



Solving dynamic multi-objective problems using polynomial fitting-based prediction algorithm

Qingyang Zhang^{a,*}, Xiangyu He^a, Shengxiang Yang^b, Yongquan Dong^a, Hui Song^a, Shouyong Jiang^{c,*}

^a School of Computer Science and Technology, Jiangsu Normal University, Xuzhou, CO 221116, China

^b School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, United Kingdom

^c Department of Computer Science, University of Aberdeen, Aberdeen AB24 3UE, United Kingdom



ARTICLE INFO

Article history:

Received 18 February 2022

Received in revised form 5 August 2022

Accepted 6 August 2022

Available online 11 August 2022

Keywords:

Dynamic multi-objective optimization

Polynomial fitting

Prediction mechanism

Dynamic engineering design

ABSTRACT

Recently, dynamic multi-objective optimization has received growing attention due to its popularity in real-world applications. Inspired by polynomial fitting, this paper proposes a polynomial fitting-based prediction algorithm (PFPA) and incorporates it into the model-based multi-objective estimation of distribution algorithm (RM-MEDA) for solving dynamic multi-objective optimization problems. When an environment change is detected, the main mission of PFPA is to predict high-quality search populations for tracking the moving Pareto-optimal set effectively. Firstly, the non-dominated solutions obtained in past environments are utilized to predict high-quality solutions based on a multi-step movement strategy. Secondly, a polynomial fitting-based strategy is designed to fit the distribution of variables according to the obtained search populations, and capture the relationship between variables in the new search environment. Thirdly, some effective search agents are generated for improving population convergence and diversity based on characteristics of variables. To evaluate the performance of the proposed algorithm, experimental results on a set of benchmark functions, with a variety of different dynamic characteristics and difficulties, and two classical dynamic engineering design problems show that PFPA is competitive with some state-of-the-art algorithms.

© 2022 Elsevier Inc. All rights reserved.

1. Introduction

Dynamic multi-objective optimization problems(DMOPs) refers to solving multi-objective optimization problems (MOPs) in dynamic or uncertain conditions, and it has a growing number of applications [1]. Without loss of generality, considering the following minimization model of DMOPs.

$$\begin{aligned} \min_{x \in \Omega} F(x, t) &= (f_1(x, t), f_2(x, t), \dots, f_m(x, t))^T \\ s.t. \quad h_i(x, t) &= 0, \quad i = 1, 2, \dots, n_h \\ g_j(x, t) &\leq 0, \quad j = 1, 2, \dots, n_g \end{aligned} \tag{1}$$

* Corresponding authors.

E-mail addresses: sweqyian@126.com (Q. Zhang), shouyong.jiang@abdn.ac.uk (S. Jiang).

where $\Omega = \prod_{i=1}^D [L_i, U_i] \subset R^D$ is the feasible area of the decision space, and F consists of m time-varying objective functions. $x = (x_1, x_2, \dots, x_D)$ defines the decision vector involving D variables, L_i and U_i represent the lower and upper bounds of the i th variable x_i , respectively, t is the time instant of the problem. n_h and n_g denote the number of equality and inequality constraints at time t , respectively.

Different from static MOPs, DMOPs require to consider two important problems, i.e., multiobjectivity and dynamism. the former involves conflicting objectives, which is a common characteristic of MOPs. The latter involves various changes that can occur in control parameters, Pareto optimal set (POS) or Pareto-optimal front (POF). They bring great challenges to the tracking ability of optimization algorithms during the course of search. It is difficult to obtain optimal tradeoff solutions satisfying the requirements in dynamic environments. One way to address DMOPs is that DMOPs can be regarded as a sequence of static MOPs along the time horizon. This means that solving dynamic multi-objective problems is a process of obtaining a sequence of approximate Pareto-optimal solutions.

Recently, a growing number of effective dynamic multi-objective evolutionary algorithms (DMOEAs) have been proposed based on various learning mechanisms and models. As shown in Algorithm 1 for a basic framework of DMOEAs, it is obvious that the overall course of solving DMOPs consists of two main components: change handling after detecting a change and multi-objective optimization algorithms.

Algorithm 1: The basic framework of DMOEAs

- 1: Initialize time instance $t = 1$;
 - 2: Generate an initial population Pop_t ;
 - 3: While the stopping criterion is not met
 - 4: Change Detection
 - 5: If change is not detected, optimize population using MOEAs;
 - 6: Otherwise, evolve population using DMOEAs;
 - 7: Return 3
-

1.1. Change detection

Change detection is a core part of the framework, since it determines the timing of environmental change and whether there is a need to adopt environmental response mechanisms. Dynamic detection techniques include re-evaluating solutions [2] and checking population statistical information [3] through sensors. The former is more popular, since it is simple and easy to implement, but it also suffers from noise sensitivity. On the contrary, the latter is not sensitive to noise, but it needs some control parameters. Besides, sensors are placed appropriately for change detection [4] and need to be tested on various problems or applications. It is difficult to select an appropriate detection technique, since each method has its advantages and limitations for different DMOPs.

1.2. Multi-objective optimization algorithms

Apart from dynamic detection, multi-objective algorithm is also one of the key components for solving DMOPs. The existing research on multi-objective algorithms mainly focuses on three types described as follows. The first category is Pareto dominated-based algorithms, which utilize the dominance concept to determine whether the obtained individuals can be retained for next iteration. Various classical and representative algorithms have been proposed recently years, such as non-dominated sorting genetic algorithm II (NSGA-II) [5], NSGA-III [6] and strength Pareto evolutionary algorithm (SPEA2)[7]. In addition, some swarm intelligence algorithms are also adopted for multi-objective optimization, such as multi-objective grasshopper optimization algorithm (MOGOA) [8], and multi-objective multi-verse optimizer (MOMVO) [9], and so on. Pareto-dominance strategies perform well in different applications, but suffer from the risk of generating excessive boundary individuals in some cases.

The second category is indicator-based algorithms, which are designed according to performance metrics. The hypervolume [10], the epsilon indicator and the R2 have been widely employed to design optimization algorithms, such as indicator-based EA (IBEA) [11], approximation-guided EMO (AGE) [12]. The last category is decomposition-based algorithms, which are designed by decomposing an MOP into several sub-problems and solving them simultaneously. Typically, MOEA based on decomposition (MOEA/D) [13] and multi-objective particle swarm optimization algorithm based on decomposition (MPSOD) [14]. This kind of algorithm is effective, but depends heavily on weight vectors.

In addition, the research on complicated real-world problems is also important for evaluating the algorithmic effectiveness and guiding the algorithmic design. Some practical applications in various areas are described briefly as follows. Mirjalili et al. [15] assessed the performance of algorithms by solving several popular engineering design problems including four-bar truss design, speed reducer design, disk brake design, welded beam design, cantilever beam design and brushless DC wheel motor. Wang et al. [16] investigated urban bus scheduling problem considering route, number of vehicles and drivers. Besides, there exist many different optimization models, such as change detection in SAR images [17], elevator group control[18] and so on.

1.3. Dynamic multi-objective optimization algorithms

Influenced by the frequency or severity of change, diverse difficulties may appear in different forms, such as intensive computation, changing feasible region, irregular Pareto fronts and constraints. This requires that the optimization algorithm keep a good balance between diversity and convergence during the search of solutions. Research on dynamic multi-objective optimization algorithms can be classified into four categories: diversity-based algorithms, memory based algorithms, multi-population based algorithms, and prediction based algorithms. Firstly, diversity-based algorithms aim to employ different effective strategy for keeping the diversity of population after change detection. Respectively, Li [19] proposed a basic framework for generating subpopulations with good diversity using hierarchical linkage clustering. Chang [20] proposed a query-based mechanism for producing some individuals for diversity. Ye [21] employs multiple source transfer learning for DMOPs. In addition, immigration-based techniques are also introduced for diversity, such as memory-based immigration [22], elitism-based immigration [23]. Deb [24] proposed two different versions of DMOEAs by combining NSGAII with diversity-increase mechanisms.

In memory-based algorithms, some promising historical individuals are recorded and reused for guiding search when a change is detected. In [25], the best agents of previous obtained population are saved in an archive and used to replace some members of the search population in the current environment. Xie [26] integrates decision variable classification-based cooperative co-evolutionary mechanisms for dynamic multiobjective problems. Azzouz [27] proposed an adaptive hybrid population management strategy by considering memory, local technique and random strategy. In [2], a steady-state mechanism is introduced for generating individuals and responding to dynamic change effectively. The kind of these algorithms performs well in periodically changing problems.

In multi-population based algorithms, the diversity of search population is maintained through multiple subpopulations. Xu et al. [28] proposed two multi-population algorithms based on PSO and NSGAII, each sub-population was endowed with different levels of sensitive variables and responses to dynamic changes using different models. Liu et al. [29] utilized an external archive to store the obtained promising individuals and each sub-population evolved one objective independently. In [30], multiple sub-populations are employed for dealing with different objectives, and an external archive was used to share information between sub-populations and enhance diversity preservation.

The main idea of prediction-based algorithms is to estimate the possible POS/POF position in new dynamic circumstances according to previously obtained individuals. Zhou [31] utilizes the univariate auto-regression (AR) model for predicting the manifold and search population after changes are detected. Muruganantham [32] designed a DMOEA based on Kalman filter for solving DMOPs. Hu [33] proposed multi-directional prediction mechanism for generating high-quality search agents during the process of evolution. Besides, various prediction approaches and models have been designed, such as knee points [34], and center points [35]. This kind of algorithm are much popular in DMOEAs, since they are easy to implement and understand.

In addition to those discussed above, real-world applications have also attracted growing attention in recent years, and the following discusses some practical examples involved in different areas. Deb et al. [24] explored power scheduling in hydro-thermal generation systems, which involves two optimization objectives and three constraints about dynamic power demand, hydraulic,fuel cost and power system networks. Zhang, et al. [36] studied the dynamic welded beam design problem and dynamic speed reducer for evaluating the performance of dynamic artificial immune systems. In [37], a dynamic railway junction rescheduling problem with two objectives and dynamic constraints was investigated in railway networks. Kong et al. [38] investigated the dynamic power supply problem with dynamic changes of objectives and constraints. Gong et al. [39] solved dynamic multi-period portfolio selection problem using a similarity-based cooperative co-evolutionary algorithm. In addition, there are still various application problems such as dynamic scheduling [40], path planning [41] and so on.

It can be found from the above analysis that most of the existing dynamic multi-objective optimization algorithms are designed based on various knowledge or machine learning mechanisms and have been applied in various applications. However, a significant property, i.e., the relationship between variables, has not been well explored for generating promising individuals. Meanwhile, the polynomial fitting strategy can not only reflect the relationship between variables, but also be used to predict new variable values [42]. Motivated by this, this paper attempts to design an effective prediction-based algorithm for solving dynamic multiobjective optimization problems.

This paper can be summarized as follows. A new polynomial fitting-based prediction algorithm is proposed for generating high-quality search population when changes are detected. The designed algorithm consists of three different parts, a multi-step movement strategy, a polynomial fitting-based strategy and a sampling strategy. Firstly, the multi-step movement strategy mainly utilizes two different step sizes to predict new individuals based on the obtained historical population. Secondly, the polynomial fitting-based strategy aims to predict the potential position of new individuals. Thirdly, the sampling strategy tends to sample some well-distributed individuals based on relationship properties of variables for guiding search and improving the convergence of population during the course of optimization. Experimental results carried on a set of recently proposed functions and two dynamic engineering design problems show that the proposed algorithm is very competitive with other compared algorithms.

The following provides the organization of this paper. Section 2 presents the framework of the proposed algorithm in detail. In Section 3, the effectiveness of the proposed algorithm is evaluated on a suite of bedsxnchmarks and compared with several existing optimization approaches. Section 4 discusses the influence of each component and key parameters in the proposed algorithm. The proposed algorithm is utilized for solving two dynamic engineering design problems in Section 5. The conclusion of this study can be found in Section 6.

2. Proposed PFPA

The details of the proposed algorithm PFPA are presented in this section. Algorithm 2 provides the basic implementation steps of PFPA. When environment changes occur, PFPA introduces a prediction mechanism for generating a high-quality search population composed of three subpopulation resulting from different strategies. RMMEDA is employed for improving the quality of population over the course of optimization.

Algorithm 2: The overall framework of the proposed algorithm

- 1: Initialize relevant parameters settings.
 - 2: Initialize and evaluate population (Pop_{iter}) and set $iter = 1$.
 - 3: If the stop condition is not satisfied.
 - 4: If change detected, go to step 5; otherwise, go to step 10.
 - 5: Generate the first subpopulation ($SubPop_1$) using an multi-step movement strategy.
 - 6: Generate the second subpopulation ($SubPop_2$) based on a polynomial fitting based strategy.
 - 7: Generate the third subpopulation ($SubPop_3$) by a sampling strategy [43].
 - 8: Merge these subpopulations $MixPop = SubPop_1 \cup SubPop_2 \cup SubPop_3$.
 - 9: Obtain a population of $Popsize$ by non-dominated sorting of the merged population.
 - 10: Optimize population using RM-MEDA.
 - 11: $iter = iter + 1$, return to step 3.
-

2.1. Multi-step movement strategy

The multi-step movement strategy is proposed to predict the position of non-dominated solutions after the change is detected. Non-dominated solutions have good information of the problem under consideration, so it is important to predict their new positions based on historical information. The proposed strategy involves two important parameters, stepsize and direction of movement, described as follows.

Inspired by the fact that the geometric center of the POF/POS is a significant feature and can be utilized to represent the changing trend of data to some extent from the statistical point of view. Therefore, the movement direction is computed based on the center points of non-dominated solutions obtained in the last two consecutive generations. Here, suppose that Pop_t^{center} is the centroid of population (Pop_t) and Pos_t is the non-dominated sets of Pop_t at the time t . Then, the Pop_t^{center} can be calculated as follows.

$$Pop_t^{center} = \frac{\sum_{x_t \in Pop_t} x_t}{|Pop_t|} \quad (2)$$

where $|Pop_t|$ refers to the population cardinality, $x_t = (x_t^1, x_t^2, \dots, x_t^D)$ is a solution at time t , and the moving direction (Dir_t) at time t can be obtained by

$$Dir_t = Pop_t^{center} - Pop_{t-1}^{center}. \quad (3)$$

Then, the possible position of Pos_{t+1} can be predicted using the formula below:

$$Pos_{t+1} = Pos_t + Dir_t \times step \quad (4)$$

where $step$ defines as the moving stepsize along the moving direction of Dir_t . Here, two different values of $step$ (i.e., 0.3 and 1.0) are utilized for representing two different moving levels of Pos_t .

[Fig. 1](#) gives the description of this movement strategy. We use multi-step strategy rather one-step strategy because different levels of dynamic change may occur to different population individuals. Therefore, generating two positions by our strategy for each individual has a higher chance to cover the true POS than generating only one by one-step strategy.

Algorithm 2: Multi-step movement strategy

- 1: Retrieve the populations Pop_t and Pop_{t-1} at time t and $t - 1$, respectively;
 - 2: Calculate the population centers according to Eq. (2);
 - 3: Predict the moving direction according to Eq. (3);
 - 4: Generate two subpopulations Pos_{t+1}^{pr1} and Pos_{t+1}^{pr2} using Eq. (4) with different $step$ values;
 - 5: Save the subpopulations to $SubPop_1$.
-

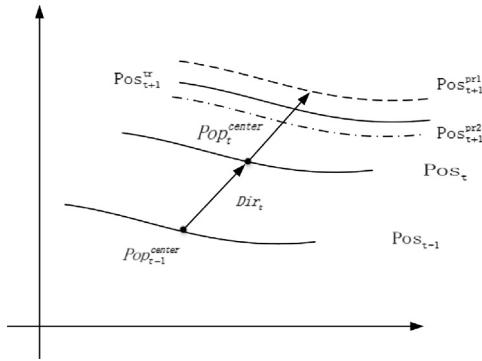


Fig. 1. Description of Multi-step movement strategy.

2.2. Polynomial fitting-based strategy

This subsection introduces the polynomial fitting-based strategy in detail, which aims at capturing complex relationship between variables according to the distribution of the population individuals, and using it to predict population positions in the new environment. This approach involves two important parts, i.e., curve prediction and generation of new individuals, which are described as follows.

Motivated by its success in machine learning applications, polynomial fitting as a simple linear model can address non-linearity of data, in addition to its computationally efficiency. This inspires us to use it to explore the relationship between variables and then to generate non-dominated solutions in order to keep track of changes. Here, as shown in Fig. 2, Fc_{t-1} and Fc_t are obtained by polynomial fitting according to the last two consecutive population data at time $t - 1$ and t , respectively. The moving direction of the curve formed by the population can be calculated using the following formulation.

$$MV_t = Fc_t - Fc_{t-1} \quad (5)$$

Then, the curve at time $t + 1$ can be predicted using the following equation.

$$Fc_{t+1} = Fc_t + MV_t \quad (6)$$

The solutions of another subpopulation can be generated by

$$Ind_{t+1} = Fc_{t+1} + cr \times ND_p \quad (7)$$

where cr is a scaling parameter, which ensures that the newly generated individuals are not far from the fitted curve and ND_p is generated by normal distribution to allow variations for the generated individuals.

Algorithm 4 presents the implementation steps of this strategy. In addition, as suggested in [43], the correlation matrix of variables can be utilized to distinguish these variables. Also, the study [44] suggests that the highest order of the fitting is not more than six. Therefore, the strategy will adaptively select the polynomial orders during the iterative process.

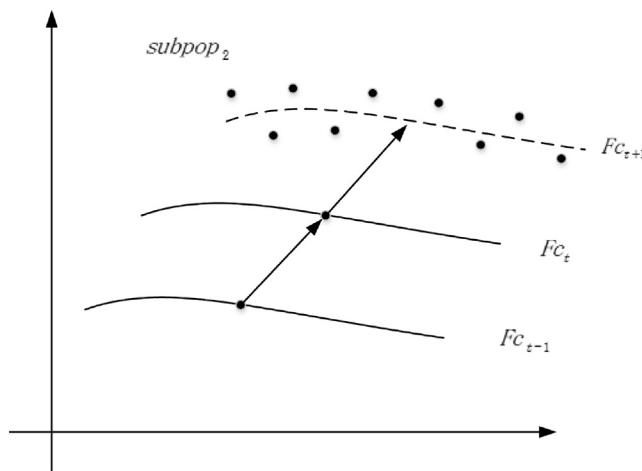


Fig. 2. Description of polynomial fitting-based strategy.

Algorithm4: Implementation details of polynomial fitting based strategy

- 1: Suppose the populations (Pop_t and Pop_{t-1}) at time t and $t - 1$, respectively.
- 2: Define the moving direction of characteristic curve based on Eq. (5).
- 3: Predict the possible polynomial curve according to Eq. (6).
- 4: Create the subpopulation $SubPop_2$ sampled from the decision space using Eq. (7).

2.3. Sampling strategy

The recent study [43] proposed a sampling strategy which is effective for handling dynamic changes. We intend to use this strategy to obtain some well-distributed individuals as the third subpopulation for improving the quality of the search population in the new environment. Fig. 3 illustrates the employed sampling strategy in a 2D scenario.

Specifically, suppose that x_d is the identified most principal variable in the POS [43], N_1 is the number of sampled points for x_d . For any non-principal variable x_k ($k \in \{1, \dots, D\} \setminus \{d\}$), N_2 denotes the number of sampling points computed by

$$hx_k^i = L_k + i \times \frac{U_k - L_k}{N_2}, \quad (i = 1, \dots, N_2) \quad (8)$$

where L_k and U_k denote the lower and upper bounds of x_k , respectively. Then, this strategy will generate a total of $N_1 N_2$ sampling points (marked by black in Fig. 3) $\bar{x} = (x_d^i, hx_k^j)$, $i = 1, \dots, N_1$ and $j = 1, \dots, N_2$, and these sampled points will be screened using the non-dominated sorting to obtain several promising agents (marked in red in Fig. 3) as the third subpopulation.

It is crucial to determine the values of N_1 and N_2 appropriately. Actually, the setting of N_1 and N_2 is problem-specific; large values may cause high computational complexity whereas small values may result in inadequate sampling. Therefore, for simplicity, N_2 is set to the number of variables ($N_2 = D$), and N_1 is set to a value such that $N_1 N_2 = N_1 D \leq N/2$.

3. Experiments

This section is to evaluate the effectiveness of the proposed algorithm through experimental studies. The experimental settings include benchmark problems, performance evaluation indicators, comparison algorithms and related parameter settings, followed by experimental results and analysis.

3.1. Test instances

This paper employs DF test problems [45], which consist of various difficulties and characteristics including irregular POF shapes, time-linkage, variable separability for comprehensive assessing the performance of algorithms.

3.2. Performance indicators

This paper utilizes different performance metrics described below for measuring the effectiveness of algorithms. Firstly, the widely used mean inverted generational distance (MIGD) is adopted to judge the convergence and diversity of the best

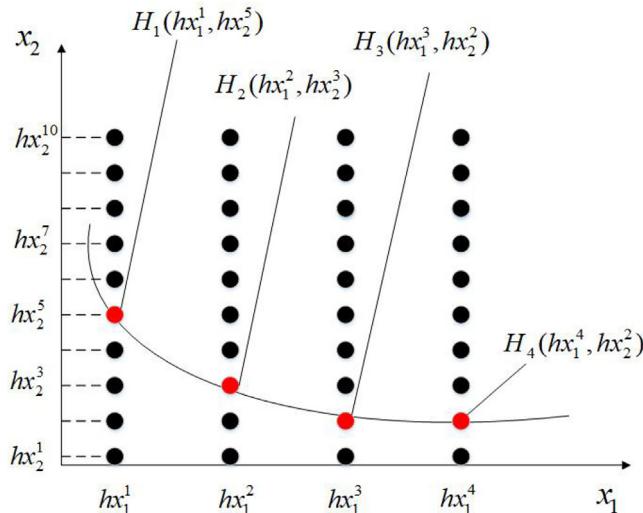


Fig. 3. An description of the sampling strategy in 2D case.

population obtained by an algorithm [32]. Secondly, the mean Schott's spacing metric (MSP) is considered to measure the distribution of the computed solutions [39]. Thirdly, the mean hypervolume (MHV) metric is used [38]. Finally, t-test is adopted to check whether the obtained results are statistically different among all the optimization algorithms at 0.05 significant level [46]. For each table of result, \ddagger , \dagger or \wr indicate that the performance of PFPA is better than, worse than or similar to that of the corresponding algorithm, respectively.

3.3. Compared algorithms

This paper employs four different dynamic multi-objective optimization algorithms for comparison with PFPA, which are described below in detail. Firstly, a first-order difference model-based MOEA/D algorithm (MOEA/D-FD) [47] employs the historically recorded search population for determining movement directions and predicting potential locations of the new POF. Meanwhile, it has a decomposition-based framework that can ensure the distribution of individuals. Secondly, TrDMOEA aims to combine transfer learning mechanism with evolutionary algorithms to handle dynamic environments [48]. Thirdly, as a dynamic version of NSGA-II, DNGAA aims to replace some existing population individuals using random solutions in the event of dynamic changes [24]. Finally, population prediction strategy (PPS) predicts a population after a change using population centers and manifolds through an autoregression (AR) model [31].

Table 1
Mean and standard deviation values of MIGD obtained by five algorithm for $(n_t, \tau_t) = (5, 20)$.

Fun.	(n_t, τ_t)	MOEA/D-FD	TrDMOEA	DNSGAA	PPS	PFPA
DF1	(5,20)	1.179e-2(1.764e-4)	1.777e-2(2.139e-3)	6.227e-2(4.221e-3)	3.668e-1(7.186e-2)	1.038e-2(4.838e-4)
	p	1.761e-1	3.644e-2	3.917e-2	2.154e-10	–
	h	0	1	1	1	–
DF2	(5,20)	1.073e-2(3.404e-4)	6.565e-3(6.454e-4)	4.261e-2(4.281e-3)	2.440e-1(5.131e-2)	5.434e-2(6.335e-3)
	p	3.020e-11	1.087e-1	6.696e-11	2.813e-2	–
	h	1	0	1	1	–
DF3	(5,20)	4.606e-2(4.499e-3)	5.734e-2(1.981e-2)	2.581e-1(3.135e-2)	1.797e-1(1.494e-1)	4.581e-2(8.669e-3)
	p	3.020e-11	2.370e-10	3.012e-11	2.922e-9	–
	h	1	1	1	1	–
DF4	(5,20)	1.186e-1(2.085e-3)	5.872e-1(1.742e-3)	8.384e-2(2.346e-3)	1.370e-1(1.003e-2)	9.570e-2(1.939e-3)
	p	7.259e-4	1.168e-2	2.052e-3	5.570e-3	–
	h	1	1	1	1	–
DF5	(5,20)	2.027e-2(2.061e-4)	2.808e-2(3.792e-4)	7.844e-2(5.077e-3)	3.723e-1(1.041e-1)	1.520e-2(7.420e-4)
	p	1.154e-1	2.753e-1	8.418e-1	2.034e-9	–
	h	0	0	0	1	–
DF6	(5,20)	4.514e+0(4.384e-1)	9.798e-1(2.154e-1)	2.267e+0(2.674e-1)	6.897e+0(8.883e-1)	4.654e-1(1.293e-2)
	p	1.734e-9	6.889e-4	1.254e-7	1.492e-6	–
	h	1	1	1	1	–
DF7	(5,20)	8.858e-2(1.863e-2)	3.829e-2(1.287e-3)	3.193e-1(5.110e-2)	6.720e-2(1.938e-2)	1.051e-2(3.048e-4)
	p	3.020e-11	4.504e-11	4.975e-11	1.206e10	–
	h	1	1	1	1	–
DF8	(5,20)	5.631e-2(1.418e-3)	8.208e-2(4.023e-4)	4.055e-2(7.530e-4)	4.636e-2(1.662e-3)	3.695e-2(3.163e-4)
	p	1.004e-3	9.941e-1	6.669e-3	1.861e-6	–
	h	1	0	1	1	–
DF9	(5,20)	9.535e-2(1.959e-2)	9.792e-2(2.423e-3)	3.618e-1(3.253e-2)	5.431e-1(1.111e-1)	9.288e-2(5.959e-3)
	p	4.077e-11	1.072e-2	7.088e-8	8.841e-7	–
	h	1	1	1	1	–
DF10	(5,20)	1.877e-1(4.411e-2)	2.804e-1(6.160e-3)	1.228e-1(7.753e-3)	1.931e-1(1.144e-2)	1.106e-1(5.332e-3)
	p	4.060e-2	3.032e-2	3.112e-1	2.783e-7	–
	h	1	1	0	1	–
DF11	(5,20)	6.514e-1(4.128e-4)	2.846e-1(3.159e-2)	6.725e-1(1.669e-3)	6.691e-1(2.447e-3)	6.589e-1(1.662e-3)
	p	2.581e-1	7.192e-5	3.403e-1	3.042e-1	–
	h	0	1	0	0	–
DF12	(5,20)	8.731e-1(3.021e-2)	3.266e-1(1.545e-2)	5.107e-1(2.905e-2)	3.123e-1(1.151e-2)	2.801e-1(6.598e-3)
	p	6.010e-8	5.606e-5	5.072e-10	6.567e-2	–
	h	1	1	1	1	–
DF13	(5,20)	2.530e-1(1.364e-2)	1.659e-1(2.258e-3)	3.256e-1(2.346e-2)	4.148e-1(4.258e-2)	1.577e-1(6.645e-3)
	p	8.352e-8	5.997e-1	7.739e-6	8.153e-11	–
	h	1	0	1	1	–
DF14	(5,20)	1.282e-1(2.850e-3)	7.204e-2(3.136e-4)	1.572e-1(4.965e-2)	1.552e-1(1.963e-2)	5.689e-2(1.817e-3)
	p	1.729e-7	1.585e-4	9.063e-8	5.072e-10	–
	h	1	1	1	1	–
$\ddagger/\dagger/\wr$		10/1/3	9/1/4	10/1/3	13/0/1	–

3.4. Parameter settings

The relevant parameters of all the compared algorithms are same as their original studies, and the population size is set to 100. Each algorithm is executed twenty-five independent runs on each benchmark instance. Besides that, ten percent of the population is used to detect each of the thirty times of change in dynamic optimization.

3.5. Experimental results

Test problems used in this paper involves two important control parameters, the severity of change (n_t) and the frequency of change (τ_t), which take values from {5, 10, 20} for evaluating the robustness of the proposed algorithm. MIGD values and t-test results are presented in Tables 1–3 (MSP, MHV results can be found in Supplementary Material), in which the best results are also highlighted in bold face.

It is observed from the MIGD and t – test values reported in Tables 1–3 that PFPA clearly outperforms the other compared algorithms on majority of the test problems. In Table 1, the statistical p -values illustrate that there are not significant differences among all the algorithms on DF5 and DF11. As the parameters change, the differences become clear for DF10 and DF11. Totally, for different combinations of n_t and τ_t , PFPA can generate best results on most of the problems, which means that the proposed prediction mechanism is able to obtain high-quality populations for tracking the Pareto-optimal front effectively in dynamic environments.

Table 2

Mean and standard deviation values of MIGD obtained by five algorithm for $(n_t, \tau_t) = (10, 10)$.

Fun.	(n_t, τ_t)	MOEA/D-FD	TrDMOEA	DNSGAA	PPS	PFPA
DF1	(10,10)	9.522e–3(1.371e–4)	8.431e–2(9.136e–2)	6.020e–2(4.333e–3)	3.729e–1(6.416e–2)	5.819e–3(2.801e–4)
	p	1.492e–6	1.094e–10	2.439e–9	3.338e–11	–
	h	1	1	1	1	–
DF2	(10,10)	1.097e–2(2.093e–4)	8.149e–3(5.478e–4)	4.289e–2(4.726e–3)	2.261e–1(4.723e–2)	4.932e–2(5.571e–3)
	p	2.253e–4	1.202e–8	7.978e–2	1.698e–8	–
	h	1	1	0	1	–
DF3	(10,10)	3.386e–2(2.023e–3)	3.358e–2(1.542e–2)	2.896e–1(2.940e–3)	1.460e–1(1.323e–1)	8.859e–3(2.957e–3)
	p	1.311e–8	3.255e–7	7.398e–11	1.777e–10	–
	h	1	1	1	1	–
DF4	(10,10)	1.069e–1(1.686e–3)	5.644e–1(1.407e–1)	7.539e–2(2.606e–3)	1.162e–1(1.222e–2)	7.146e–2(1.925e–3)
	p	1.273e–2	1.558e–7	5.395e–1	8.500e–2	–
	h	1	1	1	0	–
DF5	(10,10)	1.455e–2(2.501e–4)	2.583e–2(4.359e–3)	5.705e–2(4.674e–3)	3.628e–1(9.444e–2)	6.819e–3(1.253e–3)
	p	4.311e–8	4.195e–10	4.563e–9	4.504e–11	–
	h	1	1	1	1	–
DF6	(10,10)	5.080e+0(5.327e–1)	1.209e+0(2.704e–1)	3.478e+0(2.177e–1)	7.477e+0(7.384e–1)	6.484e–1(5.727e–2)
	p	4.841e–2	7.482e–2	1.430e–5	6.695e–11	–
	h	1	0	1	1	–
DF7	(10,10)	9.106e–2(1.418e–2)	3.546e–2(9.378e–4)	2.583e–1(6.761e–2)	6.051e–2(1.616e–2)	9.755e–3(2.019e–4)
	p	3.020e–11	6.066e–11	3.020e–11	1.329e–10	–
	h	1	1	1	1	–
DF8	(10,10)	3.053e–2(2.051e–3)	7.954e–2(7.347e–3)	1.164e–2(1.017e–3)	1.569e–2(1.487e–3)	6.980e–3(3.363e–4)
	p	4.504e–11	3.094e–6	2.107e–1	3.183e–3	–
	h	1	1	0	1	–
DF9	(10,10)	8.732e–2(1.473e–2)	7.540e–2(1.771e–2)	3.489e–1(4.264e–2)	4.713e–1(1.280e–1)	6.349e–2(4.895e–3)
	p	9.000e–1	2.708e–2	1.957e–10	8.153e–11	–
	h	0	1	1	1	–
DF10	(10,10)	1.652e–1(3.667e–2)	2.775e–1(1.289e–2)	1.140e–1(8.672e–3)	1.816e–1(1.075e–2)	1.011e–1(4.221e–3)
	p	8.771e–2	2.510e–2	2.458e–1	6.010e–8	–
	h	0	1	0	1	–
DF11	(10,10)	6.373e–1(3.384e–4)	2.877e–1(1.657e–2)	6.575e–1(2.241e–3)	6.551e–1(2.509e–3)	6.417e–1(1.518e–3)
	p	5.592e–1	2.430e–5	5.895e–1	5.395e–1	–
	h	0	1	0	0	–
DF12	(10,10)	9.526e–1(1.738e–2)	3.559e–1(4.370e–2)	5.047e–1(3.102e–2)	3.043e–1(9.512e–3)	2.820e–1(4.677e–3)
	p	5.573e–10	3.965e–8	5.573e–10	1.087e–1	–
	h	1	1	1	0	–
DF13	(10,10)	2.239e–1(5.951e–3)	1.542e–1(7.866e–3)	3.183e–1(2.451e–2)	4.057e–1(2.940e–2)	1.474e–1(5.111e–3)
	p	2.572e–7	3.871e–1	3.835e–6	8.153e–11	–
	h	1	0	1	1	–
DF14	(10,10)	1.221e–1(4.208e–3)	6.943e–2(3.387e–3)	1.435e–1(5.549e–2)	1.620e–1(2.177e–2)	5.314e–2(1.898e–3)
	p	6.528e–8	3.592e–5	1.157e–7	4.200e–10	–
	h	1	1	1	1	–
$\ddagger/\dagger/\ddagger$		10/1/3	11/2/1	10/0/4	11/0/3	–

Table 3Mean and standard deviation values of MIGD obtained by five algorithm for $(n_t, \tau_t) = (10, 20)$.

Fun.	(n_t, τ_t)	MOEA/D-FD	TrDMOEA	DNSGAA	PPS	PFPA
DF1	(10,20)	6.546e–2(1.322e–4)	5.809e–2(1.091e–3)	7.742e–2(8.680e–4)	2.479e–1(5.558e–2)	6.457e–2(3.287e–4)
	<i>p</i>	2.897e–6	5.367e–1	4.033e–3	4.616e–10	–
	<i>h</i>	1	0	1	1	–
DF2	(10,20)	9.222e–3(1.670e–4)	8.197e–3(2.152e–5)	1.955e–2(1.980e–3)	1.508e–1(4.644e–2)	4.218e–2(2.664e–3)
	<i>p</i>	4.975e–11	1.359e–7	6.066e–11	9.521e–4	–
	<i>h</i>	1	1	1	1	–
DF3	(10,20)	9.217e–2(1.145e–3)	8.895e–2(1.175e–3)	4.653e–1(4.629e–2)	1.662e–1(1.457e–1)	1.188e–1(1.303e–1)
	<i>p</i>	3.020e–11	3.601e–11	3.020e–11	9.756e–10	–
	<i>h</i>	1	1	1	1	–
DF4	(10,20)	4.451e–1(5.528e–3)	5.588e–1(1.801e–2)	4.115e–1(1.255e–2)	4.808e–1(1.193e–2)	4.398e–1(4.105e–3)
	<i>p</i>	1.850e–8	4.625e–2	1.596e–7	1.248e–4	–
	<i>h</i>	1	1	1	1	–
DF5	(10,20)	2.580e–2(1.771e–4)	3.118e–2(2.894e–4)	3.021e–2(5.962e–4)	1.044e–1(2.204e–2)	2.195e–2(4.480e–4)
	<i>p</i>	3.368e–4	4.035e–1	2.282e–1	6.722e–10	–
	<i>h</i>	1	0	0	1	–
DF6	(10,20)	3.235e+0(1.072e+0)	1.953e+0(2.570e–1)	1.692e+0(2.568e–1)	4.032e+0(6.981e–1)	3.864e–1(4.386e–2)
	<i>p</i>	2.371e–10	5.841e–6	4.573e–9	1.174e–9	–
	<i>h</i>	1	1	1	1	–
DF7	(10,20)	1.241e–1(1.096e–2)	7.679e–2(9.822e–3)	1.737e–1(3.238e–2)	8.795e–2(8.222e–3)	5.530e–2(3.028e–4)
	<i>p</i>	3.020e–11	3.338e–11	3.020e–11	1.311e–8	–
	<i>h</i>	1	1	1	1	–
DF8	(10,20)	1.374e–1(3.063e–3)	8.472e–2(2.789e–5)	1.319e–1(4.483e–3)	1.390e–1(3.522e–3)	1.303e–1(6.972e–4)
	<i>p</i>	5.012e–2	1.174e–4	6.787e–2	7.119e–9	–
	<i>h</i>	0	1	0	1	–
DF9	(10,20)	7.399e–2(1.394e–3)	6.534e–2(1.396e–2)	2.251e–1(4.792e–2)	2.970e–1(1.167e–1)	6.004e–2(3.336e–3)
	<i>p</i>	3.690e–11	1.297e–1	5.186e–7	2.921e–2	–
	<i>h</i>	1	0	1	1	–
DF10	(10,20)	3.147e–1(1.227e–2)	2.867e–1(1.412e–2)	2.632e–1(5.029e–3)	2.491e–1(7.535e–3)	2.606e–1(4.127e–3)
	<i>p</i>	1.154e–1	9.470e–1	5.793e–1	9.705e–1	–
	<i>h</i>	0	0	0	0	–
DF11	(10,20)	7.480e–1(3.787e–4)	2.507e–1(2.641e–2)	7.620e–1(1.107e–3)	7.639e–1(1.988e–3)	7.560e–1(1.422e–3)
	<i>p</i>	5.692e–1	1.148e–7	6.309e–1	5.895e–1	–
	<i>h</i>	0	1	0	0	–
DF12	(10,20)	9.348e–2(2.987e–2)	3.369e–1(1.431e–2)	4.763e–1(2.504e–2)	3.199e–1(1.016e–2)	3.149e–1(5.715e–3)
	<i>p</i>	6.121e–10	3.339e–3	5.462e–9	3.183e–1	–
	<i>h</i>	1	1	1	0	–
DF13	(10,20)	2.782e–1(1.351e–2)	1.715e–1(2.208e–3)	1.777e–1(6.624e–3)	2.996e–1(1.908e–2)	1.681e–1(4.608e–3)
	<i>p</i>	1.202e–8	5.895e–1	7.618e–1	2.572e–7	–
	<i>h</i>	1	0	0	1	–
DF14	(10,20)	1.427e–1(4.208e–3)	7.995e–2(7.074e–3)	7.382e–2(3.798e–3)	1.095e–1(9.542e–3)	6.729e–2(2.492e–3)
	<i>p</i>	5.967e–9	3.183e–3	7.483e–2	8.841e–7	–
	<i>h</i>	1	1	0	1	–
†/‡/‡		9/2/3	6/5/3	6/2/6	11/0/3	–

Tables 1–3 (presented in [Supplementary Material](#)) present the MHV and *t* – test values of all the optimization algorithms, and demonstrate that when n_t is five, PFPA performs much better than MOEADFD, PPS, NSGAA on the most of the problems, and slightly worse than TrDMOEA on DF5, DF11, DF12 according to the *p*-values. With the change of the parameters, the proposed algorithm still retains significant advantage and is competitive with TrDMOEA. Therefore, in terms of this metric, PFPA has stable and fast responses to dynamic changes.

It can be seen from the MSP and *t* – test values, shown in **Tables 4–6** (presented in [Supplementary Material](#)), that PFPA has a significant advantage over the other algorithms on some bi-objective problems (e.g., DF1, DF7 and DF8) and performs ineffectively on tri-objective problems. The frequency and severity of change has not obvious impact on all the algorithms. The reason is that most of the algorithms have a good diversity maintenance mechanism (e.g., the decomposition framework for MOEADFD). Pareto-dominance based MOEAs do not seem to be effective for generating uniformly distributed solutions, especially in three-objective cases. However, it is necessary to measure the overall effectiveness of an algorithm from multiple metrics rather than one indicator. Therefore, although PFPA does not achieve good MSP values, it has significant advantages on other metrics.

Some convergence curves of MIGD values are provided in [Fig. 4](#), which shows that the difference is small between PFPA and DNSGAA on DF4, but PFPA outperforms other algorithms in terms of convergence lines. For DF10, the proposed algorithm has a significant advantage in the late stage of environmental change. For other problems, PFPA has competitive performance. Overall, PFPA has more stable convergence and responds faster to changes than other compared algorithms on most of the problems.

Table 4

Performance comparison of different PFPA variants on MIGD

Fun.	(n_t, τ_t)	PFPAV1	PFPAV2	PFPAV3	PFPA
DF1	(10,10)	1.414e-2(4.494e-4)	1.431e-2(1.193e-3)	1.713e-2(2.824e-3)	5.819e-3(2.801e-4)
	p	1.957e-10	1.957e-10	2.610e-10	–
	h	1	1	1	–
DF2	(10,10)	1.074e-1(6.235e-3)	1.092e-1(5.436e-3)	8.080e-2(1.079e-2)	4.932e-2(5.571e-3)
	p	2.959e-5	7.221e-6	7.959e-3	–
	h	1	1	1	–
DF3	(10,10)	3.010e-2(3.089e-3)	8.484e-2(1.683e-1)	1.285e-2(6.089e-4)	8.859e-3(2.957e-3)
	p	2.439e-9	5.072e-10	2.028e-7	–
	h	1	1	1	–
DF4	(10,10)	3.421e-1(4.022e-2)	3.379e-1(4.501e-2)	7.539e-2(3.266e-3)	7.146e-2(1.925e-3)
	p	9.063e-8	1.067e-7	3.555e-1	–
	h	1	1	0	–
DF5	(10,10)	1.962e-2(1.145e-3)	1.869e-2(1.608e-3)	1.427e-2(1.311e-3)	6.819e-3(1.253e-3)
	p	9.756e-10	1.734e-9	2.195e-8	–
	h	1	1	1	–
DF6	(10,10)	6.377e-1(3.884e-2)	6.714e-1(5.660e-2)	4.336e+0(2.895e+0)	6.484e-1(5.727e-2)
	p	3.671e-1	3.122e-1	1.023e-1	–
	h	0	0	0	–
DF7	(10,10)	2.674e-2(9.112e-4)	2.761e-2(1.256e-3)	2.033e-2(6.670e-4)	9.755e-3(2.019e-4)
	p	3.020e-11	3.020e-11	1.464e-10	–
	h	1	1	1	–
DF8	(10,10)	2.944e-2(2.595e-3)	3.120e-2(2.704e-3)	1.333e-2(5.485e-4)	6.980e-3(3.363e-4)
	p	3.020e-11	3.020e-11	1.411e-9	–
	h	1	1	1	–
DF9	(10,10)	1.271e-1(4.815e-3)	1.289e-1(5.378e-3)	2.430e-1(8.753e-2)	6.349e-2(4.895e-3)
	p	1.892e-4	4.639e-5	1.167e-5	–
	h	1	1	1	–
DF10	(10,10)	1.648e-1(1.005e-2)	2.928e-1(3.636e-2)	1.017e-1(5.247e-3)	1.011e-1(4.221e-3)
	p	4.077e-11	3.020e-11	7.845e-1	–
	h	1	1	0	–
DF11	(10,10)	6.704e-1(1.363e-3)	6.947e-1(7.635e-3)	6.462e-1(1.197e-3)	6.417e-1(1.518e-3)
	p	4.918e-1	4.204e-1	8.534e-1	–
	h	0	0	0	–
DF12	(10,10)	3.384e-1(6.099e-3)	3.919e-1(1.369e-2)	2.839e-1(7.839e-3)	2.820e-1(4.677e-3)
	p	1.194e-6	6.010e-8	8.303e-1	–
	h	1	1	0	–
DF13	(10,10)	2.602e-1(1.897e-2)	5.469e-1(1.226e-1)	1.540e-1(6.515e-3)	1.474e-1(5.111e-3)
	p	1.777e-10	4.077e-11	4.305e-1	–
	h	1	1	0	–
DF14	(10,10)	4.368e-1(1.382e-2)	1.342e-1(7.323e-3)	5.106e-2(1.661e-3)	5.314e-2(1.898e-3)
	p	3.197e-9	5.967e-9	2.973e-1	–
	h	1	1	0	–
‡/†/‡		11/0/2	11/0/2	7/0/7	–

Fig. 5–8 present some POF approximations. It can be found that the individuals obtained by PFPA are closer to the real POF and are evenly distributed on DF3. For DF5, the PFPA algorithm can obtain individuals with good distribution, but there is a gap between the obtained solutions and the true POF. In later changes, the individuals obtained by PFPA are close to the POF with good distribution, and the convergence is obviously better than the other algorithms. The approximation of PFPA is significantly better than the other algorithms on DF7 and DF8. Thus, PFPA have a good approximation to the changing POF, which means that the designed algorithm has potential tracking ability in changing environments.

4. Discussion

4.1. Component analysis

As mentioned before, the proposed dynamic response mechanism contains three different key strategies for generating high-quality population and tracking the changing Pareto front effectively. This subsection focuses on the impact of each strategy on the search performance of the proposed algorithm. Specifically, to illustrate the role of multi-step movement strategy, PAPFV1 is designed by integrating a one-step movement model which is widely employed in most existing prediction approaches for dynamism handling. PFPAV2 and PFPAV3 are designed by removing the polynomial fitting strategy and sampling strategy for showing the effect of the corresponding techniques, respectively. PFPA is also utilized to compare with these three modified versions. Table 4 presents the experimental and comparison results in detail.

Table 5Performance comparison of PFPA variants on MIGD for $(n_t, \tau_t) = (10, 10)$.

Fun.	(n_t, τ_t)	PFPA S1	PFPA S2	PFPA S3	PFPA
DF1	(10,10)	1.106e-2(3.976e-4)	1.104e-2(4.503e-4)	1.111e-2(6.105e-4)	5.819e-3(2.801e-4)
	p	1.698e-8	2.390e-7	2.390e-7	–
	h	1	1	1	–
DF2	(10,10)	8.010e-2(7.415e-3)	8.261e-2(7.062e-3)	8.285e-2(4.883e-3)	4.932e-2(5.571e-3)
	p	2.006e-4	3.831e-5	5.607e-5	–
	h	1	1	1	–
DF3	(10,10)	3.676e-2(1.423e-2)	5.179e-2(1.475e-1)	8.857e-2(2.237e-1)	8.859e-3(2.957e-3)
	p	1.311e-8	6.528e-8	3.497e-9	–
	h	1	1	1	–
DF4	(10,10)	3.421e-1(4.022e-2)	7.506e-2(2.561e-3)	7.440e-2(2.222e-3)	7.146e-2(1.925e-3)
	p	5.298e-3	4.464e-1	3.790e-1	–
	h	1	0	0	–
DF5	(10,10)	1.224e-2(5.607e-4)	1.230e-2(7.611e-4)	1.164e-2(4.619e-4)	6.819e-3(1.253e-3)
	p	1.102e-8	2.015e-8	2.390e-8	–
	h	1	1	1	–
DF6	(10,10)	5.341e-1(5.014e-2)	4.911e-1(4.886e-2)	4.951e-1(3.154e-2)	6.484e-1(5.727e-2)
	p	4.643e-1	6.309e-1	6.414e-1	–
	h	0	0	0	–
DF7	(10,10)	1.767e-2(4.501e-4)	1.829e-2(1.002e-3)	1.938e-2(3.061e-3)	9.755e-3(2.019e-4)
	p	3.020e-11	3.020e-11	3.020e-11	–
	h	1	1	1	–
DF8	(10,10)	1.230e-2(4.067e-4)	1.232e-2(3.584e-4)	1.258e-2(4.527e-4)	6.980e-3(3.363e-4)
	p	8.485e-9	7.119e-9	7.119e-9	–
	h	1	1	1	–
DF9	(10,10)	7.469e-2(7.412e-3)	6.887e-2(5.683e-3)	7.263e-2(4.585e-3)	6.349e-2(4.895e-3)
	p	5.746e-2	1.958e-1	9.926e-2	–
	h	1	0	0	–
DF10	(10,10)	1.689e-1(1.147e-2)	1.643e-1(9.155e-3)	1.618e-1(5.483e-3)	1.011e-1(4.221e-3)
	p	2.610e-10	1.329e-10	2.371e-10	–
	h	1	1	1	–
DF11	(10,10)	6.653e-1(4.538e-3)	6.643e-1(2.673e-3)	6.656e-1(3.896e-3)	6.417e-1(1.518e-3)
	p	5.011e-1	5.201e-1	5.201e-1	–
	h	0	0	0	–
DF12	(10,10)	3.329e-1(6.786e-3)	3.203e-1(8.256e-3)	3.210e-1(7.308e-3)	2.820e-1(4.677e-3)
	p	6.765e-5	3.831e-5	2.433e-5	–
	h	1	1	1	–
DF13	(10,10)	2.932e-1(1.864e-2)	2.581e-1(1.670e-2)	2.475e-1(1.613e-2)	1.474e-1(5.111e-3)
	p	2.154e-10	8.101e-10	4.573e-9	–
	h	1	1	1	–
DF14	(10,10)	4.323e-1(1.675e-2)	8.260e-2(4.528e-3)	8.180e-2(4.225e-3)	5.314e-2(1.898e-3)
	p	2.195e-8	2.317e-6	7.739e-6	–
	h	1	1	1	–
†/‡/‡		12/0/2	10/0/4	10/0/4	–

4.1.1. Multi-step movement strategy

It is observed from Table 4 that, although there exists some similar results indicated by the *p*-values, PFPA performs much better than the other three variants on the majority of the test problems. This shows that the multi-step strategy is able to generate more better individuals close to the real POS and improves the diversity of the population and the search efficiency of the algorithm. Therefore, it can be seen that this strategy is an important part of the proposed algorithm. It is worth noting that this strategy is likely to generate boundary solutions, which may not benefit for global search. This strategy should be improved in future work.

4.1.2. Polynomial fitting-based Strategy

Table 4 clearly demonstrates that there are not differences between PFPA and PFPAV2 on DF6 and DF11, but PFPA is much superior to PFPAV2 on eleven benchmark problems, which means that this strategy is indeed able to improve the search performance of PFPA. This could be explained by the fact that this strategy fully considers the distribution relationship between variables, which is helpful to generate promising solutions to some extent.

4.1.3. Sampling strategy

This strategy aims to make full use of the relationship between variables to generate high-quality individuals to guide the search, thus speeding up the search efficiency of the algorithm. The experimental results also show that this strategy can improve the search performance of the algorithm to some extent in varying environments.

Table 6Performance comparison of PFPA variants on MIGD for $(n_t, \tau_t) = (10, 10)$.

Fun.	(n_t, τ_t)	PFPAR1	PFPAR2	PFPAR3	PFPA
DF1	(10,10)	1.202e-2(6.330e-4)	1.248e-2(5.582e-4)	1.326e-2(8.466e-4)	5.819e-3(2.801e-4)
	p	6.518e-9	4.998e-9	1.411e-9	–
	h	1	1	1	–
DF2	(10,10)	6.887e-2(4.200e-3)	7.040e-2(8.178e-3)	6.654e-2(8.329e-3)	4.932e-2(5.571e-3)
	p	2.531e-4	2.839e-4	5.264e-4	–
	h	1	1	1	–
DF3	(10,10)	1.531e-2(3.077e-3)	2.130e-2(1.080e-2)	3.235e-2(4.975e-2)	8.859e-3(2.957e-3)
	p	1.606e-6	6.526e-7	3.646e-8	–
	h	1	1	1	–
DF4	(10,10)	7.605e-2(3.138e-3)	7.732e-2(3.412e-3)	7.663e-2(2.257e-3)	7.146e-2(1.925e-3)
	p	3.871e-1	3.255e-1	3.790e-1	–
	h	0	0	0	–
DF5	(10,10)	1.416e-2(4.750e-4)	1.610e-2(8.475e-4)	1.755e-2(1.320e-3)	6.819e-3(1.253e-3)
	p	8.485e-9	6.518e-9	2.439e-9	–
	h	1	1	1	–
DF6	(10,10)	6.180e-1(4.740e-2)	6.473e-1(4.496e-2)	6.606e-1(4.750e-2)	6.484e-1(5.727e-2)
	p	8.500e-2	5.188e-2	4.841e-2	–
	h	0	0	1	–
DF7	(10,10)	1.787e-2(6.005e-4)	1.798e-2(7.292e-4)	1.844e-2(6.913e-4)	9.755e-3(2.019e-4)
	p	3.020e-11	3.020e-11	3.020e-11	–
	h	1	1	1	–
DF8	(10,10)	1.399e-2(4.383e-3)	1.558e-2(9.046e-4)	1.787e-2(1.247e-3)	6.980e-3(3.363e-4)
	p	1.174e-9	4.200e-10	1.094e-10	–
	h	1	1	1	–
DF9	(10,10)	7.893e-2(5.033e-3)	8.880e-2(4.479e-3)	9.118e-2(8.936e-3)	6.349e-2(4.895e-3)
	p	1.837e-2	2.380e-3	6.912e-4	–
	h	1	1	1	–
DF10	(10,10)	1.304e-1(6.197e-3)	1.282e-1(6.893e-3)	1.344e-1(7.392e-3)	1.011e-1(4.221e-3)
	p	1.596e-7	2.377e-7	3.081e-8	–
	h	1	1	1	–
DF11	(10,10)	6.497e-1(1.566e-3)	6.486e-1(2.148e-3)	6.494e-1(1.646e-3)	6.417e-1(1.518e-3)
	p	6.627e-1	7.062e-1	6.414e-1	–
	h	0	0	0	–
DF12	(10,10)	3.113e-1(7.675e-3)	3.075e-1(4.426e-3)	3.020e-1(6.396e-3)	2.820e-1(4.677e-3)
	p	6.912e-4	1.518e-3	6.972e-3	–
	h	1	1	1	–
DF13	(10,10)	1.904e-1(5.077e-3)	1.900e-1(1.015e-1)	1.885e-1(6.361e-3)	1.474e-1(5.111e-3)
	p	2.678e-6	4.421e-6	2.491e-6	–
	h	1	1	1	–
DF14	(10,10)	6.050e-2(1.628e-3)	6.061e-2(1.759e-3)	6.183e-2(1.552e-3)	5.314e-2(1.898e-3)
	p	1.383e-2	9.883e-3	3.501e-3	–
	h	1	1	1	–
†/‡/‡		11/0/3	12/0/2	12/0/2	–

4.2. Parameter analysis

In addition to the above discussion, parameter analysis is also necessary. The proposed algorithm mainly includes two key control parameters, i.e., the step size (*step*) involved in the multi-step movement strategy and the compression ratio (*cr*) involved in the polynomial fitting strategy. The relevant comparison results are shown in Table 5 in detail.

4.2.1. Influence of step values

As described before, the multi-step movement strategy utilizes two different stepsizes for generating promising solutions. Here, to explore the influence of the step values, *step* = 1 is fixed as it has proven effective in many prediction algorithms, and other step values are taken from the range [0.1, 0.7] (PFPAS1-PFPAS3), with an increment of 0.2. It can be observed from Table 5 that, although the differences among them are not significant on some cases, PFPA performs better than the other modifications, according to the *p*-value results. Therefore, 1 and 0.3 are utilized as the movement stepsizes in PFPA.

4.2.2. Influence of cr values

Eq. (7) has an important parameter, i.e., compression ratio (*cr*), for generating possible better solutions for new environments. Here, *cr* is varied from 0.1 to 1 (PFPAR1-PFPAR3) with an increment of 0.3, and the results are summarized in Table 6, which illustrate that the proposed algorithm outperforms the other three versions on almost all the problems significantly. Therefore, this paper uses 0.1 as the best *cr* value for the polynomial fitting strategy.

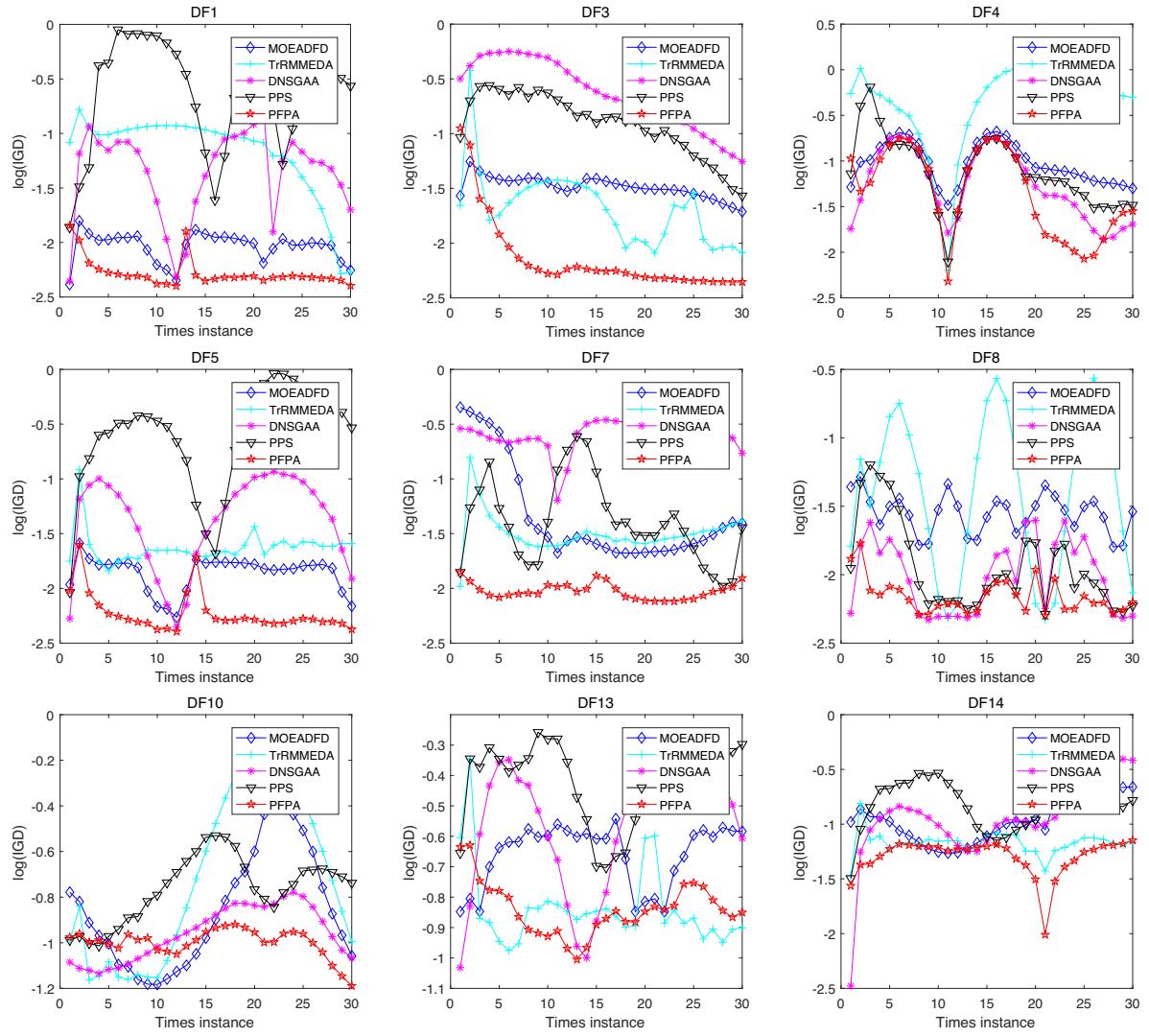


Fig. 4. Mean IGD curves for different problems with $n_t = 10$ and $\tau_t = 10$.

4.3. Different multi-objective algorithms

The main purpose of this subsection is to verify the feasibility of the proposed strategies by integrating them into three different and recently proposed multiobjective algorithms. The experimental results can be found in Table 7 and Tables 7.8 (provided in [Supplementary Material](#)). It is obvious that these algorithms are able to obtain effective results, but are not as effective as PFPA. Therefore, the proposed algorithm is suitable to be used with different algorithms to solve dynamic multi-objective optimization problems.

5. Experimental results on applications

To further evaluate the effectiveness of the proposed PFPA in practice, two widely used dynamic multi-objective engineering designed problems are adopted [36]. Note that the stopping condition, the number of runs and the other relevant settings remain the same as described in the previous sections, and a constraint handling technique [49] is also utilized to deal with the constraints that the problems have. In addition, different from the benchmark problems, the true POS/POF cannot be known in advance in real applications, so MIGD and MSP cannot be used as the performance indicators. However, the MHV can be still used, and the relevant experimental results are summarized in Tables 8,9.

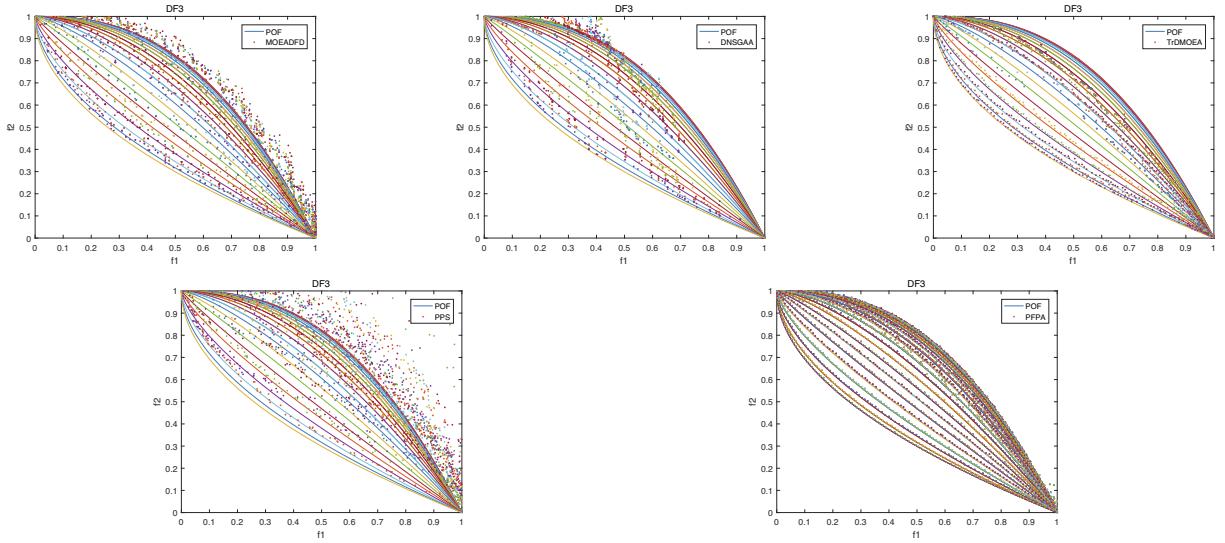


Fig. 5. POF approximations of five algorithms for D F3 with $n_t = 10$ and $\tau_t = 10$.

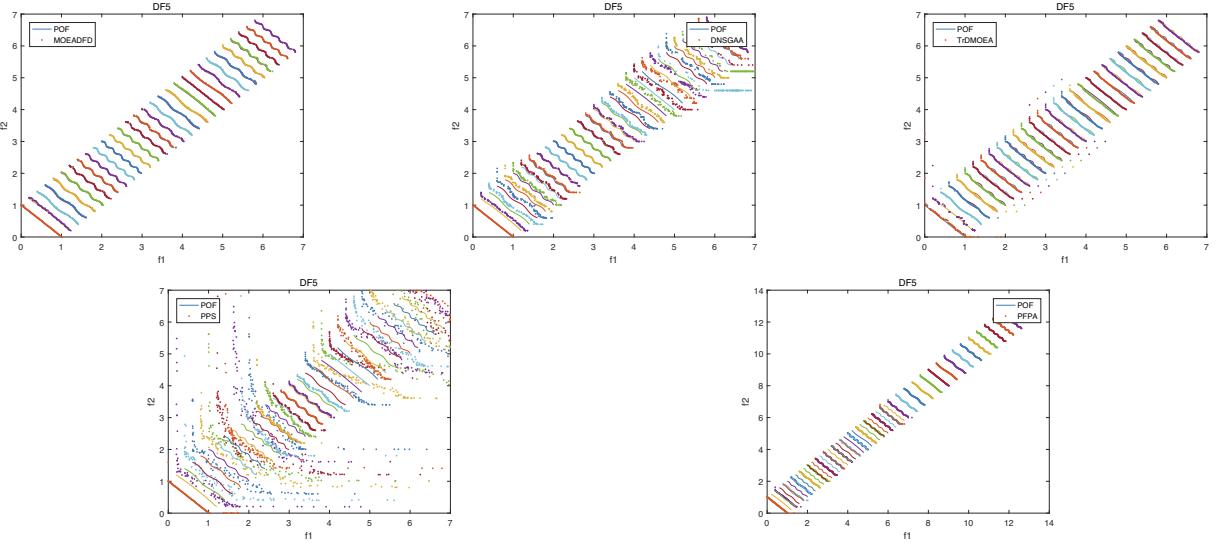


Fig. 6. POF approximations of five algorithms for DF5 with $n_t = 10$ and $\tau_t = 10$.

5.1. Case 1

The welded beam design problem was modified and proposed by integrating some time instances and dynamic factors into it to form the dynamic welded beam design problem, which consists of two important objectives: minimum total fabricating cost and minimum bar end deflection. As can be seen in the following mathematical model, there are four decision variables (x_1, x_2, x_3, x_4) and four constraints.

$$\begin{aligned}
 \text{Min. } & f_1(x, t) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \\
 \text{Min. } & f_2(x, t) = \frac{4P(t)14^3}{Ex_3^2x_4} \\
 \text{s.t. } & g_1(x) = \tau - 13,600 \leq 0 \\
 & g_2(x) = \frac{6P(t)L}{x_4x_3^2} - 30,000 \leq 0 \\
 & g_3(x) = x_1 - x_4 \leq 0 \\
 & g_4(x) = P(t) - pc \leq 0
 \end{aligned} \tag{9}$$

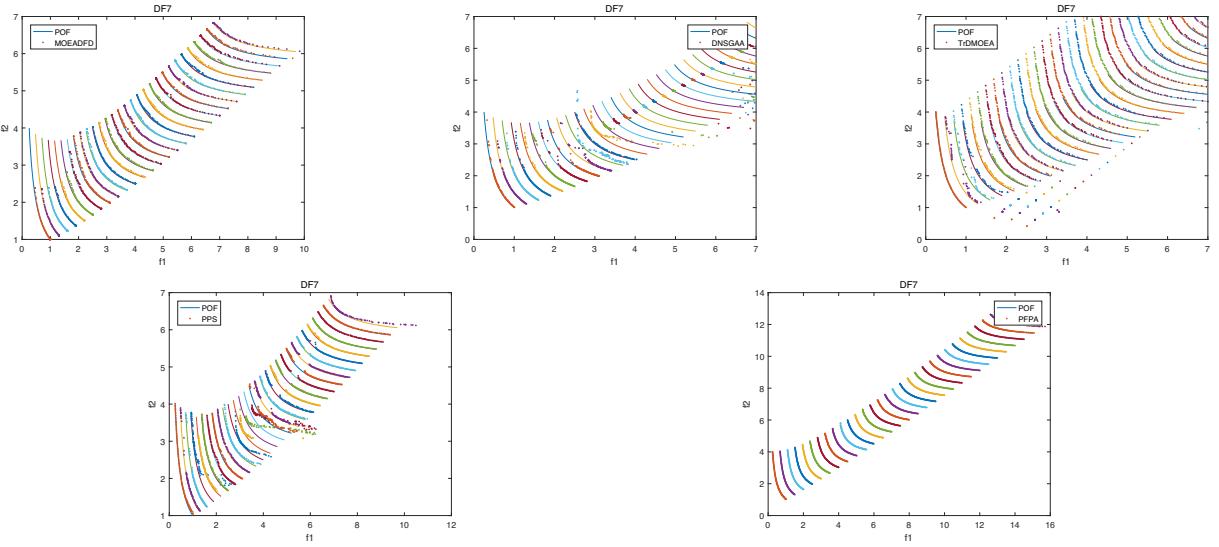


Fig. 7. POF approximations of five algorithms for DF7 with $n_t = 10$ and $\tau_t = 10$.

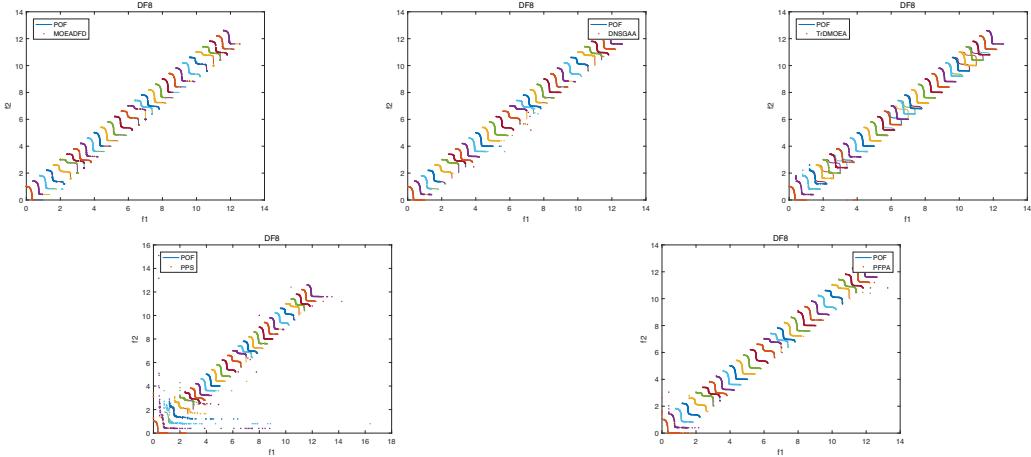


Fig. 8. POF approximations of five algorithms for DF8 with $n_t = 10$ and $\tau_t = 10$.

$$\begin{aligned}
 PC &= \frac{4.013E\sqrt{\frac{x_3^2 x_4^6}{36}}}{14^2} \left(1 - \left(\frac{x_3}{28}\sqrt{\left(\frac{E}{4G}\right)}\right)\right), R = \sqrt{\left(\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2\right)}, J = 2\sqrt{(2)x_1x_2\left(\frac{x_2^2}{12} + \left(\frac{x_1+x_3}{2}\right)^2\right)}, M = P(t)(14 + x_2/2), \tau_2 = \frac{MR}{J}, \tau_1 = \frac{P(t)}{\sqrt{(2)x_1x_2}}, \\
 \tau &= \sqrt{(\tau_1^2 + 2\tau_1\tau_2\frac{x_3}{2R} + \tau_2^2)}, E = 3 \times 10^7, G = 1.2 \times 10^7, (t, P(t)) = (1, 10000), (2, 8000), (3, 6000), (4, 3000), 55 \leq x_1 \leq 80, 75 \leq x_2 \leq 110, 1000 \leq x_3 \leq 3000, 2 \leq x_4 \leq 20
 \end{aligned}$$

According to the experimental results listed in Table 8, although the standard deviation value of PFPA is worse than that of PPS and TrDMOEAs, it outperforms the rest of the algorithms. There is not significant difference between the algorithms except MOEADFD. Such results indicate that the proposed algorithm is an attractive alternative optimizer for generating satisfactory results on challenging optimization problems.

5.2. Case 2

As one of the most popular problems in the field of mechanical engineering, the dynamic speed reducer design problem aims to optimize the weight (f_1) and stress (f_2), which contains of eleven constraints and involves seven decision variables as follows, gear face width (x_1), teeth module (x_2), number of teeth of pinion (x_3), distance between bearings (x_4), distance between bearings 2 (x_5), diameter of shaft 1 (x_6), and diameter of shaft 2 (x_7).

Table 7Experimental results comparison of different multiobjective optimization algorithms on MIGD for $(n_t, \tau_t) = (10, 10)$.

Fun.	(n_t, τ_t)	MOMVO	MOALO	MOGOA	PFPA
DF1	(10,10)	1.182e-2(4.707e-4)	3.670e-2(2.492e-3)	4.737e-2(8.374e-3)	5.819e-3(2.801e-4)
	p	1.157e-7	7.380e-10	3.197e-9	–
	h	1	1	1	–
DF2	(10,10)	1.279e-1(9.046e-3)	2.156e-1(1.932e-2)	1.903e-1(1.041e-2)	4.932e-2(5.571e-3)
	p	1.492e-6	1.206e-10	2.872e-10	–
	h	1	1	1	–
DF3	(10,10)	6.044e-2(1.959e-2)	6.217e-2(9.415e-3)	7.898e-2(1.321e-2)	8.859e-3(2.957e-3)
	p	2.670e-9	1.287e-9	1.174e-9	–
	h	1	1	1	–
DF4	(10,10)	9.836e-2(5.237e-3)	2.641e-1(4.634e-2)	1.952e-1(2.347e-2)	7.146e-2(1.925e-3)
	p	1.335e-1	7.200e-5	9.792e-5	–
	h	0	1	1	–
DF5	(10,10)	1.089e-2(4.366e-4)	4.095e-2(3.865e-3)	3.651e-2(2.985e-3)	6.819e-3(1.253e-3)
	p	1.407e-4	6.518e-9	1.429e-8	–
	h	1	1	1	–
DF6	(10,10)	6.693e-1(4.121e-2)	1.4932e+0(1.701e-1)	1.812e+0(3.526e-1)	6.484e-1(5.727e-2)
	p	2.707e-1	3.387e-2	8.771e-2	–
	h	0	1	0	–
DF7	(10,10)	4.955e-2(9.104e-3)	4.825e-2(5.287e-3)	5.797e-2(6.698e-3)	9.755e-3(2.019e-4)
	p	3.338e-11	3.020e-11	3.020e-11	–
	h	1	1	1	–
DF8	(10,10)	1.417e-2(1.117e-3)	2.617e-2(2.881e-3)	2.617e-2(2.368e-3)	6.980e-3(3.363e-4)
	p	5.874e-4	8.841e-7	1.748e-5	–
	h	1	1	1	–
DF9	(10,10)	1.530e-1(7.793e-3)	1.862e-1(6.129e-3)	1.879e-1(1.085e-2)	6.349e-2(4.895e-3)
	p	2.028e-7	1.202e-8	3.081e-8	–
	h	1	1	1	–
DF10	(10,10)	1.117e-1(6.022e-3)	1.566e-1(7.006e-3)	1.449e-1(1.396e-2)	1.011e-1(4.221e-3)
	p	2.159e-1	9.919e-11	2.439e-9	–
	h	0	1	1	–
DF11	(10,10)	6.582e-1(2.923e-3)	6.874e-1(7.728e-3)	7.098e-1(8.849e-3)	6.417e-1(1.518e-3)
	p	5.997e-1	4.733e-1	3.478e-1	–
	h	0	0	0	–
DF12	(10,10)	3.650e-1(2.923e-3)	3.941e-1(2.795e-2)	3.350e-1(2.088e-2)	2.820e-1(4.677e-3)
	p	8.352e-8	4.421e-6	1.236e-3	–
	h	1	1	1	–
DF13	(10,10)	1.152e-1(5.209e-3)	1.130e+0(3.478e-1)	6.621e-1(3.637e-1)	1.474e-1(5.111e-3)
	p	5.859e-6	1.329e-10	2.439e-9	–
	h	1	1	1	–
DF14	(10,10)	5.116e-2(2.540e-3)	4.655e-1(7.299e-1)	1.994e-1(3.309e-1)	5.314e-2(1.898e-3)
	p	5.692e-1	3.020e-11	3.825e-9	–
	h	0	1	1	–
‡/†/‡		8/0/6	13/0/1	12/0/2	–

Table 8

Experimental results considering MHV indicator for dynamic Welded Beam Design Problem

Index	MOEADFD	TrDMOEA	DNSGAA	PPS	PFPA
MHV	2.018e+0	2.316e+0	2.275e+0	2.317e+0	2.318e+0
std	1.506e-2	4.623e-3	5.213e-3	4.944e-3	5.156e-3
p	2.857e-2	8.857e-1	1.143e-1	9.015e-1	–
h	1	0	0	0	–

Table 9

Experimental results considering MHV indicator for Dynamic Speed Reducer Problem

Index	MOEADFD	TrDMOEA	DNSGAA	PPS	PFPA
MHV	5.604e+5	5.426e+5	2.265e+6	5.405e+5	5.505e+5
std	4.018e+4	4.529e+4	3.618e+2	1.547e+4	1.617e+4
p	9.651e-1	8.993e-1	8.571e-2	9.323e-1	–
h	0	0	0	0	–

$$\begin{aligned}
\text{Min.} f_1(x, t) &= a_1(t)x_1x_2^2(10/3x_3^2 + a_2(t)x_3 - a_3(t)) \\
&\quad - a_4(t)x_1(x_6^2 + x_7^2) + a_5(t)(x_6^3 + x_7^3) \\
&\quad + a_6(t)(x_4x_6^2 + x_5 + x_7^2) \\
\text{Min.} f_2(x, t) &= \frac{\sqrt{(745x_4/x_2x_3)^2 + 1.69 \times 10^7}}{a_7(t)x_6^3} \\
\text{s.t. } g_1(x) &= \frac{1}{x_1x_2^2x_3} - \frac{1}{27} \leq 0 \\
g_2(x) &= \frac{1}{x_1x_2^2x_3^2} - \frac{1}{397.5} \leq 0 \\
g_3(x) &= \frac{x_4^3}{x_2x_3x_6^4} - \frac{1}{1.93} \leq 0 \\
g_4(x) &= \frac{x_5^3}{x_2x_3x_7^4} - \frac{1}{1.93} \leq 0 \\
g_5(x) &= x_2x_3 - 40 \leq 0 \\
g_6(x) &= x_1/x_2 - 12 \leq 0 \\
g_7(x) &= 5 - x_1/x_2 \leq 0 \\
g_8(x) &= 1.9 - x_4 + 1.5x_6 \leq 0 \\
g_9(x) &= 1.9 - x_5 + 1.1x_1 \leq 0 \\
g_{10}(x) &= f_1(x, t) \leq 4300 \\
g_{11}(x) &= \frac{\sqrt{(745x_5/x_2x_3)^2 + 1.515 \times 10^8}}{0.1x_7^3} \leq 1100
\end{aligned} \tag{10}$$

$$a_j(t) = a_j + \frac{t}{10}, 1 \leq j \leq 6, a_7(t) = a_7 + 0.25 - \frac{1}{t+4}, a_1 = 0.7854, a_2 = 14.933, a_3 = 43.0934, a_4 = 1.508, a_5 = 7.477, a_6 = 0.7854, a_7 = 0.1, 2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28, 7.3 \leq x_4 \leq 8.3, 7.3 \leq x_5 \leq 8.3, 2.9 \leq x_6 \leq 3.9, 5.0 \leq x_7 \leq 5.5$$

Table 9 clearly shows that PFPA is slightly worse than that of DNSGAA, but it is competitive with the other competitors in terms of *MHV*. The t-test values indicate that there is no significant difference between them. That means PFPA can be an alternative to existing approaches for handling complex dynamic multi-objective problems in real-world applications.

6. Conclusion

In this paper, a polynomial fitting mechanism is employed for designing a dynamic multi-objective optimization algorithm, named PFPA, which includes three key parts for generating a high-quality search population after change detection. Each component plays an important role in improving population diversity and convergence during the course of optimization. Experimental results verified on a set of benchmark problems with various difficulties and two classical dynamic engineering design problems demonstrate that PFPA has competitive tracking ability compared with some state-of-the-art algorithms. In addition, each component and control parameter in the proposed algorithm is also analysed and discussed extensively. In our future work, prediction techniques and parameter settings will be further investigated.

CRediT authorship contribution statement

Qingyang Zhang: Conceptualization, Writing – review & editing, Funding acquisition, Resources. **Xiangyu He:** Methodology. **Shengxiang Yang:** Supervision, Project administration, Funding acquisition, Writing – review & editing. **Yongquan Dong:** Supervision, Project administration, Funding acquisition, Writing – review & editing. **Hui Song:** Formal analysis, Investigation. **Shouyong Jiang:** Supervision, Project administration, Funding acquisition, Writing – review & editing.

Data availability

Data will be made available on request.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants 62006103 and 61872168, in part by the Jiangsu national science research of high education under Grand 20KJB110021, and in part by the Royal Society International Exchanges Scheme IEC|NSFC|211404.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at <https://doi.org/10.1016/j.ins.2022.08.020>.

References

- [1] K. Li, K. Deb, Q. Zhang, An evolutionary many-objective optimization algorithm based on dominance and decomposition, *IEEE Trans. Evol. Comput.* 19 (5) (2014) 694–176.
- [2] S. Jiang, S. Yang, A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization, *IEEE Trans. Evol. Comput.* 21 (1) (2016) 65–82.
- [3] S. Gee, K. Tan, C. Alippi, Solving multiobjective optimization problems in unknown dynamic environments: An inverse modeling approach, *IEEE Trans. Cybern.* 47 (12) (2016) 4223–4234.
- [4] C. Wang, G. Yen, M. Jiang, A grey prediction-based evolutionary algorithm for dynamic multiobjective optimization, *Swarm. Evol. Comput.* 100695 (2020).
- [5] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, D fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.* 6 (2002) 182–197.
- [6] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: Solving problems with box constraints, *IEEE Trans. Evol. Comput.* 18 (4) (2014) 577–601.
- [7] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm, Technical Report 103, Computer Engineering and Networks Laboratory (TIK). Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.
- [8] S. Mirjalili, S. Mirjalili, S. Saremi, I. Aljarah, Grasshopper optimization algorithm for multi-objective optimization problems, *Appl. Intell.* 48 (2018) 805–820.
- [9] S. Mirjalili, P. Jangir, S. Mirjalili, S. Saremi, I. Trivedi, Optimization of problems with multiple objectives using the multi-verse optimization algorithm, *Knowl.-based Syst.* 134 (2017) 50–71.
- [10] A. Auger, J. Bader, D. Brockhoff, E. Zitzler, Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications, *Theor. Comput. Sci.* 425 (1) (2012) 75–103.
- [11] D. Phan, J. Suzuki, R2-IBEA: R2 indicator based evolutionary algorithm for multiobjective optimization, *IEEE Congr. Evol. Comput.* (2013) 1836–1845.
- [12] K. Bringmann, T. Friedrich, F. Neumann, M. Wagner, Approximation guided evolutionary multiobjective optimization, *Proc. 21st Int. Joint Conf. Artif. Intell.* (2011) 1198–1203.
- [13] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evol. Comput.* 11 (6) (2007) 712–731.
- [14] D. Cai, W. Yuping, Y. Miao, a new multi-objective particle swarm optimization algorithm based on decomposition, *Inf. Sci.* 325 (2015) 541–557.
- [15] S. Mirjalili, P. Jangir, S. Saremi, Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems, *Appl. Intell.* 1–16 (2016).
- [16] Z. Ma, Y. Wang, Shift-based penalty for evolutionary constrained multiobjective optimization and its application, *IEEE Trans. Cybern.* (2022) (in press).
- [17] W. Han, H. Li, B. Xin, M. Gong, Automatic binary and ternary change detection in SAR images based on evolutionary multiobjective optimization, *Appl. Soft Comput.* 125 (2022) 109200.
- [18] A. Vodopija, J. Stork, T. Bart-Beielstein, B. Filipic, Elevator group control as a constrained multiobjective optimization problem, *Appl. Soft Comput.* 115 (2022) 108277.
- [19] C. Li, S. Yang, A general framework of multipopulation methods with clustering in indetectable dynamic environments, *IEEE Trans. Evol. Comput.* 16 (4) (2012) 556–577.
- [20] R. Chang, H. Hsu, S. Lin, J. Ho, Query-based learning for dynamic particle swarm optimization, *IEEE Access.* 5 (2017) 7648–7658.
- [21] Y. Ye, Q. Lin, et al., Multiple source transfer learning for dynamic multiobjective optimization, *Inform. Sci.* 607 (2022) 739–757.
- [22] J. Grefenstette, Genetic algorithm for changing environments, *Parallel Problem Solving from Nature* 2 (1992) 137–144.
- [23] S. Yang, Genetic algorithms with memory- and elitism-based immigrants in dynamic environments, *Evol. Comput.* 16 (3) (2008) 385–416.
- [24] K. Deb, S. Karthik, Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling, in: *Lecture Notes in Computer Science*, Springer Science mathplus Business Media, 2007, pp. 803–817.
- [25] J. Branke, Memory enhanced evolutionary algorithms for changing optimization problems, in: *Proceedings of the 1999 congress on evolutionary computation*. Institute of Electrical and Electronics Engineers, 1999.
- [26] H. Xie, Z. Juan, S. Yang, et al., A decision variable classification-based cooperative coevolutionary algorithm for dynamic multiobjective optimization, *Inf. Sci.* 560 (2021) 307–330.
- [27] R. Azzouz, S. Bechikh, L. Said, A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy, *Soft Comput.* 21 (4) (2015) 1–22.
- [28] B. Xu, Y. Zhang, D. Gong, Y. Guo, M. Rong, Environment sensitivity-based cooperative co-evolutionary algorithms for dynamic multi-objective optimization, *IEEE ACM Trans. Comput. Biol.* 15 (6) (2017) 1877–1890.
- [29] R. Liu, J. Li, C. Mu, L. Jiao, A coevolutionary technique based on multi-swarm particle swarm optimization for dynamic multi-objective optimization, *Eur. J. Oper. Res.* 261 (3) (2017) 1028–1051.
- [30] X. Liu, Y. Zhou, X. Yu, Cooperative particle swarm optimization with reference-point-based prediction strategy for dynamic multiobjective optimization, *Appl. Soft Comput.* 87 (2020) 105988.
- [31] A. Zhou, Y. Jin, Q. Zhang, A population prediction strategy for evolutionary dynamic multiobjective optimization, *IEEE Trans. Cybern.* 44 (1) (2014) 66–77.
- [32] A. Muruganantham, K. Tan, P. Vadakkepat, Evolutionary dynamic multiobjective optimization via kalman filter prediction, *IEEE Trans. Evol. Comput.* 46 (12) (2016) 2862–2873.
- [33] Y. Hu, J. Ou, J. Zheng, et al., Solving dynamic multiobjective problems with an evolutionary multi-directional search approach, *Knowl.-based Syst.* 194 (2020) 1–15.
- [34] A. Ahrari, S. Elsayed, R. Sarker, et al., Weighted pointwise prediction method for dynamic multiobjective optimization, *Inf. Sci.* 546 (2021) 349–367.
- [35] Q. Li, J. Zhou, S. Yang, J. Zheng, G. Ruan, A predictive strategy based on special points for evolutionary dynamic multiobjective optimization, *Soft Comput.* 23 (2019) 3723–3739.
- [36] Z. Zhang, S. Qian, Artificial immune system in dynamic environments solving time-varying non-linear constrained multi-objective problems, *Soft Comput.* 15 (7) (2011) 1333–1349.
- [37] J. Eaton, S. Yang, M. Gongora, Ant colony optimization for simulated dynamic multi-objective railway junction rescheduling, *IEEE Trans. Intell. Transp.* 18 (11) (2017) 2980–2992.
- [38] W. Kong, T. Chai, S. Yang, J. Ding, A hybrid evolutionary multiobjective optimization strategy for the dynamic power supply problem in magnesia grain manufacturing, *Appl. Soft Comput.* 13 (5) (2013) 2960–2969.
- [39] 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) (2019) 142–156.
- [40] M. Hajighaei-Kesheli, M. Ammimayeri, Solving the integrated scheduling of production and railtransportation problem by Keshtel algorithm, *Appl. Soft comput.* 25 (2014) 184–203.

- [41] A. Yazici, G. Kirlik, O. Parlaktuna, A. Sipahioglu, A dynamic path planning approach for multirobot sensor-based coverage considering energy constraints, *IEEE Trans. Cybern.* 44 (3) (2014) 305–314.
- [42] H. Ma, Z. Yang, P. You, M. Fei, Multi-objective biogeography-based optimization for dynamic economic emission load dispatch considering plug-in electric vehicles charging, *Energy* 135 (2017) 101–111.
- [43] Q. Zhang, S. Yang, R. Wang, et al, Novel Prediction Strategies for Dynamic Multiobjective optimization, *IEEE Trans. Evol. Comput.* 24 (2) (2020) 260–274.
- [44] O. Theobald, Machine Learning for absolute beginners: A plain English Introduction, third ed., Scatterplot Press, 2021.
- [45] S. Jiang, S. Yang, X. Yao, K. Tan, M. Kaiser, N. Krasnogor, Benchmark problems for CEC2018 Competition on Dynamic multiobjective optimization, 2018 IEEE Congress on Evolutionary Computation, Competition on Dynamic Multiobjective Optimisation (2018).
- [46] W. Sohna, M. Jeong, K. Jeonga, Theoretical comparative study of t-tests and nonparametric tests for final status surveys of MARSSIM at decommissioning sites, *Ann. Nucl. Energy* 135 (2020) 106945.
- [47] L. Cao, L. Xu, E. Goodman, H. Li, A First-Order Difference Model-Based Evolutionary Dynamic Multiobjective Optimization, Asia-Pacific Conference on Simulated Evolution and Learning, Simulated Evolution and Learning (2017) 644–655.
- [48] M. Jiang, Z. Huang, L. Qiu, W. Huang, Transfer learning-based dynamic multiobjective optimization algorithms, *IEEE Trans. Evol. Comput.* 22 (4) (2017) 501–514.
- [49] Q. Zhang, A. Zhou, S. Zhao, P. Suganthan, W. Liu, S. Tiwari, Multiobjective optimization Test Instances for the CEC, Special Session and Competition, IEEE Congress on Evolutionary Computation (CEC) 2009 (2009).