)
DF4	7.5691e−2 (9.21e−3) −	7.5228e−2 (6.96e−3) −	7.3848e−2 (7.05e−3) −	7.4441e−2 (6.87e−3) −	6.7600e−2 (2.20e−4)
DF5	6.1078e−3 (6.36e−4) −	4.9195e−3 (1.44e−4) −	4.8802e−3 (1.83e−4) −	4.9426e−3 (1.20e−4) −	4.7283e−3 (1.11e−4)
DF6	2.8476e−1 (2.43e−1) +	2.2655e−1 (1.22e−1) +	1.2417e−1 (9.47e−2) +	1.7608e−1 (1.14e−1) +	6.1862e−1 (5.27e−1)
DF7	2.3431e−2 (2.29e−2) ≈	7.4797e−2 (4.35e−2) −	9.0480e−2 (3.68e−2) −	7.1302e−2 (3.51e−2) −	2.4692e−2 (2.15e−2)
DF8	1.3046e−1 (1.48e−2) ≈	1.2976e−1 (1.71e−2) ≈	1.2364e−1 (8.08e−3) ≈	1.2515e−1 (8.15e−3) ≈	1.2345e−1 (1.21e−3)
DF9	2.0821e+0 (5.20e−3) ≈	2.0797e+0 (5.15e−3) ≈	2.0796e+0 (3.02e−3) ≈	2.0790e+0 (6.31e−3) ≈	2.0843e+0 (2.54e−2)
DF10	1.7003e−1 (2.21e−2) −	1.3799e−1 (2.04e−2) −	1.4305e−1 (1.44e−2) −	1.3787e−1 (2.35e−2) −	1.0794e−1 (1.22e−2)
DF11	7.0025e−2 (2.99e−3) ≈	6.8309e−2 (1.76e−3) +	6.8028e−2 (1.40e−3) +	6.8609e−2 (2.40e−3) +	6.9849e−2 (4.85e−4)
DF12	2.6586e−1 (1.02e−2) −	2.6630e−1 (5.74e−3) −	2.6391e−1 (7.10e−3) −	2.6558e−1 (6.20e−3) −	2.5492e−1 (1.15e−3)
DF13	2.4360e−1 (7.74e−2) −	2.8054e−1 (7.60e−2) −	2.6460e−1 (8.26e−2) −	2.5489e−1 (8.57e−2) −	1.5522e−1 (1.43e−3)
DF14	1.2025e−1 (5.62e−2) ≈	1.0004e−1 (4.70e−2) +	7.1184e−2 (3.76e−2) +	9.5022e−2 (4.69e−2) +	1.4738e−1 (2.51e−2)
+/−/≈	1/7/6	3/7/4	3/7/4	3/7/4	

"+", "−", and "≈" indicate that the performance of AB-DMOEA is worse than, better than, and similar to that of the corresponding algorithm, respectively.

Table 11
The detailed upper and lower bounds of environmental parameters.

Environmental parameter	Upper and lower bound
H_{soil} (%rh)	[0, 1]
T_{soil} (°C)	[−1, 1]
H_{air} (%rh)	[−1, 1]
T_{air} (°C)	[−1, 1]
CO_2 (ppm)	[−1, 1]
I (lx)	[−1, 1]

where $HCap$ is the heat capacity of the air and the plants, u_{heat} is the heating through the heating system, $HSun$ is the heating from the sun, $HExVent$ is the heat exchange with the environment through ventilation, $HExGround$ heat exchange through the ground, $HExHull$ is the heat exchange through the greenhouse hull, $HCondEvap$ is the heat change due to condensation on the greenhouse hull, $HHum$ is the heat change due to change in indoor humidity.

(3) CO_2 *density*

The CO2 density is modeled by the following equations.

$$\dot{x}_{CO_2} = \frac{u_{CO_2} - CPhoto - CExVent}{3600 \cdot 10^{-6} \cdot DC \cdot GH} \tag{25}$$

where u_{CO_2} is the artificially injected CO2, $CPhoto$ is the CO2 consumption by the plants through photo-synthesis and transpiration, $CExVent$ is the exchange with the environment through ventilation.

Detailed parameter settings can be found in [5,70]. A fourth-order Runge–Kutta formula can approximate the solution to the equations. Control variables are sampled at each half-hour interval, which are subsequently converted into seconds within the equations. The conversion is essential because the differential equations require approximation through small time slices to ensure both stability and accuracy. Our work applies the mathematical model of the smart greenhouse mentioned above to simulation. For the comparison, five state-of-the-art DMOEAs were selected and the relevant settings are the same as in Section 4.

The experimental results of the smart greenhouse problem (SG) are shown in Table 12. The test results of all the algorithms were obtained by running them 30 times independently and the Wilcoxon rank-sum test [60] and Friedman test [61] are conducted on the results to provide statistically reasonable conclusions. "+", "−", and "≈" represent the compared algorithm is statistically significantly better, worse, and similar to the proposed AB-DMOEA. The results show that AB-DMOEA ranked first in SG, demonstrating its effectiveness in real-world applications such as smart greenhouses. The proposed mechanism is highly robust and can respond promptly to time-varying environments. In this table, SGEA, MOEA/D-HMPS, HPPCM, and RVCP demonstrate comparable performance, whereas MOEA/D-ARMS stands out as significantly different from the other algorithms.

6. Conclusion

In this paper, an adaptive boosting mechanism is proposed and applied to both dynamic and static environments to solve DMOPs. The AB-DMOEA consists of three components, dominated solution reinforcement strategy, adaptive boosting response mechanism, and static optimization boosting mechanism. These components work collaboratively to balance the exploitation and exploration during optimization. When the environment changes for the first time, the static optimization boosting mechanism selects the appropriate static multi-objective optimizer based on the non-dominated solutions and *MED* ratio. The adaptive boosting response mechanism includes three strategies based on prediction, memory, and diversity. These strategies adjust the proportion of the optimized population based on the history and current results of each strategy. To ensure the effectiveness of the adaptive boosting response mechanism, the dominated solution reinforcement strategy is used for maintenance. If there are fewer non-dominated solutions in the population, the strategy moves dominated solutions towards non-dominated solutions to enhance convergence. Afterward, the population undergoes the GA and DE operators, then makes a selection based on the *MED* to increase diversity.

To examine the performance of the adaptive boosting response mechanism, experiments are conducted on five benchmark suites with seven state-of-the-art DMOEAs. The results show that the proposed method outperforms others in terms of MIGD and MHV metrics. In addition, the proposed method has low computational costs and demonstrates strong robustness, indicating its potential for application in a wider range of fields. The application in the smart greenhouse problem demonstrates its practicability in real-world cases. However, there is still improvement room in the algorithm to optimize the DF and F test suites and some 3-dimensional test instances. Further improvement of the adaptive enhancement mechanism is needed for accuracy and robustness. In the future, we plan to apply the adaptive boosting mechanism to dynamic many-objective optimization and real-world applications. The Matlab source code of AB-DMOEA can be downloaded from Hu Peng's homepage: https://whuph.github.io/index.html.

Table 12
Mean and standard deviation MIGD were obtained by six algorithms on real-world problem SG (smart greenhouse problem).

Problem	SGEA	MOEA/D-HMPS	MOEA/D-ARMS	HPPCM	RVCP	AB-DMOEA
SG	1.1497e−1 (1.94e−2) −	1.2843e−1 (2.12e−2) −	5.4138e−1 (2.08e−2) −	1.0097e−1 (1.85e−2) −	1.0780e−1 (4.23e−2) −	**9.2185e−2 (2.19e−2)**
+/−/≈	0/1/0	0/1/0	0/1/0	0/1/0	0/1/0	–
Rank	4.00	5.00	6.00	2.00	3.00	**1.00**

CRediT authorship contribution statement

Hu Peng: Writing – original draft, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization. **Jianpeng Xiong:** Writing – original draft, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Chen Pi:** Software, Investigation. **Xinyu Zhou:** Supervision, Project administration, Funding acquisition. **Zhijian Wu:** Supervision, Project administration.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Hu Peng reports financial support was provided by National Natural Science Foundation of China. Xinyu Zhou reports financial support was provided by National Natural Science Foundation of China. Hu Peng reports was provided by Science and Technology Plan Projects of Jiangxi Provincial Education Department. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62266024, 61966019) and the Science and Technology Plan Projects of Jiangxi Provincial Education Department (GJJ2201906).

References

[1] K. Deb, U.B. Rao N, S. Karthik, Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2007, pp. 803–817.
[2] Y.-N. Guo, J. Cheng, S. Luo, D. Gong, Y. Xue, Robust dynamic multi-objective vehicle routing optimization method, IEEE/ACM Trans. Comput. Biol. Bioinform. 15 (6) (2018) 1891–1903.
[3] S. Nguyen, M. Zhang, M. Johnston, K.C. Tan, Automatic design of scheduling policies for dynamic multi-objective job shop scheduling via cooperative coevolution genetic programming, IEEE Trans. Evol. Comput. 18 (2) (2014) 193–208.
[4] D. Gong, B. Xu, Y. Zhang, Y. Guo, S. Yang, A similarity-based cooperative co-evolutionary algorithm for dynamic interval multiobjective optimization problems, IEEE Trans. Evol. Comput. 24 (1) (2020) 142–156.
[5] Z. Zhang, Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control, Appl. Soft Comput. 8 (2) (2008) 959–971.
[6] H. Peng, C. Wang, Y. Han, W. Xiao, X. Zhou, Z. Wu, Micro multi-strategy multi-objective artificial bee colony algorithm for microgrid energy optimization, Future Gener. Comput. Syst. 131 (2022) 59–74.
[7] H. Peng, Z. Xu, J. Qian, X. Dong, W. Li, Z. Wu, Evolutionary constrained optimization with hybrid constraint-handling technique, Expert Syst. Appl. 211 (2023) 118660.
[8] H. Peng, W. Xiao, Y. Han, A. Jiang, Z. Xu, M. Li, Z. Wu, Multi-strategy firefly algorithm with selective ensemble for complex engineering optimization problems, Appl. Soft Comput. 120 (2022) 108634.
[9] H. Peng, C. Mei, S. Zhang, Z. Luo, Q. Zhang, Z. Wu, Multi-strategy dynamic multi-objective evolutionary algorithm with hybrid environmental change responses, Swarm Evol. Comput. 82 (2023) 101356.
[10] R. Chen, K. Li, X. Yao, Dynamic multiobjectives optimization with a changing number of objectives, IEEE Trans. Evol. Comput. 22 (1) (2018) 157–171.
[11] W.T. Koo, C.K. Goh, K.C. Tan, A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment, Memet. Comput. 2 (2) (2010) 87–110.
[12] C.-K. Goh, K.C. Tan, A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization, IEEE Trans. Evol. Comput. 13 (1) (2009) 103–127.
[13] A. Zhou, Y. Jin, Q. Zhang, A population prediction strategy for evolutionary dynamic multiobjective optimization, IEEE Trans. Cybern. 44 (1) (2014) 40–53.
[14] A. Muruganantham, K.C. Tan, P. Vadakkepat, Evolutionary dynamic multiobjective optimization via Kalman filter prediction, IEEE Trans. Cybern. 46 (12) (2016) 2862–2873.
[15] M. Jiang, Z. Huang, L. Qiu, W. Huang, G.G. Yen, Transfer learning-based dynamic multiobjective optimization algorithms, IEEE Trans. Evol. Comput. 22 (4) (2018) 501–514.
[16] D. Chen, F. Zou, R. Lu, X. Wang, A hybrid fuzzy inference prediction strategy for dynamic multi-objective optimization, Swarm Evol. Comput. 43 (2018) 147–165.
[17] F. Zou, G.G. Yen, L. Tang, C. Wang, A reinforcement learning approach for dynamic multi-objective optimization, Inform. Sci. 546 (2021) 815–834.
[18] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[19] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[20] Q. Zhang, A. Zhou, Y. Jin, RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm, IEEE Trans. Evol. Comput. 12 (1) (2008) 41–63.
[21] M.-F. Leung, J. Wang, A collaborative neurodynamic approach to multiobjective optimization, IEEE Trans. Neural Netw. Learn. Syst. 29 (11) (2018) 5738–5748.
[22] S. Jiang, J. Zou, X. Yao, Evolutionary dynamic multi-Objective optimisation: a survey, ACM Comput. Surv. 55 (4) (2022) 1–47.
[23] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, E. Tsang, Prediction-Based Population Re-Initialization for Evolutionary Dynamic Multi-Objective Optimization, Springer Berlin Heidelberg, 2007, pp. 832–846.
[24] M. Farina, K. Deb, P. Amato, Dynamic multiobjective optimization problems: test cases, approximations, and applications, IEEE Trans. Evol. Comput. 8 (5) (2004) 425–442.
[25] S. Jiang, S. Yang, A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization, IEEE Trans. Evol. Comput. 21 (1) (2017) 65–82.
[26] X.-F. Liu, X.-X. Xu, Z.-H. Zhan, Y. Fang, J. Zhang, Interaction-based prediction for dynamic multiobjective optimization, IEEE Trans. Evol. Comput. 27 (6) (2023) 1881–1895.
[27] S. Sahmoud, H.R. Topcuoglu, Sensor-based change detection schemes for dynamic multi-objective optimization problems, in: IEEE Symposium Series on Computational Intelligence, SSCI, 2016, pp. 1–8.
[28] C. Wang, G.G. Yen, M. Jiang, A grey prediction-based evolutionary algorithm for dynamic multiobjective optimization, Swarm Evol. Comput. 56 (1) (2020) 100695.
[29] H. Richter, Detecting change in dynamic fitness landscapes, in: IEEE Congress on Evolutionary Computation, 2009, pp. 1613–1620.
[30] S. Sahmoud, H. Topcuoglu, Hybrid techniques for detecting changes in less detectable dynamic multiobjective optimization problems, in: Genetic & Evolutionary Computation Conference Companion, 2019, pp. 1449–1456.
[31] Z. Peng, J. Zheng, J. Zou, M. Liu, Novel prediction and memory strategies for dynamic multiobjective optimization, Soft Comput. 19 (9) (2015) 2633–2653.
[32] X. Zhang, G. Yu, Y. Jin, F. Qian, Elitism-based transfer learning and diversity maintenance for dynamic multi-objective optimization, Inform. Sci. 636 (2023) 118927.
[33] K. Yu, D. Zhang, J. Liang, K. Chen, C. Yue, K. Qiao, L. Wang, A correlation-guided layered prediction approach for evolutionary dynamic multiobjective optimization, IEEE Trans. Evol. Comput. 27 (5) (2023) 1398–1412.
[34] L. Yan, W. Qi, A. Qin, S. Yang, D. Gong, B. Qu, J. Liang, Manifold clustering-based prediction for dynamic multiobjective optimization, Swarm Evol. Comput. 77 (2023) 101254.
[35] Z. Liang, S. Zheng, Z. Zhu, S. Yang, Hybrid of memory and prediction strategies for dynamic multiobjective optimization, Inf. Sci. 485 (2019) 200–218.

[36] R. Azzouz, S. Bechikh, L.B. Said, A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy, Soft Comput. 21 (4) (2017) 885–906.
[37] Q. Zhang, S. Yang, S. Jiang, R. Wang, X. Li, Novel prediction strategies for dynamic multiobjective optimization, IEEE Trans. Evol. Comput. 24 (2) (2020) 260–274.
[38] X. Zhang, Y. Tian, Y. Jin, A knee point-driven evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 19 (6) (2015) 761–776.
[39] J. Zou, Q. Li, S. Yang, H. Bai, J. Zheng, A prediction strategy based on center points and knee points for evolutionary dynamic multi-objective optimization, Appl. Soft Comput. 61 (2017) 806–818.
[40] J. Zheng, B. Zhang, J. Zou, S. Yang, Y. Hu, A dynamic multi-objective evolutionary algorithm based on Niche prediction strategy, Appl. Soft Comput. 142 (2023) 110359.
[41] J. Zheng, F. Zhou, J. Zou, S. Yang, Y. Hu, A dynamic multi-objective optimization based on a hybrid of pivot points prediction and diversity strategies, Swarm Evol. Comput. 78 (2023) 101284.
[42] Y. Guo, H. Yang, M. Chen, J. Cheng, D. Gong, Ensemble prediction-based dynamic robust multi-objective optimization methods, Swarm Evol. Comput. 48 (2019) 156–171.
[43] Y. Guo, H. Yang, M. Chen, D. Gong, S. Cheng, Grid-based dynamic robust multi-objective brain storm optimization algorithm, Soft Comput. 24 (10) (2020) 7395–7415.
[44] L. Yan, W. Qi, J. Liang, B. Qu, K. Yu, C. Yue, X. Chai, Interindividual correlation and dimension-based dual learning for dynamic multiobjective optimization, IEEE Trans. Evol. Comput. 27 (6) (2023) 1780–1793.
[45] H. Sun, C. Wang, X. Li, Z. Hu, A decision variable classification strategy based on the degree of environmental change for dynamic multiobjective optimization, European J. Oper. Res. 313 (1) (2024) 296–311.
[46] H. Li, Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comput. 13 (2) (2009) 284–302.
[47] M. Jiang, Z. Wang, S. Guo, X. Gao, K.C. Tan, Individual-based transfer learning for dynamic multiobjective optimization, IEEE Trans. Cybern. 51 (10) (2021) 4968–4981.
[48] M. Jiang, Z. Wang, H. Hong, G.G. Yen, Knee point-based imbalanced transfer learning for dynamic multiobjective optimization, IEEE Trans. Evol. Comput. 25 (1) (2021) 117–129.
[49] Y. Freund, R.E. Schapire, A desicion-theoretic generalization of on-line learning and an application to boosting, in: European Conference on Computational Learning Theory, Springer, 1995, pp. 23–37.
[50] K. Zhang, C. Shen, X. Liu, G.G. Yen, Multiobjective evolution strategy for dynamic multiobjective optimization, IEEE Trans. Evol. Comput. 24 (5) (2020) 974–988.
[51] L. Chen, H. Wang, D. Pan, H. Wang, W. Gan, D. Wang, T. Zhu, Dynamic multiobjective evolutionary algorithm with adaptive response mechanism selection strategy, Knowl.-Based Syst. 246 (2022) 108691.
[52] L. Cao, L. Xu, E.D. Goodman, H. Li, Decomposition-based evolutionary dynamic multiobjective optimization using a difference model, Appl. Soft Comput. 76 (2019) 473–490.
[53] K. Deb, M. Goyal, A combined genetic adaptive search (geneas) for engineering design, in: Computer Science and Informatics, Vol. 26, 1996, pp. 30–45.
[54] C.-K. Goh, K.C. Tan, A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization, IEEE Trans. Evol. Comput. 13 (1) (2009) 103–127.
[55] S. Biswas, S. Das, P.N. Suganthan, C.A.C. Coello, Evolutionary multiobjective optimization in dynamic environments: A set of novel benchmark functions, in: 2014 IEEE Congress on Evolutionary Computation, CEC, 2014, pp. 3192–3199.
[56] S. Jiang, S. Yang, X. Yao, K.C. Tan, M. Kaiser, N. Krasnogor, Benchmark problems for CEC2018 competition on dynamic multiobjective optimisation, in: IEEE Congress on Evolutionary Computation, CEC, 2018, pp. 1–18.
[57] P. Bosman, D. Thierens, The balance between proximity and diversity in multiobjective evolutionary algorithms, IEEE Trans. Evol. Comput. 7 (2) (2003) 174–188.
[58] Y. Chen, J. Zou, Y. Liu, S. Yang, J. Zheng, W. Huang, Combining a hybrid prediction strategy and a mutation strategy for dynamic multiobjective optimization, Swarm Evol. Comput. 70 (2022) 101041.
[59] J. Zheng, Q. Wu, J. Zou, S. Yang, Y. Hu, A dynamic multi-objective evolutionary algorithm using adaptive reference vector and linear prediction, Swarm Evol. Comput. 78 (2023) 101281.
[60] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.
[61] F. Wilcoxon, Individual comparisons by ranking methods. 1 (1945) 196–202.
[62] F. Zou, D. Chen, Q. Xu, R. Lu, A new prediction strategy combining T-S fuzzy nonlinear regression prediction and multi-step prediction for dynamic multi-objective optimization, Swarm Evol. Comput. 59 (2020) 100749.
[63] Y. Wu, Y. Jin, X. Liu, A directed search strategy for evolutionary dynamic multiobjective optimization, Soft Comput. 19 (11) (2015) 3221–3235.
[64] Q. Zhao, B. Yan, Y. Shi, M. Middendorf, Evolutionary dynamic multiobjective optimization via learning from historical search process, IEEE Trans. Cybern. 52 (7) (2022) 6119–6130.
[65] G. Chen, Y. Guo, M. Huang, D. Gong, Z. Yu, A domain adaptation learning strategy for dynamic multiobjective optimization, Inform. Sci. 606 (2022) 328–349.
[66] P. Wang, Y. Ma, M. Wang, A dynamic multi-objective optimization evolutionary algorithm based on particle swarm prediction strategy and prediction adjustment strategy, Swarm Evol. Comput. 75 (2022) 101164.
[67] Y. Yang, Y. Ma, M. Wang, P. Wang, A dynamic multi-objective evolutionary algorithm based on gene sequencing and gene editing, Inform. Sci. 644 (2023) 119256.
[68] Y. Yang, Y. Ma, P. Wang, Y. Xu, M. Wang, A dynamic multi-objective evolutionary algorithm based on two-stage dimensionality reduction and a region Gauss adaptation prediction strategy, Appl. Soft Comput. 142 (2023) 110333.
[69] J. Zheng, Y. Zhou, J. Zou, S. Yang, J. Ou, Y. Hu, A prediction strategy based on decision variable analysis for dynamic Multi-objective Optimization, Swarm Evol. Comput. 60 (2021) 100786.
[70] R. Ursem, T. Krink, B. Filipič, A Numerical Simulator for a Crop-Producing Greenhouse, EVALife Technical Report, 2002, p. 11.