A prediction strategy based on center points and knee points for evolutionary dynamic multi-objective optimization

Juan Zou[a], Qingya Li[a,*], Shengxiang Yang[a,c], Hui Bai[a], Jinhua Zheng[a,b]

[a]*Key Laboratory of Intelligent Computing and Information Processing, Ministry of Education, College of Information Engineering, Xiangtan University, Xiangtan 411105, China*
[b]*School of Computer Science and Technology, Hengyang Normal University, Hunan, China*
[c]*Center for Computational Intelligence (CCI), School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, UK*

Abstract

In real life, there are many dynamic multi-objective optimization problems which vary over time, requiring an optimization algorithm to track the movement of the Pareto front (Pareto set) with time. In this paper, we propose a novel prediction strategy based on center points and knee points (CKPS) consisting of three mechanisms. First, a method of predicting the non-dominated set based on the forward-looking center points is proposed. Second, the knee point set is introduced to the predicted population to predict accurately the location and distribution of the Pareto front after an environmental change. Finally, an adaptive diversity maintenance strategy is proposed, which can generate some random individuals of the corresponding number according to the degree of difficulty of the problem to maintain the diversity of the population. The proposed strategy is compared with four other state-of-the-art strategies. The experimental results show that CKPS is effective for evolutionary dynamic multi-objective optimization.

Keywords: Evolutionary dynamic multi-objective optimization, Prediction, Center point, Knee point, Adaptive diversity maintenance mechanism

1. Introduction

In the real world, there exists a large class of optimization problems which have multiple objectives and those objectives conflict with on another. These kinds of problems are called multi-objective optimization problems(MOPs)[1]. However, real-world problems often contain many uncertain and dynamic factors. For example, air traffic scheduling is usually affected by unexpected events such as bad weather or emergencies. Therefore, it is unlikely that any solutions found for these problems would keep valid for a long time[2]. If objective functions, constraints, or parameters of MOPs change over time, these problems become dynamic MOPs(DMOPs). Due to the presence of dynamics, DMOPs are more difficult to solve than static MOPs. Because evolutionary algorithms perform well when dealing with DMOPs in a new way, they have been widely applied in many areas including scheduling [2, 3, 4], control [5], wireless network design [6], machine learning [7, 8], and constrained optimization [9]. For example, in [2], a method of centroid-based adaptation(CBA) was used to solve the problem of mission planning. Mission planning is one kind of the problems of resource-constrained project scheduling(RCPS). There are two objectives in this problem: the execution time of the plan, and the cost of operating capabilities. Also, several time-varying constraints are proposed, including executing time, the failure of capabilities, and change of task-relationship network. The obtained results showed that CBA was effective for mission planning. The mathematical form of DMOPs is as follows:

$$\begin{cases} \min_{x\in\Omega} F(x,t) = (f_1(x,t), f_2(x,t), \ldots, f_m(x,t))^T \\ s.t.\ \ g_i(x,t) \le 0, i = 1, 2, \ldots, p; \\ \quad\quad h_j(x,t) = 0, j = 1, 2, \ldots, q \end{cases}$$

where $x = (x_1, x_2, \ldots, x_n)$ is the n-dimensional decision vector whose domain of definition is Ω. t represents time variable. $F = (f_1, f_2, \ldots, f_m)$ is the m-dimensional objective vector. g represents p-dimensional inequality constraints, and h is q-dimensional equality constraints. The optimal tradeoff solution set is called the Pareto set (PS) in the decision space and Pareto front (PF) in the objective space.

When dealing with DMOPs, traditional static optimization methods are not applicable [10, 11]. While using these methods, the population gradually converges to the current environment. At the same time, the gene of the population becomes single, and leads the population to lose diversity. Finding the optimal solution as the environment changes is the challenge of static algorithms in the dynamic environment [12, 13, 14, 15]. Thus researchers need to adjust traditional optimization algorithms to solve DMOPs [4, 16]. The easiest way to solve the problem is to increase the diversity of the population. Therefore, many methods have been proposed, such as hyper mutation methods [4], random immigrants and other immigrants' methods [17, 18], employing multiple populations and parallel computing [19, 20], and applying different crossover and mutation operators [21]. However, there is some blindness only increasing the diversity. The ultimate convergence of the popu-

[*] Corresponding author
Email address: smileliqingya@gmail.com (Qingya Li)

lation needs to be able to independently evolve, so the increase of diversity is not a fundamental method to solve the problem.

Thus, some researchers have introduced other methods, such as the random initialization method [22, 23, 24, 4], the memory strategy [25, 26, 27, 28, 29, 4, 30, 31] and prediction strategy [32, 33, 34, 35, 36, 37, 38, 39, 40]. The memory strategy responds to the environment by memorizing previous historical optimal solutions so that it can achieve the purpose of fast convergence. However the memory method is also insufficient. For periodic problems, it can achieve good results. However, the memory method has a poor effect for non-periodic problems. The prediction strategy has good results for a variety of problems. This strategy predicts the optimal solution after an environmental change to respond quickly to the environmental changes by using some historical information or other means. In 2006, Hatzakis and Wallace proposed a feed-forward prediction strategy (FPS) [36] which uses historical boundary point information to predict the boundary points in the new environment, and achieved certain results. However, it is difficult to track the PF just by the boundary points. In 2013, Zhou and Jin et al. proposed a population prediction strategy (PPS) [37] which is very effective when dealing with DMOPs. However, the results are poor due to lack of experience in the early learning stage. In 2015, Wu and Jin et al. proposed a directed search strategy [38], which predictes the population's optimal solution directly using the method of feed-forward center points. This method is effective. However, the whole population predicted by the feed-forward center points, in addition to the non-dominated individuals, does include some useless individuals.

In view of these shortcomings, this paper presents a prediction strategy based on center points and knee points. Its main contributions are as follows:

(1) The non-dominated set can be directly predicted by using the feed-forward center points, and some useless individuals, which are introduced by using the feed-forward center method to predict the total population directly, are eliminated.

(2) The knee point set, which is introduced to the population, can track the PF well, and make the population converge quickly.

(3) An adaptive population diversity maintenance mechanism is proposed, which can increase population diversity by introducing random individuals of corresponding number according to the degree of difficulty of the problem.

The rest of paper is organized as follows. Section 2 presents some related work. Section 3 describes CKPS in detail. Section 4 introduces the test problems and evaluation metric. Section 5 gives the experimental results and analysis, and Section 6 gives more discussion. Section 7 outlines the conclusions and future work.

2. Related work

2.1. A general framework of dynamic multi-objective evolutionary algorithm

A general dynamic framework is as follows:

(1) Initialize the population and set the relevant parameters.

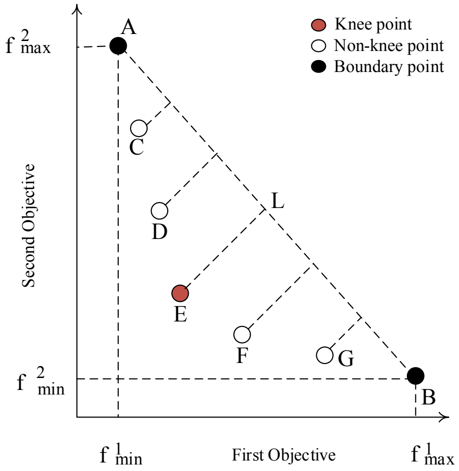


Figure 1: Knee point schematic.

(2) Detect the environmental change. If an environmental change is found, go to step 4; if no change, then go to step 3.

(3) Optimize the problem by evolutionary algorithm.

(4) Take a certain response strategy such as re-initialization, memory, prediction, etc., to respond to the environmental change.

(5) Judge the termination conditions. If not terminated, then go to the second step, if terminated, then exit.

2.2. Knee point

Many researchers have tried to find the local points or regions with special significance on the PF [41, 42, 43, 44, 45]. The knee point or knee region has been of particular interest to researchers.

The knee point on the Pareto optimal front refers to the point with the maximum marginal rates of return. It means that there is a small improvement in one objective, accompanied by severe degradation of at least one other objective. In addition, Zhang [46] proved that the knee point is better than the other points on the PF in the metric of HV [47]. The higher the HV, the better the convergence and distribution of the population.

There are many definitions for the knee point [48, 49, 50, 46]. In this paper, we use the definition of Das [49] and Zhang [46]. As shown in Figure 1, A, B, C, D, E, F and G are seven non-dominated points, where A and B are the boundary points in the non-dominated set. The perpendicular of C, D, E, F, G to the line L formed by A, B represents the distance, shown in formula (2). E has the longest distance from the straight line L, so E is the knee point.

The mathematical formula of straight line L is as follows:

$$ax + by + c = 0. \tag{1}$$

The point is K, and the coordinate is (x_k, y_k), so the distance of the point K to the line L is as follows:

$$d(K, L) = \frac{|ax_K + by_K + c|}{\sqrt{a^2 + b^2}}. \tag{2}$$

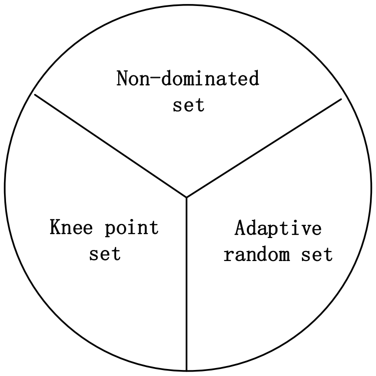


Figure 2: Population composition structure diagram.

For the minimum problem, only the convex knee region is of special interest [49]. So the formula (2) can be changed as follows:

$$d(K,L) = \begin{cases} \frac{|ax_K+by_K+c|}{\sqrt{a^2+b^2}} & if \quad ax_k + by_k + c < 0 \\ -\frac{|ax_K+by_K+c|}{\sqrt{a^2+b^2}} & others \end{cases} \tag{3}$$

From formula (3), we can imagine that when the objective number of the optimization problem is greater than two, the line consisting of the boundary points becomes a plane or a hyperplane, and it becomes the distance of the point to the plane or hyperplane.

3. Predictive strategy based on center points and knee points.

In a dynamic environment, some methods are usually used to predict the population after an environmental change. In this paper, the population is composed of three parts: the first part is a knee point set predicted by historical information; the second part is the non-dominated set predicted by the feed-forward center points, and the third part is the random individuals generated by the adaptive population diversity maintenance strategy, which is called the adaptive random set. The structure is shown in Figure 2.

3.1. Knee point set

Tracking PF or PS has been a hot topic in the dynamic environment. Many dynamic multi-objective optimization problems have been proved to have some regularity among the previous work [37, 51, 52].

We can use historical information to predict the PF after an environmental change. There have been a number of studies on which information to use to predict the PF. The feed-forward prediction strategy (FPS) [36] proposed by Hatzakis and Wallace uses the Pareto boundary point information to track PF. The predictive multiobjective genetic algorithm(PMGA) proposed by Wu et al. [39] tracked the PF by using the center points generated by clustering the PF. In this paper, we propose a method to track PF with knee point set so as to achieve fast convergence.

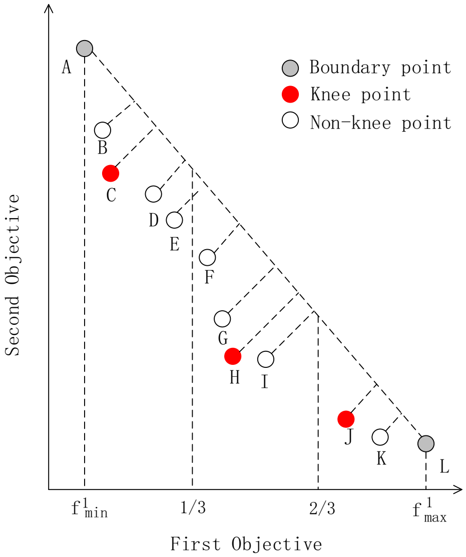


Figure 3: The situation of two objectives and three knee points.

Algorithm 1 Seek the knee point of each partition

Step 1: Layer the population, and then select the individuals in the first layer as the non-dominated set.
Step 2: Seek the boundary individuals in non-dominated set, and the line or plane composed by boundary individuals.
Step 3: Calculate the distance of the point to the line (two objectives) or the plane (three objectives) according to formula (3).
Step 4: According to the number of knee points, num, evenly divide the range of the first objective into num area and then partition the non-dominated set by the value of the first objective of the non-dominated individuals to which partition. Then select the point whose distance is the largest to the straight line or plane in each partition as the knee point. (If a partition is empty, randomly initialize a value.)

Predicting the optimal solution after an environmental change usually requires a great deal of historical information to accumulate experience in a dynamic environment. There is too much information (such as the center point, the boundary points, etc.) on the PF, so it can not be fully predicted. Therefore, if we want to predict the PF, we generally need to select some representative points.

In this paper, we choose the knee point set on the PF to represent the whole PF. Set the number of knee points to be num. The PF is evenly divided into num parts according to the first objective, and then a point, whose distance from the straight line (2 objectives) or plane (3 objectives) composed by connecting the boundary points is the largest, is selected as the knee point of the partition. As shown in Fig. 3, C, H and J are the knee points of each partition. The detailed steps are shown as Algorithm 1.

We select the knee point set as historical information in every environmental change. The autoregressive(AR) [37] prediction model is used to predict the knee point set of the next environment. And then the PF is tracked to achieve the purpose of fast

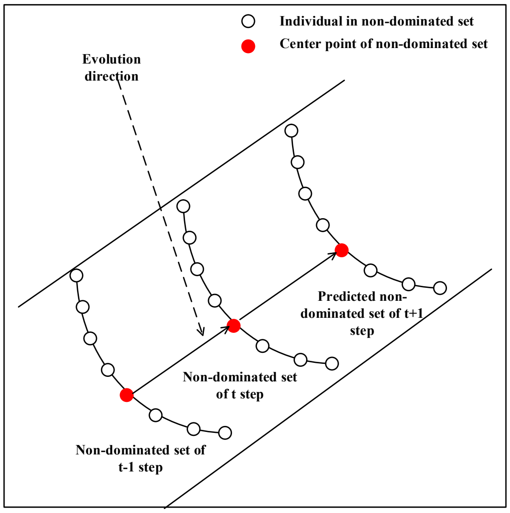

Individual in non-dominated set
Center point of non-dominated set
Evolution direction
Predicted non-dominated set of t+1 step
Non-dominated set of t step
Non-dominated set of t-1 step

Figure 4: Schematic diagram of predicting non-dominated set by feed-forward center points.

convergence with the predicted knee point set.

3.2. Predicting the non-dominated set by feed-forward center points

The method of feed-forward center points [37, 38, 40, 53, 54] has been widely used to predict the population. However, most of these algorithms are used to predict the whole population with feed-forward center points. Generally, the predicted whole population includes three kinds of individuals: (1) the individuals, which play a role of convergence. (2) the individuals, which play a role of diversity. (3) the individuals, which have a small convergent and diverse effect or has the repeat effect with the first two kinds of individuals. Here, The third kind of individuals are called useless individuals. Therefore we only predict the non-dominated individuals using the method of feed-forward center points. The prediction formula is as follows:

$$NonInd_k^{t+1} = C_k^t - C_k^{t-1} + NonInd_k^t + Gaussian(0, d), \quad (4)$$

where $k = 1, 2, \ldots, n$, n represents the dimension of the decision vector. $NonInd_k^{t+1}$ and $NonInd_k^t$ represent the non-dominated individuals in t+1 step and t step in the k-th dimension, respectively. d represents the perturbation of variance. t denotes the time variable. C_k^t and C_k^{t-1} are the center points and in t step and t-1 step in the k-th dimension. C_k^t can be shown as follows:

$$C_k^t = \frac{1}{|P_{Non-dom}^t|} \sum_{NonInd_k^t \in P_{Non-dom}^t} NonInd_k^t \quad (5)$$

where $|P_{Non-dom}^t|$ represents the cardinality of the non-dominated set.

As shown in Figure 4, the hollow circles on the arc lines represent the non-dominated individuals. The red circles represent the centers of the non-dominated sets, and the lines between the center points represent evolutionary directions. The non-dominated set of t+1 time step can be predicted by means of the non-dominated set of the t time step plus the evolution direction obtained by the method of center point t time step minus center point of t-1 time step. Because it is not an accurate prediction, a Gaussian perturbation is added.

```
Algorithm 2 Generate random set adaptively
Input: $N_{pop}$, the size of the population $pop$ ; $N_{non-dom}$, the size of predicted non-dominated set; $N_{kp}$, the size of predicted knee point set $pop_{kp}$.
Output: $pop_{rand}$, adaptive random set and its size,$N_{rand}$.
Step 1: If $N_{kp} + N_{non-dom} > N_{pop}$, calculate the crowing distance of the individuals in the non-dominated set [10], and delete the individuals whose crowing distance is the smallest. Set the size of the non-dominated set after deleted to be $N''_{Non-dom}$, to make it be: $N_{kp} + N''_{Non-dom} = N_{pop}$.
    $Non-pop[i]_{dist} = \sum_{k=1}^{r}(Non-pop[i+1].f_k - Non-pop[i-1].f_k)$ (6)
  where r represents that the individual has r objectives; $Non-pop[i+1].f_k$ denotes the k-th objective of the (i + 1)-th individual in the non-dominated set. At this time, $N_{rand}$ is 0;
Step 2: If $N_{kp}+N_{Non-dom} \leq Npop$, calculate the size of the random set using formula (7).
    $N_{rand} = N_{pop} - N_{Non-dom} - N_{kp}$ (7)
  Then generate random set $pop_{rand}$ by formula (8).
    $x_k = random(low_k, up_k),$ (8)
  where $k = 1, 2, \ldots, n$. n is the dimensions of the decision space; $x_k$ represents the random individual in the k-th dimension. $low_k$ denotes the lower bound of the k-th dimension of the decision space; $up_k$ denotes the upper bound in the k-th dimension in the decision space. Formula (8) denotes that $x_k$ is generated randomly between $low_k$ and $up_k$.
Step 3: return $pop_{rand}$ and $N_{rand}$.
```

3.3. Adaptive diversity maintenance strategy

Maintaining diversity is very important in a dynamic environment. When dealing with optimization problems with static algorithms, the population tends to converge to the current environment and loses diversity, which makes it difficult to find the optimal set when the environment changes. A diversity maintenance strategy based on center points and knee points is proposed to introduce random individuals of different numbers according to the degree of difficulty of the problem.

The more difficult the problem is, the more difficult it is for dynamic multi-objective optimization algorithm to converge toward PF or PS. Under the same optimization generation number, the size of the non-dominated set of the first layer is usually less. On the contrary, the easier the problem is, the easier it will tend to converge toward PF or PS. Under the same optimization generation number, the size of non-dominated set is usually more. Based on the above principle, random individuals of the adaptive number are introduced to keep diversity.

The detailed steps are shown in Algorithm 2. Normally, N_{pop}, which is the size of the population pop, is fixed. And

Algorithm 3 Prediction strategy based on center points and knee points

```
Initialization: number of time change t: = 0; generation counter $gt$ := 0; total evolution generation number, $gmax$.
Step 1: Detect the environmental change. If no change, go to Step 7; if change, go to Step 2.
Step 2: Get knee point set using Algorithm 1, $pop_{kp}$. And its size, $N_{kp}$.
Step 3: Calculate the center point in each dimension, $C_k^t$, using formula (5).
Step 4: Predict the non-dominated set after an environmental change using formula (4), set to be $pop_{Non-dom}$. The size is $N_{Non-dom}$.
Step 5: Calculate the adaptive random set, $pop_{rand}$ and its size, $N_{rand}$ using Algorithm 2.
Step 6: Get the new population $pop_{t+1} = pop_{kp} + pop_{Non-dom} + pop_{rand}$
Step 7: Optimize the population with optimization algorithm RM-MEAD [55].
Step 8: If $gt > gmax$, output $pop_t$, and end; or $gt := gt + 1$, go to Step 1.
```

N_{kp}, which is the size of predicted knee point set, is set by parameter, which is also fixed. Nnon-dom, is the size of predicted non-dominated set; So the value of N_{rand} is decided by $N_{Non-dom}$. According to formula (7), the greater the $N_{Non-dom}$, the smaller the N_{rand}. Similarly, the smaller the $N_{non-dom}$, the greater the N_{rand}.

In general, when the multi-objective optimization problem is particularly easy, the optimal solution will be very easy to predict. Meanwhile, the random individuals used to keep diversity are not so important and more random individuals will affect the convergence rate of the population. In this case, N_{rand} in this strategy is relatively small, or even likely to be 0. When the optimization problem is difficult, the predicted non-dominated set is no longer accurate. In this case, a number of random individuals are needed to keep the diversity, and then accelerate the convergence of the population. Consistently, the value of N_{rand} in this paper is also large, just to maintain the diversity of the population.

3.4. Detailed description of CKPS

CKPS is under the framework of the dynamic multi-objective evolutionary algorithm. After an environmental change, CKPS initializes the population so that the population can respond quickly to environmental changes. A detailed description of CKPS is shown in Algorithm 3.

4. Test problems and performance indicators

4.1. Test problems

In this paper, several commonly used dynamic multi-objective test problems are used, such as the FDA test suite [51], dMOP test suite [52] and F5-F10 [37]. Both the FDA and dMOP test suits are linearly related between decision variables. F5-F10 are the test problems whose decision variables are nonlinearly related. FDA4 and F8 are the problems with three objectives, and the others are problems with two objectives. F9-F10 are the test problems which are more difficult to converge than the other test problems.

4.2. Performance indicators

In our experimental studies, we adopted the performance metrics which could help deeply investigate the performance of algorithms, such as convergence and distribution.

(1) Modified Inverted Generational Distance(MIGD)

Inverted Generational Distance(IGD)[56, 5] has been widely used to measure both the convergence and distribution. Here, let PF_t be a set of uniformly distributed Pareto optimal points of the PF in t time and let P_t be an approximation set of PF in t time. IGD metric is defined as follows,

$$IGD(PF_t, P_t) = \frac{\sum_{v \in PF_t} d(v, P_t)}{|PF_t|} \tag{9}$$

where $d(v, P_t) = min_{u \in P_t} \|F(v) - F(u)\|$ is the distance between v and P_t. $|PF_t|$ is the cardinality of PF_t.

Since the environment is dynamic, it is hard to say which algorithm is better between two algorithms in some situations if only using simply IGD [32]. MIGD metric [30, 37, 57, 53, 54] is a modified version of IGD, which is defined as the average of the IGD values in some time steps over a run.

$$MIGD = \frac{1}{|T|} \sum_{t \in T} IGD(PF_t, P_t) \tag{10}$$

where T is a set of discrete time points in a run and $|T|$ is the cardinality of T. MIGD is also a comprehensive performance metric like IGD to evaluate the performance of algorithms regarding convergence and distribution. The smaller the value is, the better the performance of the algorithm is.

(2)Modified Hypervolume Difference(MHVD)

The Hypervolume Difference(HVD)[32, 58] measures the gap between the hypervolume of the obtained PF and that of the true PF. let PF_t be a set of uniformly distributed Pareto optimal points of the PF in t time and let P_t be an approximation set of PF in t time.

$$HVD(PF_t, P_t) = HV(PF_t) - HV(P_t) \tag{11}$$

where $HV(S)$ is the hypervolume of a set S.

MHVD is a modified version of HVD like MIGD to IGD. MHVD is defined as the average of the HVD values in some time steps over a run.

$$MHVD = \frac{1}{|T|} \sum_{t \in T} HVD(PF_t, P_t) \tag{12}$$

where T is a set of discrete time points in a run and $|T|$ is the cardinality of T. The reference point for the computation of hypervolume is $(Z_1^t + 0.5, Z_2^t + 0.5, \ldots, Z_M^t + 0.5)$, where Z_j^t is the maximum value of the j-th objective of the true PF in t time; M is the number of objectives. MHVD is also a comprehensive metric like HV to evaluate both the convergence and distribution. The smaller the value is, the better the performance of the algorithm is.

Table 1: Mean and Standard Deviation of MIGD values of five strategies on FDA and dMOP test suites. ‡ and † indicate CKPS performs significantly better than and equivalently to the corresponding algorithm, respectively.

Problems	Statistic	RIS	FPS	PMGA	PPS	CKPS
FDA1	total	1.3155(0.03030)‡	0.0516(0.00864)‡	0.0581(0.00316)‡	0.0528(0.00913)‡	**0.0276(0.00360)**
	1st stage	1.2215(0.07518)‡	0.2090(0.04391)‡	**0.0634(0.00658)**	0.2406(0.04615)‡	0.1051(0.01887)
	2nd stage	1.3310(0.04145)‡	0.0151(0.00117)‡	0.0567(0.00606)‡	0.0102(0.00103)‡	**0.0092(0.00015)**
	3rd stage	1.3447(0.04696)‡	0.0134(0.00084)‡	0.0581(0.00316)‡	**0.0062(0.00008)**	0.0091(0.00012)
FDA2	total	0.0500(0.00078)‡	**0.0085(0.00068)**	0.4435(0.01275)‡	0.0097(0.00075)‡	0.0086(0.00053)
	1st stage	0.0491(0.00115)‡	0.0198(0.00332)†	0.5137(0.04302)‡	0.0232(0.00317)‡	**0.0185(0.00260)**
	2nd stage	0.0503(0.00123)‡	**0.0060(0.00034)**	0.4269(0.01171)‡	0.0070(0.00068)‡	0.0066(0.00026)
	3rd stage	0.0501(0.00130)‡	**0.0056(0.00003)**	0.4435(0.01275)‡	0.0060(0.00004)	0.0060(0.00006)
FDA3	total	1.7564(0.06554)‡	0.0645(0.00927)‡	0.7617(0.00886)‡	0.0941(0.01582)‡	**0.0408(0.00790)**
	1st stage	1.5737(0.12477)‡	0.2084(0.04439)‡	0.7552(0.01274)‡	0.3209(0.07801)‡	**0.1374(0.04224)**
	2nd stage	1.7709(0.11832)‡	0.0305(0.00339)‡	0.7610(0.01097)‡	0.0420(0.00821)‡	**0.0176(0.00217)**
	3rd stage	1.8287(0.11044)‡	0.0303(0.00421)‡	0.7617(0.00886)‡	0.0384(0.00675)‡	**0.0180(0.00163)**
FDA4	total	0.4566(0.00922)‡	0.1414(0.00337)‡	3.7558(0.00341)‡	0.1307(0.00205)‡	**0.1204(0.00165)**
	1st stage	0.4390(0.02098)‡	0.1629(0.00852)‡	0.1583(0.00239)‡	0.1660(0.00816)‡	**0.1385(0.00561)**
	2nd stage	0.4594(0.01512)‡	0.1376(0.00427)‡	4.6121(0.01405)‡	0.1247(0.00278)‡	**0.1159(0.00203)**
	3rd stage	0.4622(0.01131)‡	0.1386(0.00340)‡	3.7558(0.00341)‡	0.1231(0.00283)‡	**0.1163(0.00161)**
dMOP1	total	0.6386(0.01427)‡	**0.0072(0.00115)**	0.2437(0.01891)‡	0.0379(0.05152)‡	0.0076(0.00141)
	1st stage	0.6486(0.03460)‡	0.0195(0.00610)†	0.2037(0.02545)‡	0.1413(0.21119)‡	**0.0183(0.00727)**
	2nd stage	0.6383(0.02790)‡	**0.0043(0.00008)**	0.2724(0.02718)‡	0.0209(0.02865)†	0.0051(0.00005)
	3rd stage	0.6341(0.01865)‡	**0.0043(0.00007)**	0.2437(0.01891)‡	0.0057(0.00002)‡	0.0051(0.00006)
dMOP2	total	1.6968(0.05407)‡	0.0622(0.00788)‡	0.1451(0.00650)‡	0.0607(0.01023)‡	**0.0272(0.00611)**
	1st stage	1.6332(0.08749)‡	0.2552(0.03931)‡	0.1774(0.01274)‡	0.2799(0.05178)‡	**0.1028(0.03196)**
	2nd stage	1.7087(0.09220)‡	0.0170(0.00113)‡	0.1384(0.00727)‡	0.0111(0.00103)‡	**0.0094(0.00023)**
	3rd stage	1.7150(0.06717)‡	0.0159(0.00080)‡	0.1451(0.00650)‡	**0.0061(0.00006)**	0.0091(0.00023)
dMOP3	total	1.3215(0.03752)‡	0.0523(0.00654)‡	0.0581(0.00316)‡	0.0527(0.01084)‡	**0.0271(0.00480)**
	1st stage	1.2558(0.08243)‡	0.2124(0.03335)‡	**0.0634(0.00658)**	0.2403(0.05530)‡	0.1030(0.02493)
	2nd stage	1.3319(0.05383)‡	0.0149(0.00077)‡	0.0567(0.00606)‡	0.0101(0.00093)‡	**0.0091(0.00014)**
	3rd stage	1.3423(0.06014)‡	0.0136(0.00076)‡	0.0581(0.00316)‡	**0.0062(0.00010)**	0.0090(0.00011)

5. Experimental results and analysis

5.1. Parameter settings

The strategies to compare with CKPS are as follows: (1) randomly initialize strategy (RIS), which randomly generates individuals in the feasible region of the decision space. (2) feedforward prediction strategy(FPS) [36] (3) predictive multiobjective genetic algorithm(PMGA) [39], which tracks the PF by using the center points generated by clustering the PF. (4) population prediction strategy(PPS) [37]. The optimization algorithm uses RM-MEDA [55] to optimize problems in RIS, FPS and PPS. Because NSGA-II was used to optimize problems in original PMGA, the optimization algorithm in PMGA is still NSGA-II here. Run 20 times independently on each problem and there are 100 environmental changes in every run. Environmental change degree nt is 10 and the size of the population is 100. The dimensions of the decision space is 20. The parameter p is 3 and the length of the history information sequence M is 23 in AR(p) model. These parameters are consistent with [37], [53].

Parameters in FPS: There are 3(m+1) predicted individuals. Seventy percent of the other individuals in the new population inherit the population in the last environmental change, and the other 30% are generated randomly in search space[37],[53].

Parameters in PMGA: The number of centroid centers is 5 [39].

Parameters in CKPS: The number of knee points is 9.

Environmental change detection: Select 5% of the individuals of the population to detect the environmental change. A mismatch in the objective values suggests the environment has changed[37],[53].

5.2. Comparison of performance evaluation results

In dynamic environments, the performance of different strategies in different periods needs to be discussed in some situations [32, 37, 57, 53]. So, we divided 100 environmental changes into three stages in the experiment: the first stage (1st stage) included the front 20 environmental changes; the second stage (2nd stage) included the middle 40 environmental changes; the third stage (3rd stage) included the last 40 environmental changes. Mean and standard deviation of MIGD values and MHVD values are shown in Tables 1, 2, 3 and 4, respectively. The best values are in bold face by comparison of the five algorithms. The Wilcoxon ranksum test [59] was carried out to indicate significance between different results at the 0.05 significance level.

5.2.1. Experimental results of comparison experiment on FDA and dMOP test suits

As shown in Table 1, the five strategies are compared on the FDA test suite and the dMOP test suite whose decision variables are linearly related for MIGD metric. The values in each small cell denote average value and standard deviation. Total represents all environmental changes, which includes 100 environmental changes.

(1) For the total stage, CKPS is slightly worse than FPS on FDA2 and dMOP2 whose PS are changeless. But for all the other problems, CKPS is better than the four other strategies. It indicates that CKPS has the ability to quickly respond to environmental changes. The results of RIS are the worst in most cases, which indicates that these prediction strategies are better than random strategy.

(2) For the first stage, CKPS is slightly worse than PMGA on FDA1 and dMOP3. For all the other problems, CKPS is significantly better than RIS, FPS, PMGA and PPS.

Table 2: Mean and Standard Deviation of MHVD values of five strategies on FDA and dMOP test suites. ‡ and † indicate CKPS performs significantly better than and equivalently to the corresponding algorithm, respectively.

Problems	Statistic	RIS	FPS	PMGA	PPS	CKPS
FDA1	total	1.2328(0.01073)‡	0.0968(0.01053)‡	0.1326(0.00669)‡	0.0948(0.01448)‡	**0.0461(0.00287)**
	1st stage	1.2145(0.01900)‡	0.3680(0.05223)‡	**0.1424(0.01292)**	0.4192(0.07139)‡	0.1525(0.01524)
	2nd stage	1.2353(0.01564)‡	0.0342(0.00249)‡	0.1302(0.00557)‡	0.0222(0.00227)†	**0.0209(0.00046)**
	3rd stage	1.2390(0.01275)‡	0.0305(0.00222)‡	0.1301(0.01206)‡	**0.0133(0.00022)**	0.0207(0.00031)
FDA2	total	0.0714(0.00125)‡	**0.0320(0.00079)**	0.7283(0.01669)‡	0.0325(0.00072)†	**0.0320(0.00067)**
	1st stage	0.0719(0.00221)‡	0.0372(0.00413)†	0.9557(0.05030)‡	0.0405(0.00408)	**0.0365(0.00364)**
	2nd stage	0.0714(0.00209)‡	0.0306(0.00026)‡	0.6743(0.00871)‡	**0.0301(0.00029)**	0.0308(0.00016)
	3rd stage	0.0711(0.00188)‡	**0.0310(0.00004)**	0.6742(0.00871)‡	**0.0310(0.00005)**	0.0312(0.00005)
FDA3	total	1.9361(0.01767)‡	0.7761(0.01825)‡	0.8604(0.00959)‡	0.8420(0.02751)‡	**0.6668(0.00662)**
	1st stage	1.9195(0.04820)‡	1.0479(0.05214)‡	0.9852(0.01236)‡	1.2829(0.14896)‡	**0.7489(0.02380)**
	2nd stage	1.9387(0.02533)‡	0.7167(0.01667)‡	0.8310(0.01652)‡	0.7436(0.01854)‡	**0.6450(0.00915)**
	3rd stage	1.9414(0.02960)‡	0.7064(0.01855)‡	0.8310(0.02220)‡	0.7310(0.02692)‡	**0.6497(0.01029)**
FDA4	total	1.3838(0.01837)‡	0.4249(0.01308)‡	1.3108(0.01204)‡	0.3818(0.00756)‡	**0.3363(0.00552)**
	1st stage	1.3392(0.03975)‡	0.4993(0.03182)‡	0.4503(0.00852)‡	0.5100(0.02893)‡	**0.4034(0.02098)**
	2nd stage	1.3942(0.02764)‡	0.4051(0.01404)‡	1.5138(0.01953)‡	0.3541(0.01032)‡	**0.3198(0.00664)**
	3rd stage	1.3945(0.02189)‡	0.4093(0.01419)‡	1.5165(0.01596)‡	0.3487(0.01104)‡	**0.3209(0.00458)**
dMOP1	total	1.1531(0.01823)‡	0.1501(0.00097)‡	0.4267(0.03460)‡	0.1688(0.03108)‡	**0.1491(0.00079)**
	1st stage	1.0885(0.04142)‡	0.1180(0.00484)†	0.3385(0.05409)‡	0.1715(0.09166)†	**0.1165(0.00398)**
	2nd stage	1.1718(0.03312)‡	0.1577(0.00018)‡	0.3985(0.02189)‡	0.1780(0.03702)‡	**0.1569(0.00013)**
	3rd stage	1.1651(0.02816)‡	0.1576(0.00020)‡	0.4968(0.04069)‡	0.1583(0.00015)‡	**0.1567(0.00025)**
dMOP2	total	1.2672(0.01126)‡	0.2156(0.01314)‡	0.2138(0.01806)‡	0.2190(0.01537)‡	**0.1646(0.00325)**
	1st stage	1.1689(0.02085)‡	0.4303(0.06200)‡	0.2783(0.02538)‡	0.4484(0.07664)‡	**0.1862(0.01751)**
	2nd stage	1.2967(0.02139)‡	0.1656(0.00309)‡	0.1933(0.01614)‡	0.1709(0.00237)‡	**0.1596(0.00034)**
	3rd stage	1.2844(0.02055)‡	0.1636(0.00140)‡	0.2036(0.02210)‡	**0.1581(0.00015)**	0.1593(0.00057)
dMOP3	total	1.2282(0.00948)‡	0.0972(0.00848)‡	0.1326(0.00669)‡	0.0947(0.01470)‡	**0.0459(0.00324)**
	1st stage	1.2046(0.02570)‡	0.3695(0.04400)‡	**0.1424(0.01292)**	0.4193(0.07355)‡	0.1530(0.01662)
	2nd stage	1.2317(0.01575)‡	0.0340(0.00172)‡	0.1302(0.00557)‡	0.0219(0.00206)†	**0.0206(0.00033)**
	3rd stage	1.2359(0.02242)‡	0.0312(0.00184)‡	0.1302(0.01206)‡	**0.0133(0.00024)**	0.0203(0.00030)

Table 3: Mean and Standard Deviation of MIGD values of five strategies on F5-F10. ‡ and † indicate CKPS performs significantly better than and equivalently to the corresponding algorithm, respectively.

Problems	Statistic	RIS	FPS	PMGA	PPS	CKPS
F5	total	1.1439(0.04184)‡	0.1852(0.08194)‡	3.1110(0.07348)‡	0.2323(0.07728)‡	**0.0366(0.00964)**
	1st stage	1.1844(0.06775)‡	0.5886(0.41127)‡	2.4771(0.36662)‡	1.0473(0.36248)‡	**0.1163(0.04642)**
	2nd stage	1.1344(0.04050)‡	0.1088(0.02892)‡	3.2192(0.24029)‡	0.0664(0.02698)‡	**0.0176(0.00136)**
	3rd stage	1.1341(0.06016)‡	0.0746(0.02526)‡	3.1110(0.07348)‡	**0.0169(0.00131)**	0.0178(0.00127)
F6	total	0.5399(0.01228)‡	0.0548(0.01680)‡	1.2521(0.01678)‡	0.0751(0.04238)‡	**0.0209(0.00279)**
	1st stage	0.6958(0.03924)‡	0.1291(0.07744)‡	1.2312(0.06075)‡	0.3084(0.20490)‡	**0.0468(0.01387)**
	2nd stage	0.5103(0.01898)‡	0.0404(0.00820)‡	1.2799(0.03064)‡	0.0269(0.00916)‡	**0.0148(0.00035)**
	3rd stage	0.4956(0.01674)‡	0.0352(0.00504)‡	1.2521(0.01678)‡	0.0143(0.00050)‡	**0.0140(0.00037)**
F7	total	0.6165(0.01535)‡	0.1273(0.02343)‡	1.2892(0.08301)‡	0.1006(0.04024)‡	**0.0236(0.00751)**
	1st stage	0.6764(0.03353)‡	0.3499(0.10311)‡	1.3816(0.11334)‡	0.4575(0.20361)‡	**0.0642(0.03934)**
	2nd stage	0.6009(0.01925)‡	0.0879(0.02554)‡	1.2468(0.09617)‡	0.0208(0.00415)‡	**0.0139(0.00032)**
	3rd stage	0.6037(0.02552)‡	0.0642(0.02467)‡	1.2892(0.08301)‡	**0.0133(0.00055)**	0.0140(0.00027)
F8	total	0.9083(0.02482)‡	**0.1418(0.00363)**	0.3795(0.00647)‡	0.1455(0.00463)	0.1542(0.00428)
	1st stage	0.7666(0.04113)‡	**0.1944(0.01645)**	0.3978(0.02375)‡	0.2106(0.02369)†	0.2090(0.02210)
	2nd stage	0.9473(0.04443)‡	**0.1313(0.00293)**	0.3708(0.00756)‡	0.1335(0.00198)	0.1422(0.00423)
	3rd stage	0.9366(0.02643)‡	0.1309(0.00169)	0.3795(0.00647)‡	**0.1301(0.00250)**	0.1401(0.00323)
F9	total	1.1923(0.03253)‡	0.3542(0.06750)‡	2.5320(0.11486)‡	0.6186(0.19477)‡	**0.1128(0.03742)**
	1st stage	1.2308(0.10033)‡	0.9770(0.19262)‡	4.2814(0.09784)‡	2.5294(0.95190)‡	**0.3626(0.16418)**
	2nd stage	1.1787(0.06723)‡	0.2203(0.07491)‡	1.2583(0.00682)‡	0.2632(0.09615)‡	**0.0540(0.02247)**
	3rd stage	1.1875(0.04852)‡	0.2012(0.08490)‡	2.5320(0.11486)‡	0.0819(0.01601)‡	**0.0528(0.02338)**
F10	total	1.0691(0.04720)‡	0.4280(0.05312)‡	0.7665(0.05569)‡	0.5097(0.09978)‡	**0.0834(0.02518)**
	1st stage	1.1510(0.07724)‡	0.6341(0.12031)‡	1.1785(0.25386)‡	1.5974(0.44011)‡	**0.2839(0.12102)**
	2nd stage	1.0608(0.04913)‡	0.4188(0.05751)‡	0.6958(0.01573)‡	0.3004(0.08774)‡	**0.0379(0.00357)**
	3rd stage	1.0385(0.07501)‡	0.3499(0.06662)‡	0.7665(0.05569)‡	0.2150(0.04395)‡	**0.0338(0.00449)**

(3) For the second stage, CKPS is better than the other strategies for most of the problems. But for the third stage, CKPS is worse than PPS on FDA1, dMOP2 and dMOP3.

(4) For FDA4 which is three-dimensional, CKPS is better than RIS, FPS, PMGA and PPS at all stages.

Exploring its reason to know, both FPS and PPS require the accumulation of experience. Although knee points in CKPS also need the accumulation of experience when predicting the knee points, the non-dominated set can be directly predicted not to accumulate experience. Coupled with the adaptive diversity maintenance mechanism of CKPS, CKPS converges faster than FPS and PPS. However, for the third stage, because the experience has been accumulated enough, PPS is better than CKPS for some test problems.

As shown in Table 2, the five strategies are compared on FDA and dMOP test suites for DHVD metric. The comparable results are almost same as MIGD. However, the effect of CKPS seems better in some problems for DHVD. For example, CKPS is better than FPS in all stages on dMOP1 for MHVD metric, but not that for MIGD. The reason may be that the introduction of knee point set makes CKPS seem better on MHVD metric.

Table 4: Mean and Standard Deviation of MHVD values of five strategies on F5-F10. ‡ and † indicate CKPS performs significantly better than and equivalently to the corresponding algorithm, respectively.

Problems	Statistic	RIS	FPS	PMGA	PPS	CKPS
F5	total	1.3849(0.01419)‡	0.4745(0.07067)‡	1.7364(0.01230)‡	0.4845(0.05354)‡	**0.2867(0.01269)**
	1st stage	1.5641(0.03251)‡	0.9525(0.29421)‡	1.8196(0.04988)‡	1.3391(0.21636)‡	**0.4558(0.06450)**
	2nd stage	1.3441(0.01687)‡	0.3951(0.05421)‡	1.7344(0.03510)‡	0.3148(0.04150)‡	**0.2466(0.00213)**
	3rd stage	1.3406(0.02808)‡	0.3269(0.04382)‡	1.6988(0.01811)‡	0.2483(0.00111)‡	**0.2464(0.00160)**
F6	total	0.9995(0.01531)‡	0.3050(0.03246)‡	1.5555(0.00974)‡	0.3436(0.04867)‡	**0.2680(0.00663)**
	1st stage	1.3182(0.04079)‡	0.4923(0.14787)‡	1.7031(0.03430)‡	0.7177(0.22655)‡	**0.3569(0.03433)**
	2nd stage	0.9273(0.02093)‡	0.2646(0.02006)‡	1.5148(0.01269)‡	0.2626(0.01394)‡	**0.2470(0.00086)**
	3rd stage	0.9204(0.02104)‡	0.2565(0.00858)‡	1.5259(0.03027)‡	**0.2468(0.00105)**	0.2469(0.00076)
F7	total	1.0988(0.01522)‡	0.4448(0.04393)‡	1.5220(0.02041)‡	0.3625(0.02877)†	**0.2706(0.01470)**
	1st stage	1.3190(0.03876)‡	0.8952(0.17527)‡	1.7286(0.01664)‡	0.8413(0.14139)‡	**0.3804(0.07631)**
	2nd stage	1.0465(0.02740)‡	0.3599(0.05159)‡	1.4745(0.02450)‡	0.2509(0.00448)‡	**0.2444(0.00081)**
	3rd stage	1.0464(0.02178)‡	0.3157(0.04522)‡	1.4715(0.02634)‡	0.2467(0.00126)‡	**0.2445(0.00069)**
F8	total	2.2789(0.02309)‡	**0.3797(0.00916)**	0.9777(0.02016)‡	0.3954(0.01577)	0.4036(0.01049)
	1st stage	2.1534(0.06297)‡	**0.5431(0.04274)**	1.0026(0.06811)‡	0.6076(0.07844)‡	0.5639(0.04777)
	2nd stage	2.3030(0.02905)‡	**0.3413(0.00900)**	0.9828(0.04316)‡	0.3504(0.00781)	0.3690(0.01400)
	3rd stage	2.3144(0.03879)‡	0.3405(0.00750)	0.9609(0.01666)‡	**0.3397(0.00780)**	0.3622(0.00999)
F9	total	1.3930(0.01395)‡	0.6393(0.08717)‡	1.7018(0.01392)‡	0.6481(0.07278)†	**0.3540(0.02149)**
	1st stage	1.5966(0.03298)‡	1.2334(0.16517)†	1.9478(0.00387)‡	1.6530(0.26199)†	**0.6979(0.09374)**
	2nd stage	1.3371(0.01545)‡	0.5259(0.10356)‡	1.7470(0.02775)‡	0.5175(0.10465)‡	**0.2712(0.01756)**
	3rd stage	1.3523(0.03077)‡	0.4705(0.12764)‡	1.5398(0.00912)‡	0.3014(0.02611)‡	**0.2735(0.02093)**
F10	total	1.3534(0.01879)‡	0.8905(0.07059)‡	1.1394(0.02832)‡	0.7476(0.05592)‡	**0.3603(0.03571)**
	1st stage	1.5587(0.03333)‡	1.2692(0.11566)‡	1.7628(0.10585)‡	1.5369(0.09760)‡	**0.7263(0.16954)**
	2nd stage	1.3022(0.02225)‡	0.8541(0.09269)‡	1.0906(0.03339)‡	0.6028(0.09317)‡	**0.2756(0.00777)**
	3rd stage	1.3071(0.02902)‡	0.7471(0.10528)‡	0.8921(0.00826)‡	0.5175(0.07129)‡	**0.2711(0.00851)**

5.2.2. Experimental results of comparison experiment on F5-F10

As shown in Table 3, CKPS is slightly worse than PPS on three-dimensional problem F8 for the total stage, but better than RIS, FPS, PMGA and PPS for the other problems, which indicates that the average performance of CKPS is better. In the early two stages, CKPS is better than RIS, FPS, PMGA and PPS for most of the problems. But for the third stage, PPS is better than CKPS for some problems. Explanations can also be used as explained in Section 5.2.1. F5-F8 are the problems with nonlinear correlation between decision variables. The prediction of population is more difficult than FDA and dMOP test suits for F5-F8. PPS also needs to accumulate experience, and for the third stage, CKPS is slightly worse than PPS except F6. F9-F10 are two more complicated problems than F5-F8. For F9, environmental changes are slight, but sometimes, the Pareto set can jump from one area to another. For F10, the shape of two continual PFs are different. As shown in Table 3, CKPS is better than PPS at all stages of F9-F10. This may be the fact that the adaptive population diversity maintenance mechanism plays an important role except for knee points. The comparable results in Table 4 are almost consistent with Table 3.

As shown in Tables 1, 2, 3 and 4, the standard deviation of CKPS is small. It proves that robustness of CKPS is good. The reason may be that the three mechanisms of CKPS work together to make CKPS have a good robustness. For example, when the problem is easy, the non-dominated set predicted and knee point set predicted play a important role. However, when the problem is difficult, the adaptive diversity mechanism plays a more important role.

5.3. Distribution diagram of final population

For a more intuitive comparison, we selected FDA1, dMOP2, F6 and F10 to draw the final population distribution of the five strategies. It is the test problem of fixed PS in Figure 5, selecting six moments to observe results. Eight moments are selected in the other figures. It can be seen from Figure 5 and Figure 6 that the experimental results are the same as those in Tables 1 and 2. In the early stages, CKPS has better convergence and distribution, which shows that CKPS can better respond to environmental changes. In the later stage, the effect of PPS and CKPS is the best, and their points are very close to PF. From Figure 7 and Figure 8, it can be seen that the experimental results are substantially the same as those in Tables 3 and 4. The problem in Figure 7 is F6, which is a complicated problem. Figure 8 shows a more complex problem, F10. It can be seen that CKPS works very well for complex problems.

6. More discussion

CKPS includes predicting the non-dominated set by using the feed-forward center points and knee point set using the AR model. It has been widely applied to predict the whole population with feed-forward center points. In this section, CKPS and a common method of predicting the entire population using the feed-forward center points are compared. Here, the method of predicting the entire population using the feed-forward center points is called Center-point Prediction Strategy(CPS). In addition, CKPS with different numbers of knee points are compared to further observe the effect of the knee point on CKPS.

6.1. Comparison experiment of CKPS and CPS

6.1.1. A simple form of CPS

A simple form of CPS is as follows:

$$pop_k^{t+1} = (C_k^t - C_k^{t-1}) + pop_k^t + Gaussion(0, d) \tag{13}$$

where $k = 1, 2, \ldots, n$, n represents the dimensions of the decision vector. pop_k^{t+1} and pop_k^k denote the individuals in t+1 step

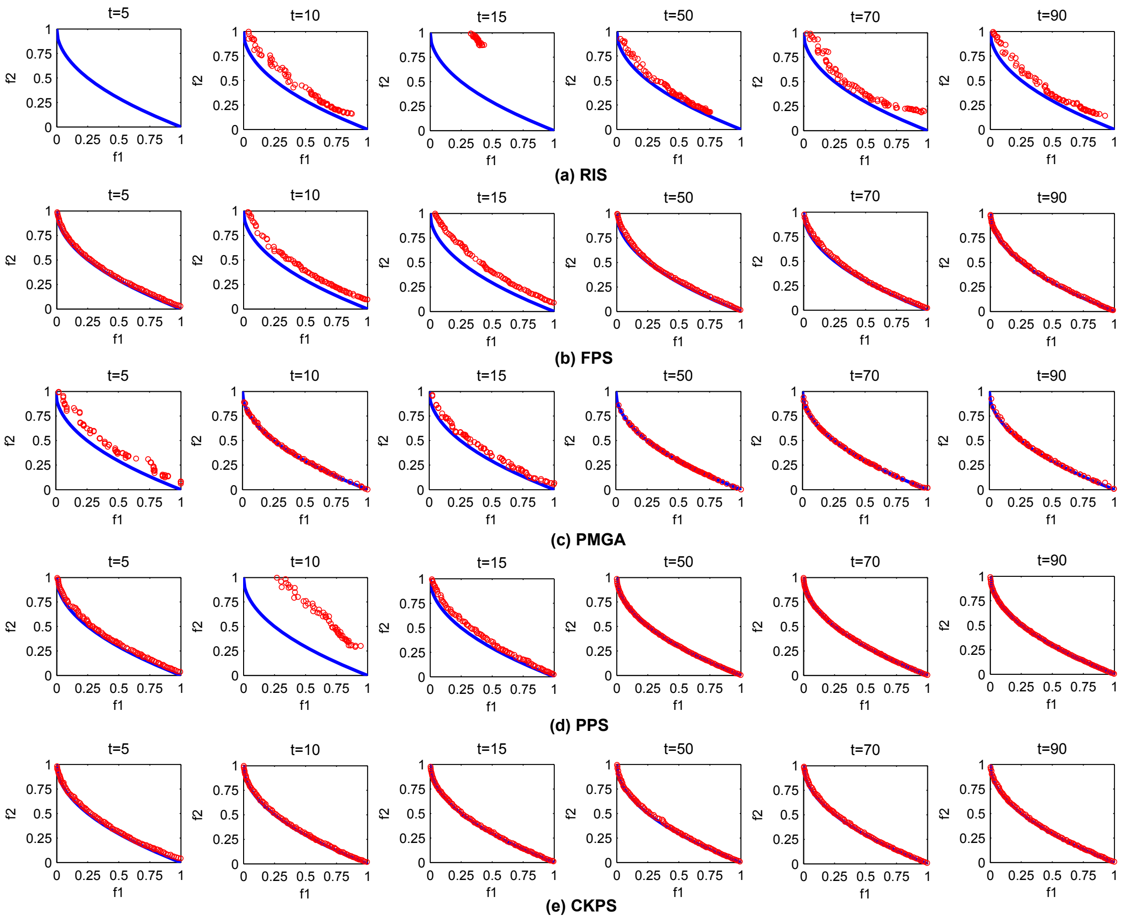


Figure 5: Final population distribution of the five strategies on FDA1.

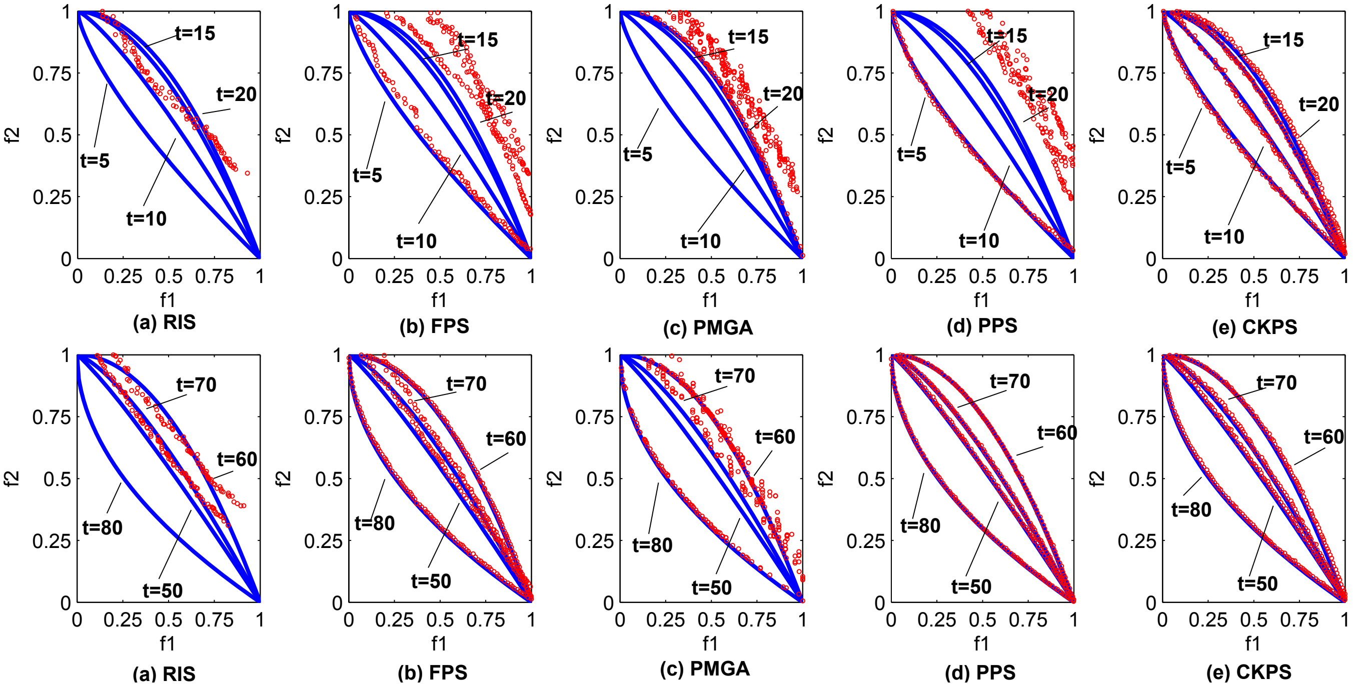
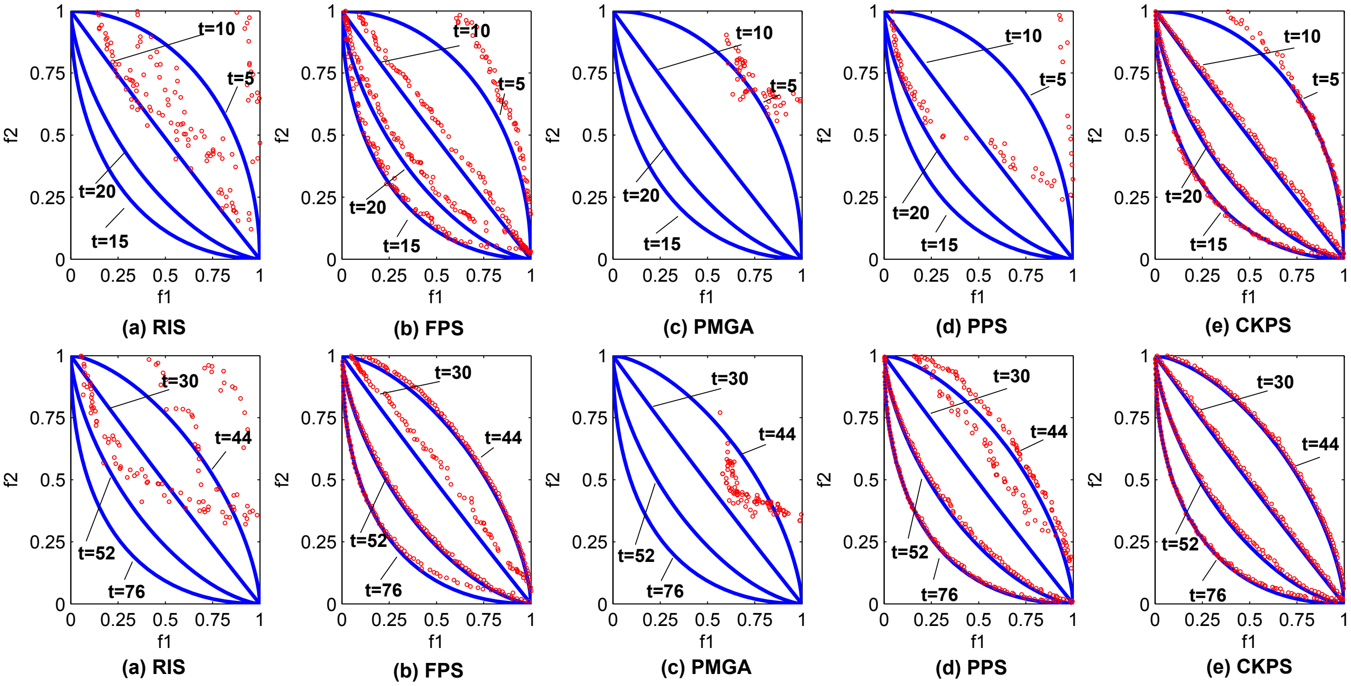
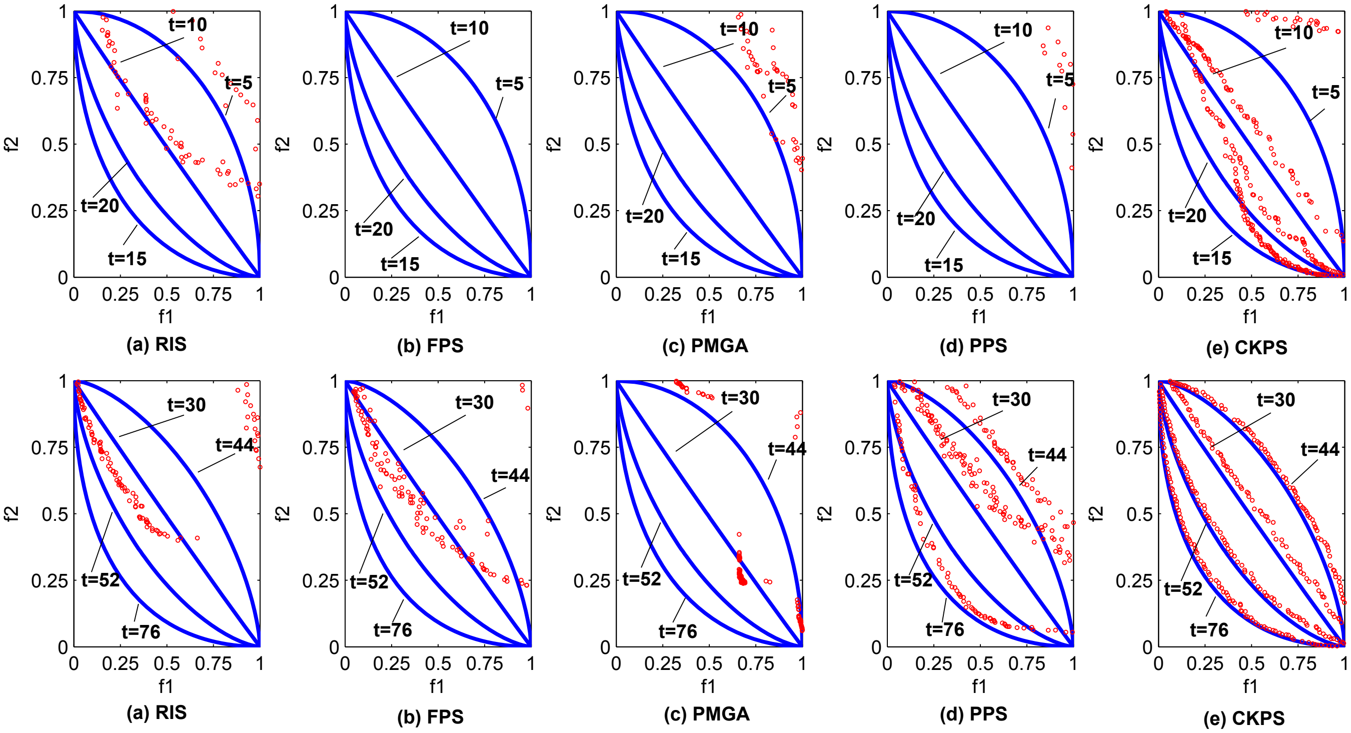


Figure 6: Final population distribution of the five strategies on dMOP2.


(a) RIS (b) FPS (c) PMGA (d) PPS (e) CKPS

(a) RIS (b) FPS (c) PMGA (d) PPS (e) CKPS

Figure 7: Final population distribution of the five strategies on F6.


(a) RIS (b) FPS (c) PMGA (d) PPS (e) CKPS

(a) RIS (b) FPS (c) PMGA (d) PPS (e) CKPS

Figure 8: Final population distribution of the five strategies on F10.

and t step in k dimension, respectively. Gaussion(0,d) denotes a perturbation with expectation 0 and variance d. C_k^t and C_k^{t-1} denote the center points in t and t-1 step in the k-th dimension, respectively. C_k^t is shown as formula (5).

Table 5: Mean and Standard Deviation of MIGD values of CPS and CKPS. ‡ and † indicate CKPS performs significantly better than and equivalently to the corresponding algorithm, respectively.

Problem	Strategy	total	1st stage	2nd stage	3rd stage
FDA1	CPS	0.0292(0.00479)†	0.1068(0.02595)‡	0.0108(0.00012)‡	0.0134(0.00084)‡
	CKPS	**0.0276(0.00360)**	**0.1051(0.01887)**	**0.0092(0.00015)**	**0.0091(0.00012)**
FDA4	CPS	**0.1191(0.00151)**	**0.1306(0.00587)**	0.1164(0.00166)‡	0.1386(0.00340)‡
	CKPS	0.1204(0.00165)	0.1385(0.00561)	**0.1159(0.00203)**	**0.1163(0.00161)**
dMOP1	CPS	0.1135(0.04520)‡	0.0853(0.03992)‡	0.1178(0.08200)‡	**0.0043(0.00007)**
	CKPS	**0.0076(0.00141)**	**0.0183(0.00727)**	**0.0051(0.00005)**	0.0051(0.00006)
F5	CPS	0.0546(0.02595)‡	0.1878(0.13015)‡	0.0245(0.00485)‡	0.0746(0.02526)‡
	CKPS	**0.0366(0.00964)**	**0.1163(0.04642)**	**0.0176(0.00136)**	**0.0178(0.00127)**
F10	CPS	0.8113(0.09099)‡	0.8323(0.19708)‡	0.8288(0.17514)‡	0.3499(0.06662)‡
	CKPS	**0.0834(0.02518)**	**0.2839(0.12102)**	**0.0379(0.00357)**	**0.0338(0.00449)**

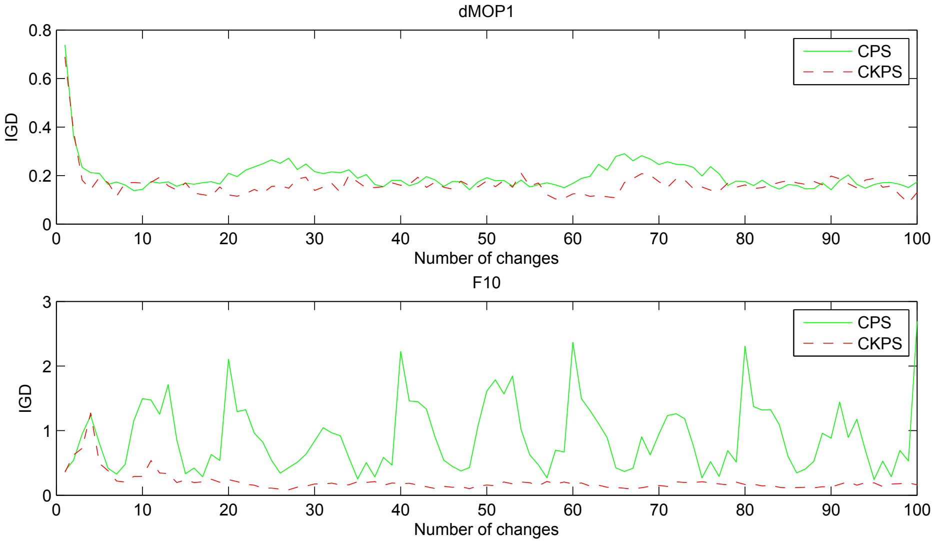


Figure 9: IGD trend comparison of CPS and CKPS over number of changes for 20 runs on dMOP1 and F10.

6.1.2. *Comparison experiment*

The parameters of CKPS are consistent with the experimental parameters in Section 5. Here, we selected five test problems: FDA1, FDA4, dMOP1, F5 and F10. These problems are characterized by various difficulties. As shown in Table 5, CKPS is better than CPS for the most of test problems. For FDA1 and F5, CKPS is better than CPS for all stages. At the beginning stage of FDA4, the effect of CKPS is worse than CPS. However, at the second and third stage of FDA4, CKPS is better than CPS. It indicates that CKPS are effective on the 3-dimensional problem FDA4. For complex problems F10, the experimental results of CKPS are significantly better than CPS. For more intuitive observation, Figure 9 selected dMOP1 and F10 to show the results which was consistent with Tables 1, 2, 3 and 4. Especially for F10, the advantages of CKPS can be clearly seen from Figure 9. The effect of CKPS is much better than CPS.

Exploring its reason to know, the reason that CKPS is better than CPS may be the introduction of the knee point set, which makes CKPS better respond to the environment for relatively simple problems, such as FDA and dMOP test suits. For more complex problems such as F10, the experimental results of CKPS are significantly better than CPS. The specific reason may be the adaptive diversity maintenance strategy of CKPS. Because the adaptive diversity maintenance mechanism of CKPS can introduce random individuals of different numbers according to the degree of difficulty of the problem, CKPS has the powerful ability to deal with complex problems.

6.2. *Comparison of different numbers of knee points in CKPS*

It can be seen from Section 6.1 that the introduction of the knee points in CKPS is undoubtedly a very important part of the strategy in order to make CKPS track the PF more accurately, and respond to the environment faster. For a clearer analysis of the effect of the knee point, we compare the CKPS with the knee points of different numbers.

6.2.1. *A simple comparison of CKPS with knee points of different numbers*

To analysis simply the effect of knee points on CKPS, a simple comparison of CKPS with knee points of different numbers was done. Because the knee point set was as the guided individuals, the number of knee points should not large. In the experiment, the number of knee points varied from zero to nine and dMOP1 and F10 were selected to observe results.

From Fig. 10, we can see that the trend of IGD is downward almost the whole on dMOP1 and F10. When the number of

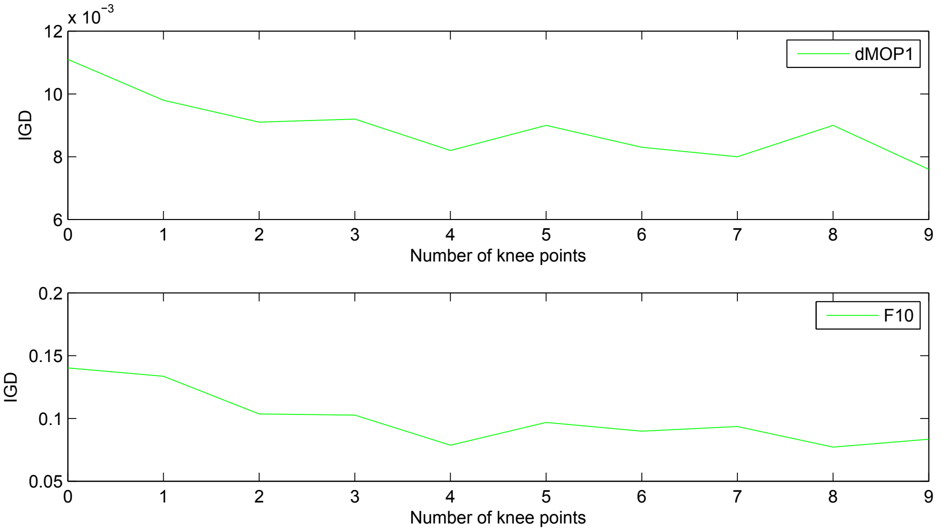


Figure 10: IGD trend comparison of CKPS with knee points of different numbers on dMOP1 and F10.

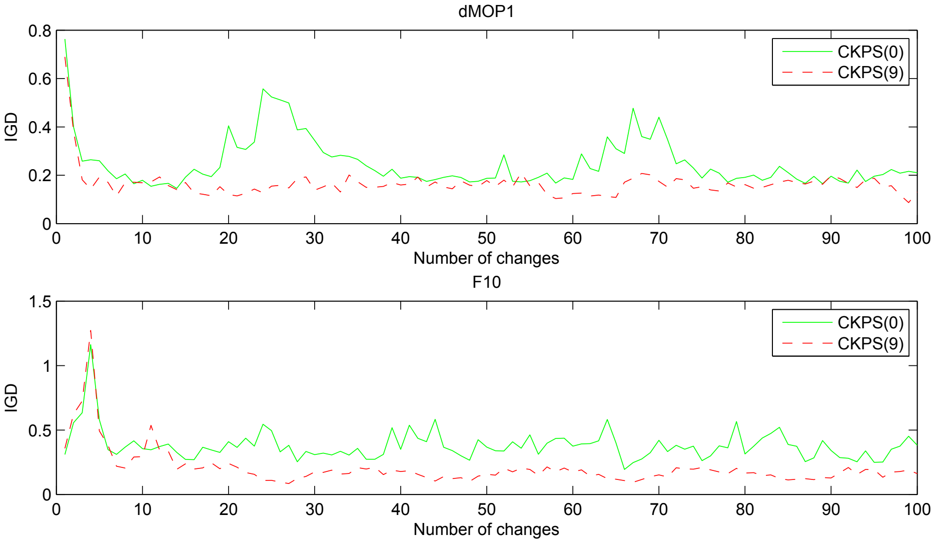


Figure 11: IGD trend comparison of CKPS(0) and CKPS(9) over number of changes for 20 runs on dMOP1 and F10.

knee points is nine, IGD has a smaller value than other numbers. Therefore, the knee point set has the effect to accelerate convergence.

6.2.2. Comparison of two different numbers of knee points in CKPS

We compared the CKPS with zero knee point to the CKPS with nine knee points. Suppose that the number of knee points is k. Let CKPS of k=0 be CKPS(0), and CKPS of k=9 be CKPS(9). Here, we selected FDA1, dMOP1, dMOP2, F5 and F10 to observe results. From Table 6 we can see that, CKPS(9) is better than CKPS(0) for almost all problems. In Fig. 11, two representative test questions, dMOP1 and F10, are chosen to describe the IGD trend comparison of CKPS(0) and CKPS(9) over number of changes for 20 runs. It can be intuitively seen that CKPS(9) is much better than CKPS(0) on dMOP1 and F10.

Above all, we know that the introduction of the knee points in CKPS can make the population response faster to the environment.

7. Conclusions and future work

This paper has presented a novel prediction strategy based on center points and knee points. The non-dominated set predicted by feed-forward center points is the body of the predicted population. The knee point set predicted by prediction model is as the guidance individuals, which makes the strategy track the PF more accurately and respond to the environment faster. The adaptive diversity maintenance strategy can adaptively introduce random individuals of different numbers according to the degree of difficulty of the problem to keep the diversity of

Table 6: Comparison of two different numbers of knee points in CKPS on FDA1, dMOP1, dMOP2, F5, F10 for Mean and Standard Deviation of MIGD values.

Problem	Strategy	total	1st stage	2nd stage	3rd stage
FDA1	CKPS(0)	0.0297(0.00593)†	0.1092(0.03091)‡	0.0109(0.00022)‡	0.0108(0.00013)‡
	CKPS(9)	**0.0276(0.00360)**	**0.1051(0.01887)**	**0.0092(0.00015)**	**0.0091(0.00012)**
dMOP1	CKPS(0)	0.1617(0.03600)‡	0.1337(0.06011)‡	0.1798(0.05438)‡	0.1570(0.07509)‡
	CKPS(9)	**0.0076(0.00141)**	**0.0183(0.00727)**	**0.0051(0.00005)**	**0.0051(0.00006)**
dMOP2	CKPS(0)	0.0332(0.00553)‡	0.1189(0.02839)‡	0.0129(0.00029)‡	0.0128(0.00036)‡
	CKPS(9)	**0.0272(0.00611)**	**0.1028(0.03196)**	**0.0094(0.00023)**	**0.0091(0.00023)**
F5	CKPS(0)	0.0391(0.01514)†	**0.1059(0.06841)**	0.0233(0.00413)‡	0.0233(0.00456)‡
	CKPS(9)	**0.0366(0.00946)**	0.1163(0.04642)	**0.0176(0.00136)**	**0.0178(0.00127)**
F10	CKPS(0)	0.3015(0.06009)‡	0.3500(0.11253)‡	0.3003(0.06600)‡	0.2796(0.06322)‡
	CKPS(9)	**0.0834(0.02518)**	**0.2839(0.12102)**	**0.0379(0.00357)**	**0.03380.00449**

the population. We selected the FDA test suite and dMOP test suite, a total of seven test problems whose decision variables were linearly related. In addition, F5-F10 were selected because their decision variables are nonlinear correlation. It can be seen that CKPS was better than RIS, FPS, PMGA and PPS in total, first and second stage. In the third stage, CKPS was better than PPS for most of the problems, but also slightly worse than PPS on other problems. It was because PPS accumulates enough experience to make it better than CKPS in the third stage. It will be also the disadvantage of CKPS. This is the focus of our future work. In Section 6, CKPS and CPS were compared with each other. In addition, CKPS(0) and CKPS(9) were compared. We have found that the introduction of the knee points in CKPS can give more rapid response to the environment. How to better partition knee points and how to better track PF, so as to better reflect the advantage of the knee point is another important work for future study.

Acknowledgements

The authors wish to thank the support of the National Natural Science Foundation of China (Grant No. 61502408, 61673331), the Education Department Major Project of Hunan Province (Grant No. 17A212), CERNET Innovation Project (Grant No. NGII20150302), The MOE Key Laboratory of Intelligent Computing and Information Processing, the Science and Technology Plan Project of Hunan Province (Grant No. 2016TP1020), the Provinces and Cities Joint Foundation Project (Grant No. 2017JJ4001).

References

[1] J. Zou, J. Zheng, C. Deng, R. Shen, An evaluation of non-redundant objective sets based on the spatial similarity ratio, Soft Computing 19 (8) (2015) 2275–2286.

[2] L. T. Bui, Z. Michalewicz, E. Parkinson, M. B. Abello, Adaptation in dynamic environments: a case study in mission planning, IEEE Transactions on Evolutionary Computation 16 (2) (2012) 190–209.

[3] M. B. Abello, L. T. Bui, Z. Michalewicz, An adaptive approach for solving dynamic scheduling with time-varying number of taskspart ii, in: Evolutionary Computation (CEC), 2011 IEEE Congress on, IEEE, 2011, pp. 1711–1718.

[4] K. Deb, S. Karthik, et al., 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.

[5] Z. Zhang, Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control, Applied Soft Computing 8 (2) (2008) 959–971.

[6] F. V. Martins, E. G. Carrano, E. F. Wanner, R. H. Takahashi, G. R. Mateus, A dynamic multiobjective hybrid approach for designing wireless sensor networks, in: Evolutionary Computation, 2009. CEC'09. IEEE Congress on, IEEE, 2009, pp. 1145–1152.

[7] K. Kim, R. I. McKay, B.-R. Moon, Multiobjective evolutionary algorithms for dynamic social network clustering, in: Proceedings of the 12th annual conference on Genetic and evolutionary computation, ACM, 2010, pp. 1179–1186.

[8] B. S. Rabil, R. Sabourin, E. Granger, Watermarking stack of grayscale face images as dynamic multi-objective optimization problem., in: MDA, 2011, pp. 63–77.

[9] S. Zeng, S. Chen, J. Zhao, A. Zhou, Z. Li, H. Jing, Dynamic constrained multi-objective model for solving constrained optimization problem, in: Evolutionary Computation (CEC), 2011 IEEE Congress on, IEEE, 2011, pp. 2041–2046.

[10] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii, in: International Conference on Parallel Problem Solving From Nature, Springer, 2000, pp. 849–858.

[11] E. Zitzler, M. Laumanns, L. Thiele, Spea2: Improving the strength pareto evolutionary algorithm.

[12] J. Branke, Evolutionary optimization in dynamic environments, Vol. 3, Springer Science & Business Media, 2012.

[13] C.-K. Goh, K. C. Tan, Evolutionary multi-objective optimization in uncertain environments, Issues and Algorithms, Studies in Computational Intelligence 186 (2009) 5–18.

[14] Y. Jin, J. Branke, Evolutionary optimization in uncertain environments-a survey, IEEE Transactions on evolutionary computation 9 (3) (2005) 303–317.

[15] T. T. Nguyen, S. Yang, J. Branke, Evolutionary dynamic optimization: A survey of the state of the art, Swarm and Evolutionary Computation 6 (2012) 1–24.

[16] C. A. C. Coello, G. B. Lamont, D. A. Van Veldhuizen, et al., Evolutionary algorithms for solving multi-objective problems, Vol. 5, Springer, 2007.

[17] V. S. Aragón, S. C. Esquivel, C. Coello Coello, Evolutionary multiobjetive optimization in non-stationary environments, Journal of Computer Science & Technology 5.

[18] C. R. Azevedo, A. F. Araújo, Generalized immigration schemes for dynamic evolutionary multiobjective optimization, in: Evolutionary Computation (CEC), 2011 IEEE Congress on, IEEE, 2011, pp. 2033–2040.

[19] M. Cámara, J. Ortega, F. de Toro, Approaching dynamic multi-objective optimization problems by using parallel evolutionary algorithms, Advances in Multi-Objective Nature Inspired Computing (2010) 63–86.

[20] M. Cámara, J. Ortega, F. de Toro, A single front genetic algorithm for parallel multi-objective optimization in dynamic environments, Neurocomputing 72 (16) (2009) 3570–3579.

[21] Y. Jin, B. Sendhoff, Constructing dynamic optimization test problems using the multi-objective optimization concept, in: Workshops on Applications of Evolutionary Computation, Springer, 2004, pp. 525–536.

[22] L. Chun'an, W. Yuping, Multiobjective evolutionary algorithm for dynamic nonlinear constrained optimization problems, Journal of Systems Engineering and Electronics 20 (1) (2009) 204–210.

[23] C. Liu, Y. Wang, New evolutionary algorithm for dynamic multiobjective optimization problems, in: Advances in Natural Computation, Second International Conference, Icnc 2006, Xi'an, China, September 24-28, 2006. Proceedings, 2006, pp. 889–892.

[24] M. Greeff, A. P. Engelbrecht, Solving dynamic multi-objective problems

with vector evaluated particle swarm optimisation, in: Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on, IEEE, 2008, pp. 2917–2924.

[25] M. Helbig, A. P. Engelbrecht, Analyses of guide update approaches for vector evaluated particle swarm optimisation on dynamic multi-objective optimisation problems, in: Evolutionary Computation (CEC), 2012 IEEE Congress on, IEEE, 2012, pp. 1–8.

[26] C. A. C. Coello, G. T. Pulido, M. S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Transactions on evolutionary computation 8 (3) (2004) 256–279.

[27] E. Vinek, P. P. Beran, E. Schikuta, A dynamic multi-objective optimization framework for selecting distributed deployments in a heterogeneous environment, Procedia Computer Science 4 (2011) 166–175.

[28] A. Isaacs, V. Puttige, T. Ray, W. Smith, S. Anavatti, Development of a memetic algorithm for dynamic multi-objective optimization and its applications for online neural network modeling of uavs, in: Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, IEEE, 2008, pp. 548–554.

[29] S.-U. Guan, Q. Chen, W. Mo, Evolving dynamic multi-objective optimization problems with objective replacement, Artificial Intelligence Review 23 (3) (2005) 267–293.

[30] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, E. Tsang, Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2007, pp. 832–846.

[31] I. Hatzakis, D. Wallace, Topology of anticipatory populations for evolutionary dynamic multi-objective optimization, in: 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2006, p. 7071.

[32] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, E. Tsang, Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2007, pp. 832–846.

[33] Y. Ma, R. Liu, R. Shang, A hybrid dynamic multi-objective immune optimization algorithm using prediction strategy and improved differential evolution crossover operator, in: International Conference on Neural Information Processing, Springer, 2011, pp. 435–444.

[34] R. Liu, W. Zhang, L. Jiao, F. Liu, J. Ma, A sphere-dominance based preference immune-inspired algorithm for dynamic multi-objective optimization, in: Proceedings of the 12th annual conference on Genetic and evolutionary computation, ACM, 2010, pp. 423–430.

[35] J. Wei, M. Zhang, Simplex model based evolutionary algorithm for dynamic multi-objective optimization., in: Australasian Conference on Artificial Intelligence, Springer, 2011, pp. 372–381.

[36] I. Hatzakis, D. Wallace, Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach, in: Proceedings of the 8th annual conference on Genetic and evolutionary computation, ACM, 2006, pp. 1201–1208.

[37] A. Zhou, Y. Jin, Q. Zhang, A population prediction strategy for evolutionary dynamic multiobjective optimization, IEEE transactions on cybernetics 44 (1) (2014) 40–53.

[38] Y. Wu, Y. Jin, X. Liu, A directed search strategy for evolutionary dynamic multiobjective optimization, Soft Computing 19 (11) (2015) 3221–3235.

[39] Y. Wu, X.-X. Liu, C.-Z. Chi, Predictive multiobjective genetic algorithm for dynamic multiobjective optimization problems, Control and Decision 28 (5) (2013) 677–682.

[40] J. H. Zheng, Z. Peng, J. Zou, R. M. Shen, A prediction strategy based on guide-individual for dynamic multi-objective optimization.

[41] L. B. Said, S. Bechikh, K. Ghédira, The r-dominance: a new dominance relation for interactive evolutionary multicriteria decision making, IEEE Transactions on Evolutionary Computation 14 (5) (2010) 801–818.

[42] J. Molina, L. V. Santana, A. G. Hernández-Díaz, C. A. C. Coello, R. Caballero, g-dominance: Reference point based dominance for multiobjective metaheuristics, European Journal of Operational Research 197 (2) (2009) 685–692.

[43] L. Thiele, K. Miettinen, P. J. Korhonen, J. Molina, A preference-based evolutionary algorithm for multi-objective optimization, Evolutionary computation 17 (3) (2009) 411–436.

[44] J. Zheng, G. Yu, Q. Zhu, X. Li, J. Zou, On decomposition methods in interactive user-preference based optimization, Applied Soft Computing 52 (2017) 952–973.

[45] G. Yu, J. Zheng, R. Shen, M. Li, Decomposing the user-preference in multiobjective optimization, Soft Computing 20 (10) (2016) 4005–4021.

[46] K. Deb, S. Gupta, Understanding knee points in bicriteria problems and their implications as preferred solution principles, Engineering optimization 43 (11) (2011) 1175–1204.

[47] L. While, P. Hingston, L. Barone, S. Huband, A faster algorithm for calculating hypervolume, IEEE transactions on evolutionary computation 10 (1) (2006) 29–38.

[48] X. Zhang, Y. Tian, Y. Jin, A knee point-driven evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation 19 (6) (2015) 761–776.

[49] I. Das, On characterizing the knee of the pareto curve based on normal-boundary intersection, Structural and Multidisciplinary Optimization 18 (2) (1999) 107–115.

[50] J. Branke, K. Deb, H. Dierolf, M. Osswald, et al., Finding knees in multi-objective optimization, in: PPSN, Vol. 3242, 2004, pp. 722–731.

[51] M. Farina, K. Deb, P. Amato, Dynamic multiobjective optimization problems: test cases, approximations, and applications, IEEE Transactions on evolutionary computation 8 (5) (2004) 425–442.

[52] C.-K. Goh, K. C. Tan, A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization, IEEE Transactions on Evolutionary Computation 13 (1) (2009) 103–127.

[53] Z. Peng, J. Zheng, J. Zou, M. Liu, Novel prediction and memory strategies for dynamic multiobjective optimization, Soft Computing 19 (9) (2015) 2633–2653.

[54] Z. Peng, J. Zheng, J. Zou, A population diversity maintaining strategy based on dynamic environment evolutionary model for dynamic multiobjective optimization, in: Evolutionary Computation (CEC), 2014 IEEE Congress on, IEEE, 2014, pp. 274–281.

[55] Q. Zhang, A. Zhou, Y. Jin, Rm-meda: A regularity model-based multiobjective estimation of distribution algorithm, IEEE Transactions on Evolutionary Computation 12 (1) (2008) 41–63.

[56] Y. Yuan, H. Xu, B. Wang, B. Zhang, X. Yao, Balancing convergence and diversity in decomposition-based many-objective optimizers, IEEE Transactions on Evolutionary Computation 20 (2) (2016) 180–198.

[57] G. Ruan, G. Yu, J. Zheng, J. Zou, S. Yang, The effect of diversity maintenance on prediction in dynamic multi-objective optimization, Applied Soft Computing 58 (2017) 631–647.

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

[59] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics bulletin 1 (6) (1945) 80–83.