

Article

A New Method for Dynamic Multi-Objective Optimization Based on Segment and Cloud Prediction

Peng Ni ¹, Jiale Gao ², Yafei Song ^{1,*} , Wen Quan ^{3,*} and Qinghua Xing ¹

¹ Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China; nipeng8866@163.com (P.N.); afeu_xqh@163.com (Q.X.)

² No. 95806 of PLA, Beijing 100021, China; gaojianle@163.com

³ Air Traffic Control and Navigation College, Air Force Engineering University, Xi'an 710051, China

* Correspondence: yafei_song@163.com (Y.S.); afeu_quanwen@163.com (W.Q.)

Received: 27 January 2020; Accepted: 11 March 2020; Published: 16 March 2020



Abstract: In the real world, multi-objective optimization problems always change over time in most projects. Once the environment changes, the distribution of the optimal solutions would also be changed in decision space. Sometimes, such change may obey the law of symmetry, i.e., the minimum of the objective function in such environment is its maximum in another environment. In such cases, the optimal solutions keep unchanged or vibrate in a small range. However, in most cases, they do not obey the law of symmetry. In order to continue the search that maintains previous search advantages in the changed environment, some prediction strategy would be used to predict the operation position of the Pareto set. Because of this, the segment and multi-directional prediction is proposed in this paper, which consists of three mechanisms. First, by segmenting the optimal solutions set, the prediction about the changes in the distribution of the Pareto front can be ensured. Second, by introducing the cloud theory, the distance error of direction prediction can be offset effectively. Third, by using extra angle search, the angle error of prediction caused by the Pareto set nonlinear variation can also be offset effectively. Finally, eight benchmark problems were used to verify the performance of the proposed algorithm and compared algorithms. The results indicate that the algorithm proposed in this paper has good convergence and distribution, as well as a quick response ability to the changed environment.

Keywords: dynamic multi-objective optimization; cloud theory; linear search; dynamic multi-objective; segment prediction

1. Introduction

Dynamic multi-objective optimization problems (DMOOPs) belong to a kind of time-varying multi-objective optimization problems that is frequently encountered in the disciplines of science and engineering. These problems not only display the conflict in optimization targets and high dimension in solution space, but also have the time-varying characteristics in optimization objective, constraints, and decision space [1–7]. As a result, the high conditionality of traditional algorithms makes it difficult to meet requirements about large-scale, high timeliness and complex non-deterministic polynomial (NP) hard problems. Compared with traditional algorithms, evolution methods have low requirements in problem model, high efficiency, as well as embarrassingly parallel self-organizing and self-adaption, so they have been widely used in the modern industry and scientific research field [8–10].

The easiest way is to keep the diversity of the population in evolution methods used to solve DMOOPs. For example, DNSGA-II (dynamic multi-objective optimization and decision-making using the modified NSGA-II) proposed by Deb [11] uses partial random initialization or random variation in order to enhance the flexibility of the population in the changed environment. Moreover, three

immigration schemes proposed by Azevedo are used to initialize the population in the changed environment [12], as well as the immune clonal algorithm proposed by Shang Rong-hua [13]. Because of the unknown variation trend for the dynamic environment, those methods can be formulated into an attempted exploration method, which could ensure the superiority individual could be reserved in the changed environment. The upside for these methods is that they can adapt to a variety of changing environments, while their downside is that advantages of search before environmental change would be difficult to keep after change. Thus, the characteristics of this kind of methods are their powerful environmental adaptability and poor prediction accuracy.

Compared with the early methods, by maintaining the diversity of the population, historical information from multiple environmental changes is used to predict the position of the population in the next environmental change. That is, it is a type of statistical method to study the non-dominant solution set. There are two solutions for this, namely the memory strategy and the prediction strategy.

The memory strategy predicts the position of the population in the next environmental change by rules, which stores a lot of information about non-dominant solution set. The memory strategy is suited to solve the dynamic problem of “short periodicity”. Here, “Periodicity” could be described by the historical information, and “short” indicates less storage [14–16].

However, this has several drawbacks, namely large memory consumption and incorrect predictions to solve time-varying nonlinear problems or more frequent environmental changes.

The prediction strategy uses historical information by statistical analysis and predicts the possible direction of population migration using prediction theory (for example, linear regression, autoregressive model, etc.), relative to the memory strategy. The FPS (feed-forward prediction strategy) proposed by Hatzakis [17] is used to predict the migration locations of the non-dominant solution set, which is based on an autoregressive model. Because of the incomplete statistical analysis about historical information, the FPS is weak and ineffective. The PPS (population prediction strategy) proposed by Zhou [18] uses an autoregressive model to predict the central migration locations of the population, and shape estimation to migrate the whole population. Compared with the FPS, the PPS has mainly two characteristics: (1) further treatment of historical information, which is used to predict the central locations of the population; and (2) shape estimation, which is better than estimation one by one. All of these methods based on autoregressive model have a fundamental disadvantage, namely, a large amount of statistical information is needed to construct the model. Moreover, if the prediction error is increasing repeatedly, the prediction error will be amplified. As a result, it is difficult to adjust when the change rule of the environment is changed.

In short, the autoregressive model is widely used to solve regular change problems, but has the problems of slow response speed and large amounts of statistical data. To compensate, the diversity-keeping strategy is widely used. The DSS (directed search strategy) proposed by Wu [19] is used to solve the dynamic problem, which uses both central population prediction and cross search. Moreover, the method of population center prediction is a linear prediction. Compared with MLR (multiple linear regression), it needs less historical information and has a better adaptive capacity to the environment. For DMOOPS, a method proposed by Rong [20] is used to predict the changes, which uses a linear model based on segmentation of the optimal front. Furthermore, the linear model used by Li [21] is based on the prediction of the autoregressive model. It predicts the key points of the optimal front, such as the boundary point, inflection point, etc. Also, the linear model is used by Ruan [22] to predict the position of the optimal front. In maintaining population diversity, the changes in population are predicted based on the extreme point of the previous generation [20]. However, it is worth noting that the prediction error should always exist in the linear model when solving dynamic problems (progressive increase, progressive decrease, or more complex changes). Therefore, all the literature above employs the diversity keeping scheme to redress prediction errors.

According to the problem of remedying the shortage of the linear model, the segmentation cloud predict strategy (SCPS) is proposed in this paper to improve the precision of prediction. First, the searched optimal front would be segmented to divide the population. Second, a new linear prediction

analysis is applied to each part of the population, which would draw on the thoughts of the cloud model. According to the notion of uncertain time in the cloud model, entropy (or super entropy) would be used to redress the prediction errors. Moreover, for some abrupt nonlinear changes in dynamic problems, the extra angle search strategy is used to ensure the diversity of the population during the prediction process.

The rest of this paper is organized as follows. The problem of dynamic multi-objective is described in Section 2. The segmentation cloud prediction strategy is proposed in Section 3. Benchmark problems are applied in Section 4 to show the performance of the proposed method, together with related experimental analysis. Conclusions are provided in the last section.

2. Description of Dynamic Multi-Objective Problem

Considering the symmetry between minimum and maximum under symmetric environments, the optimization problem can be normalized to a unit form. Generally, the DMOP could be described as follows,

$$\begin{cases} \min F(x, t) = (f_1(x, t), \dots, f_m(x, t))^T \\ \text{s.t. } g(v, t) \leq 0, h(v, t) = 0 \end{cases} \quad (1)$$

where the decision variables are $x = (x_1, \dots, x_n)^T \in \Omega$, Ω is the n dimension decision space, and the time variable is t . $g(v, t)$ is the equality constraint, while $h(v, t)$ is the inequality constraint. The objective function vector is $y = (f_1, \dots, f_m)^T \in \Lambda$, and Λ is the m dimension decision space. The evaluation function $F(x, t) : \Omega \times T \rightarrow \Lambda$ would define the mapping from decision space to object space [23].

Definition 1. (Pareto domination). For some time t , if individual p , it can as $p < q$. If and only if $\forall i = \{1, 2, \dots, m\}: f_i(p, t) \leq f_i(q, t), \exists j = \{1, 2, \dots, m\}: f_j(p, t) < f_j(q, t)$.

Definition 2. (Pareto optimal solution set, PS). Let $x \in \Omega$, which is the decision variable. PS could be defined as follows,

$$\text{PS} := \{x \in \Omega \mid \neg \exists y \in \Omega, y < x\} \quad (2)$$

Definition 3. (Pareto optimal front, PF). Let $x \in \Omega$, which is the decision variable. PF could be defined as follows,

$$\text{PF} := \{y = F(x, t) \mid x \in \text{PS}\} \quad (3)$$

3. Segmentation Cloud Prediction Strategy

First, the optimal front is segmented into multiple fragments, and the segmentation result can be used to divide the population in the decision space. Second, certain individuals are chosen to search the change directions of the optimal front by extra angle search strategy. At last, some add search is conducted to search for the possible offset position of the optimal front by extra angle search.

3.1. Population Segmentation

For the DMOOPs, the optimal front might be deflected or deformed. In order to predict the variation of PF more accurately, the PF is segmented into multiple fragments, and each fragment is used to make predictions. Compared with the linear prediction for the population center, this method would have a more accurate predictive ability of PF deformation and nonlinear migration.

First, for PF, a boundary point is chosen as the first key point, and the distance between this point and the others is calculated. Then, the point with the maximum distance is the second key point. In the same manner of choosing second key point, the third key point would be the point with maximum sum of distance between the two key points, and so on for $m+1$ key points of the m objective functions

$(y = (f_1, \dots, f_m)^T)$. As a result, each individual would choose the key point with minimum distance to constitute $m+1$ fragments. Thus, populations would be segmented according to $m+1$ key points.

This method has simple computational properties and low complexity. The time complexity for choosing each key point is $O(N)$, where N is the population size. The computation complexity of this method is $O(N(m+2)/2)=O(Nm)$, while the computation complexity of the clustering method is $O(N^2)$. Therefore, the proposed method of segmentation is simpler for population segmentation and has lower computation complexity.

3.2. Directional Cloud Prediction Strategy

Suppose the population size is N , and the i th population Pop_i contains K_i individuals. The center of sub-population Pop_i could be described as follows,

$$C_i(t) = \frac{1}{|Pop_i|} \sum_{x \in Pop_i} x \tag{4}$$

where $Pop_i = \{x_{k1}^i, \dots, x_{kK_i}^i\}$ are all the individuals of the population, the k th individual is $x_k^i = (x_{k1}^i, \dots, x_{kn}^i)$, $C_i(t)$ is the center position of population at time t , and $|\cdot|$ is the cardinality of set.

The predicted migration vector of the population $d_i(t)$ and the predicted error of two migrations $\Delta(t)$ at time $t+1$ could be calculated as follows, based on the predictions of center positions of populations at time t and time $t-1$.

$$\begin{cases} d_i(t) = C_i(t) - C_i(t-1) \\ \Delta(t) = d_i(t) - d_i(t-1) \end{cases} \tag{5}$$

The prediction of the migration position of the population is based on the normal cloud model. Suppose the moving direction of the population is expectation, the Euclidean distance of motion vector $d_i(t)$ is entropy, and the deviation $\Delta(t)$ between two moving direction of the population is super entropy. The normal cloud generator could be built by expectation, entropy, and super entropy. First, we generate normal random vectors $En' \sim N(\Delta(t), \|\Delta(t)\|^2/n^2)$ whose expectation is $\Delta(t)$ and standard deviation is $\|\Delta(t)\|/n$, where $En' = (En'_1, \dots, En'_n)$. Then, we generate normal random vectors $v_1 \sim N(d_i(t), En'^2)$, whose expectation is $d_i(t)$ and standard deviation is En' . Therefore, the optimal solution position of predicted sub-population could be shown as follows,

$$y = x + v_1 \tag{6}$$

where the individual position before change is x , the direction of prediction is v_1 , and the individual position of prediction after change is y .

Figure 1 shows the schematic diagram of directional cloud search, where the direction of directional prediction is $d(t)$, and the possible direction of cloud prediction is $d'(t)$. That is, the main search direction of directional cloud prediction is based on $d(t)$, and it could search at a certain probability to predict the position for the last offset direction $\Delta(t)$.

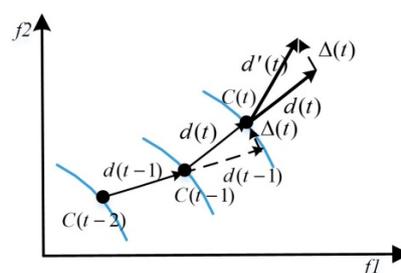


Figure 1. Directional cloud search.

3.3. Extra Angle Search

Directional cloud prediction contains the error compensation of linear prediction. However, this is only effective for gradual and regular changes. If the dynamic problem shows inverted reciprocating or non-gradual changes, the direction of directional cloud prediction would be opposite or vertical to the change direction of the dynamic problem. To deal with this problem, the extra angle search strategy would be purposed. That is, based on some random chosen individuals, it would search all possible directions in a stated angle. Figure 2 shows the schematic diagram of the extra angle search strategy.

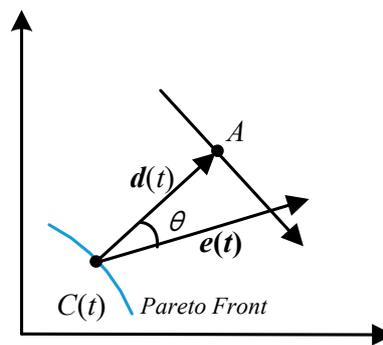


Figure 2. Extra angle search.

First, the random vectors are constructed, perpendicular to $d_i(t)$. Based on the random vector $r = (r_1, \dots, r_n)$ within the scope of $[-1, 1]$, the j th dimension of r would be assigned as Equation (7),

$$r_j = -\sum_{i \neq j}^n d_i(t) r_i(t) \quad (7)$$

The direction vector of extra angle search would be calculated as follows,

$$e_i(t) = d_i(t) + \frac{r}{\|r\|} \|d_i(t)\| \cot(\theta) \quad (8)$$

where θ is deflection angel, which means the possible range of search and can be calculated as

$$\theta_i = \frac{d_i(t-1) \cdot d_i(t-2)}{\|d_i(t-1)\| \cdot \|d_i(t-2)\|} \quad (9)$$

When the direction vector of extra angle search is determined, we use the normal cloud model to search the angular deviation of in all individuals. The normal random vector $En^* \sim N(d_i(t), \Delta(t)^2)$ is generated, whose expectation is $e_i(t)$ and standard deviation is $\Delta(t)$ and $En^* = (En_1^*, \dots, En_n^*)$. Then, the normal random vector $v_2 \sim N(e_i(t), En^{*2})$ can be generated, whose expectation is $e_i(t)$ and standard deviation is En^* . The position of prediction from the angular deviation search can be calculated as follow,

$$y^* = x + v_2 \quad (10)$$

where the position of individuals before change is x , the direction of prediction is v_2 , and the position of prediction after change is y^* .

3.4. Environmental Detection

The sensitivity of the environment is very important for the algorithm. The computational efficiency of DMOP will decrease if the sensitivity is excessive or insufficient. Moreover, each DMOP might not change at the same time, and a different objective function would have different amplitude

of variation. Therefore, the change in values of each objective function would be overall considered and normalized. The environmental sensitivity can be calculated as follows

$$\varepsilon(t) = \frac{\sum_{j=1}^H \|F(x_j, t) - F(x_j, t-1)\|}{H \max_{1 \leq i \leq H} \|F(x_i, t) - F(x_i, t-1)\|}. \quad (11)$$

where H is the number of individuals is randomly selected for the population. Random selection mainly reduces the computing cost for environment detection. The vector of the objective function for individual x_j at time t is $F(x_j, t)$. Generally, the proportion of the population selected for environment detection is 5%.

3.5. SCPS Framework

The SCPS will iterate over the basic algorithm framework of the dynamic multi-objective evolutionary algorithm (MOEA). The basic algorithm framework of the dynamic MOEA (DMOEA) mainly includes two parts: dynamic prediction and static MOEA search. This paper mainly focused on the prediction performance of DMOEA, so the classical NSGA-II [24] can be selected as static MOEA search. The SCPS is described in detail as below.

Input: When enough historical information cannot be collected, the proportion of the population in random initialization is ζ , the proportion of directional prediction population is L_1 , the population size is N , and the final time of environmental change is T_{max} , $t:= 0$, $d(t-1)= 0$.

Output: PS.

Step 1: Randomly initialize the population.

Step 2: According to Equation (11), detect if the environment changes or not. If change, turn to Step 3; otherwise, turn to Step 8.

Step 3: If the value of $d(t-1)$ is 0, turn to Step 4; otherwise turn to Step 5.

Step 4: Randomly select $\zeta \times N$ individuals to evolve, let $d(t-1) = d(t)$.

Step 5: According to Section 1, segment the population into $m+1$ parts. Calculate the center of the population by Equation (4), and calculate the moving direction of population at time t . Then select $L_1 \times N$ individuals with the binary tournament selection model and predict the position with the cloud model.

Step 6: Select $N \times (1 - L_1)$ individuals with the binary tournament selection model and calculate the search vector of extra angle search for these individuals using Equation (7) to Equation (9).

Step 7: Boundary detect for new individuals.

Step 8: Calculate the non-dominated sort and crowding distance. Hold the first n individuals.

Step 9: Termination conditional judgment. If meet, turn to Step 10. Otherwise turn to Step 2.

Step 10: End, output the population PS.

One-step prediction (Step 5) and search of deflection angle (Step 6) would randomly select $K_1 \times N$ and $N \times (1 - K_1)$ individuals to predict, respectively. As the predicted individuals might be beyond the decision space, the Step 7 is to boundary detect new individuals, and revise the ones beyond.

$$y_i = \begin{cases} x_i & \text{if } a_i \leq x_i \leq b_i \\ 0.5(a_i + x_i) & \text{if } y_i < a_i \\ 0.5(b_i + x_i) & \text{if } y_i > b_i \end{cases} \quad (12)$$

where the i th dimension of individual x before prediction is x_i , the i th dimension predicted position of individual is y_i , $i = \{1, \dots, n\}$, and a_i and b_i are the upper and lower bound of the i th dimension of the decision variable, respectively.

4. Experimental Analysis

4.1. Benchmark Problems

In [25], a new dynamic benchmark problem (generator) was constructed. Many dynamic benchmark problems grown from this generator are related to various practical engineering problems such as the greenhouse system, hydraulic and hydroelectric engineering, and line scheduling.

4.2. Parameter Setting

In order to test the searching performance of the proposed algorithm in dynamic problems more effectively, the environmental change degree and frequentness of 3 groups were set as benchmark problems, and (n, τ) respectively were (5,10), (10,10), (10,20). Multi-objective evolutionary algorithms based on decomposition (MOEA/D) [26], PPS [18], multi-direction prediction (MDP) [20], and SCPS were chosen as the comparing algorithms in the test. The population sizes of all the comparing algorithms were 200 ($N = 200$), and the evolution termination time $T_{\max} = 10$.

- (1) MOEA/D: $T = 20$.
- (2) PPS: retain number of population center $M = 23$, and the model is p linear regression model, $p = 3$.
- (3) SCPS: $\zeta = 0.2$, and the chosen individual number of prediction models $K_1 = 0.5N$.

4.3. Metrics

There are numerous metrics for dynamic MOEA. The inverted generational distance (IGD) is chosen as the evaluative criteria of each iteration, and the modified IGD (MIGD) is used to evaluate each algorithm running several times in each benchmark problem [18].

$$IGD(P(t), P^*(t)) = \frac{\sum_{x \in P^*} d(x, P(t))}{|P^*(t)|} \quad (13)$$

where the equally distributed ideal PF solution set at time t is $P^*(t)$, and the result of the algorithm is $P(t)$. The distance between individual x and solution set $P(t)$ is $d(x, P(t)) = \min_{a \in P(t)} \|F(x, t) - F(a, t)\|$, and the cardinality of set $P^*(t)$ is $|P^*(t)|$. The mean IGD of the algorithm, which would run in the benchmark problem for a period of time is evaluated by MIGD. MIGD is defined as follows,

$$MIGD = \frac{1}{|T|} \sum_{t \in T} IGD(P(t), P^*(t)) \quad (14)$$

where a group of discrete points in time is T , and the potential of T is $|T|$.

4.4. Test and Analysis

Table 1 shows the mean MIGD results of SCPS and other compared methods running 20 times in eight benchmark problems. The smaller the value of MIGD, the higher the average prediction accuracy. The optimal values are highlighted in bold.

Table 1. Results of all the compared algorithms regarding MIGD.

Function	n, τ	SCPS		MDP		PSS		MOEA/D	
		Mean Value	Variance	Mean Value	Variance	Mean Value	Variance	Mean Value	Variance
JY1	5,10	1.75×10^{-2}	3.99×10^{-4}	4.87×10^{-2}	1.18×10^{-3}	2.03×10^{-2}	4.79×10^{-3}	5.16×10^{-2}	2.06×10^{-3}
	10,10	7.55×10^{-3}	1.31×10^{-4}	1.64×10^{-2}	3.86×10^{-4}	3.17×10^{-2}	9.98×10^{-3}	3.43×10^{-2}	1.16×10^{-3}
	10,20	4.28×10^{-3}	6.01×10^{-5}	8.50×10^{-3}	1.30×10^{-4}	8.73×10^{-3}	1.53×10^{-3}	2.34×10^{-2}	1.77×10^{-3}
JY2	5,10	4.76×10^{-2}	2.15×10^{-4}	6.17×10^{-2}	7.78×10^{-4}	5.09×10^{-2}	9.68×10^{-4}	7.15×10^{-2}	1.27×10^{-3}
	10,10	7.39×10^{-3}	5.10×10^{-5}	1.66×10^{-2}	3.06×10^{-4}	3.09×10^{-2}	1.38×10^{-2}	3.30×10^{-2}	6.96×10^{-4}
	10,20	4.13×10^{-3}	3.05×10^{-5}	8.30×10^{-3}	1.28×10^{-4}	8.47×10^{-3}	1.27×10^{-3}	2.48×10^{-2}	1.83×10^{-3}
JY3	5,10	9.17×10^{-3}	4.36×10^{-4}	1.75×10^{-2}	3.46×10^{-4}	1.52×10^{-1}	1.33×10^{-1}	2.12×10^{-1}	1.86×10^{-1}
	10,10	9.11×10^{-3}	6.18×10^{-4}	1.74×10^{-2}	3.07×10^{-4}	2.91×10^{-1}	2.38×10^{-1}	1.87×10^{-1}	1.78×10^{-1}
	10,20	5.09×10^{-3}	7.72×10^{-5}	1.34×10^{-2}	1.74×10^{-4}	1.26×10^{-2}	1.60×10^{-3}	1.02×10^{-1}	1.30×10^{-1}
JY4	5,10	4.07×10^{-2}	1.05×10^{-3}	8.55×10^{-2}	1.57×10^{-3}	3.85×10^{-2}	5.97×10^{-3}	5.66×10^{-2}	2.03×10^{-3}
	10,10	2.52×10^{-2}	5.52×10^{-4}	5.36×10^{-2}	5.52×10^{-4}	5.25×10^{-2}	7.97×10^{-3}	3.72×10^{-2}	1.15×10^{-3}
	10,20	3.41×10^{-3}	6.64×10^{-5}	5.43×10^{-3}	6.43×10^{-5}	7.47×10^{-3}	1.01×10^{-3}	1.07×10^{-2}	4.64×10^{-4}
JY5	5,10	2.42×10^{-3}	5.83×10^{-6}	1.13×10^{-2}	2.38×10^{-4}	1.82×10^{-2}	9.69×10^{-3}	3.48×10^{-2}	6.36×10^{-4}
	10,10	2.42×10^{-3}	3.29×10^{-6}	1.10×10^{-2}	2.06×10^{-4}	2.11×10^{-2}	1.31×10^{-2}	2.46×10^{-2}	1.03×10^{-3}
	10,20	2.41×10^{-3}	7.04×10^{-6}	6.03×10^{-3}	5.48×10^{-5}	7.41×10^{-3}	2.17×10^{-3}	2.06×10^{-2}	1.52×10^{-4}
JY6	5,10	2.11	6.81×10^{-2}	5.90	1.65×10^{-1}	2.97	1.96×10^{-1}	3.35	1.18×10^{-1}
	10,10	1.61	6.26×10^{-2}	3.34	6.99×10^{-2}	3.23	3.76×10^{-1}	2.04	1.25×10^{-1}
	10,20	3.88×10^{-1}	1.34×10^{-2}	1.18	3.45×10^{-2}	7.97×10^{-1}	8.59×10^{-2}	5.36×10^{-1}	1.49×10^{-2}
JY7	5,10	1.42	1.26×10^{-1}	16.6	1.68	89.8	64.7	20.51	3.76
	10,10	2.06×10^{-1}	9.54×10^{-2}	4.33×10^{-1}	5.27×10^{-2}	83.9	67.6	8.38×10^{-1}	5.71×10^{-1}
	10,20	8.55×10^{-2}	4.93×10^{-3}	2.40×10^{-1}	2.10×10^{-2}	5.80	7.09	3.35×10^{-1}	8.29×10^{-2}
JY8	5,10	3.34×10^{-3}	4.25×10^{-5}	1.12×10^{-2}	2.12×10^{-4}	1.93×10^{-2}	1.38×10^{-2}	6.31×10^{-2}	3.93×10^{-3}
	10,10	3.17×10^{-3}	3.45×10^{-5}	1.13×10^{-2}	2.23×10^{-4}	2.86×10^{-2}	1.44×10^{-2}	6.34×10^{-2}	1.92×10^{-2}
	10,20	2.77×10^{-3}	2.06×10^{-5}	6.72×10^{-3}	1.19×10^{-4}	7.57×10^{-3}	1.93×10^{-3}	4.21×10^{-2}	1.47×10^{-3}

SCPS has strong searching ability for the eight dynamic test problems. Only on the test problem JY4 ($(n, \tau) = (5, 10)$), the PSS method is slightly better than the SCPS. For all the test problems, four algorithms have strong searching ability on JY1-JY5 as well as JY8. Their MIGDs reached the 10^{-2} magnitude. That is, all these four algorithms have good ability to track the dynamic front.

JY6 is a Type III dynamic problem as shown in [25], where the multimodal PS and PD change over time, and the distribution of the optimal solution would be also in a constant process of change. Although the SCPS would be better than the other three algorithms on this problem, its MIGD is unsatisfactory.

JY7 is also a multimodal problem, and its PF shape is in a constant process of change. As the number of local optimum is fixed, JY7 is relatively simpler than JY6. Thus, the four algorithms running in JY7 are better than that in JY6. According to the standard deviation of MIGD, SCPS has better stability than the other 3 algorithms and a higher solution.

In order to analyze the dynamic searching ability of the four algorithms, eight group benchmark problems were chosen, and under the condition that $(n, \tau) = (10, 20)$, the time-varying characteristic of IGD of algorithms was analyzed. Figure 3 shows the timely varying curves of IGD of the four algorithms running in the benchmark problem. Each time the environment changes, the smaller the IGD value is, and the more accurate the prediction of the algorithm.

It is indicated that the MOEA/D has a good searching performance in the static problem, but it could not dynamically predict. Thus, MOEA/D has an inadequate capability to solve the dynamic problem in bounded time. For the problems JY2 and JY5, PSS is better than MDP. However, for the problems JY7 and JY8, MDP is better than PSS. Especially, for JY7, large errors of the prediction might exist and keep increasing in PSS. In all eight benchmark problems, SCPS is better than the other three algorithms.

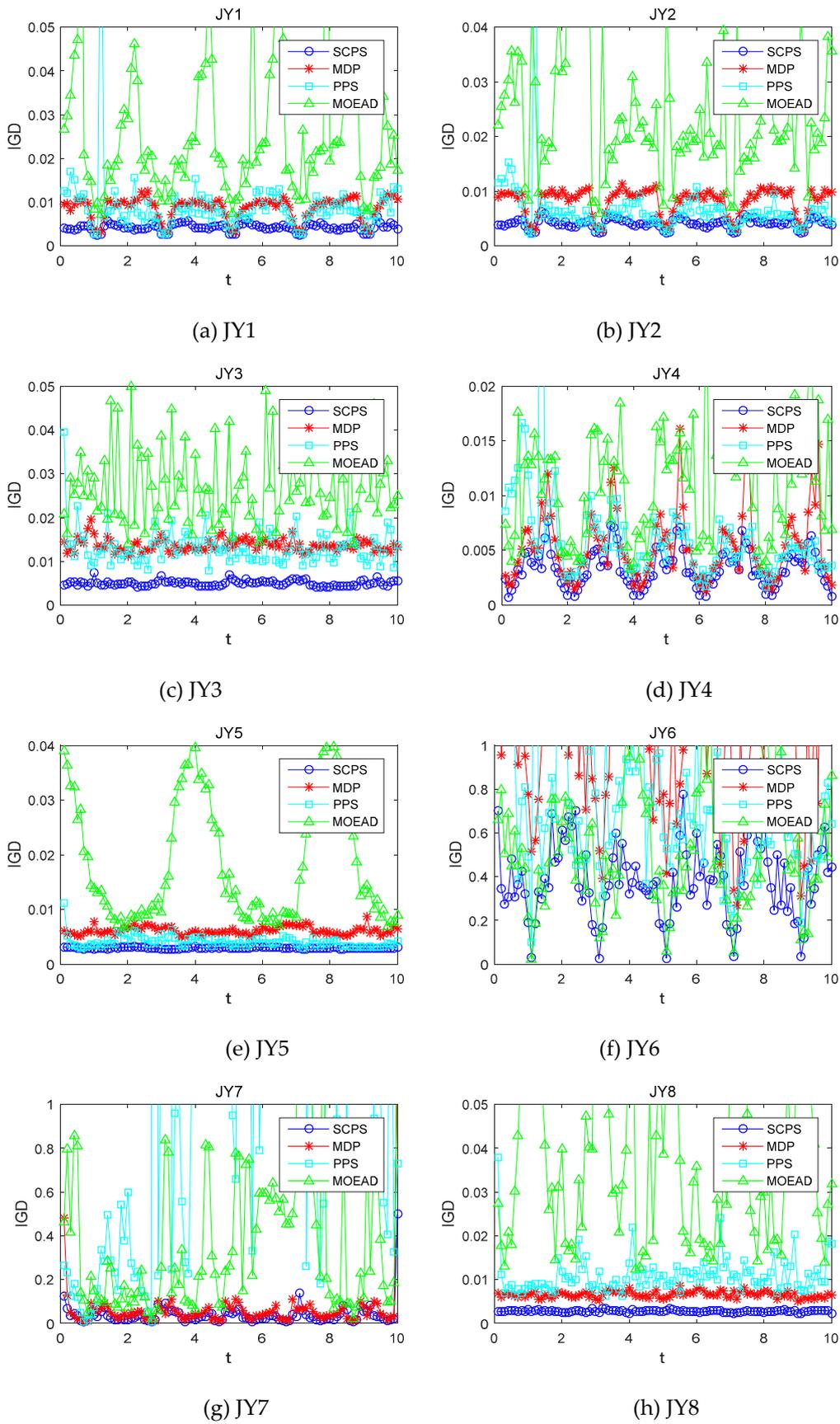


Figure 3. IGD indicator change of 4 algorithms running in benchmark problem.

In order to further analyze the experimental results, the distribution of predicted results at five moments was compared for JY4, JY5, JY7, and JY8. Figure 4 shows the distribution of predicted results at five moments for the four algorithms running in four benchmark problems.

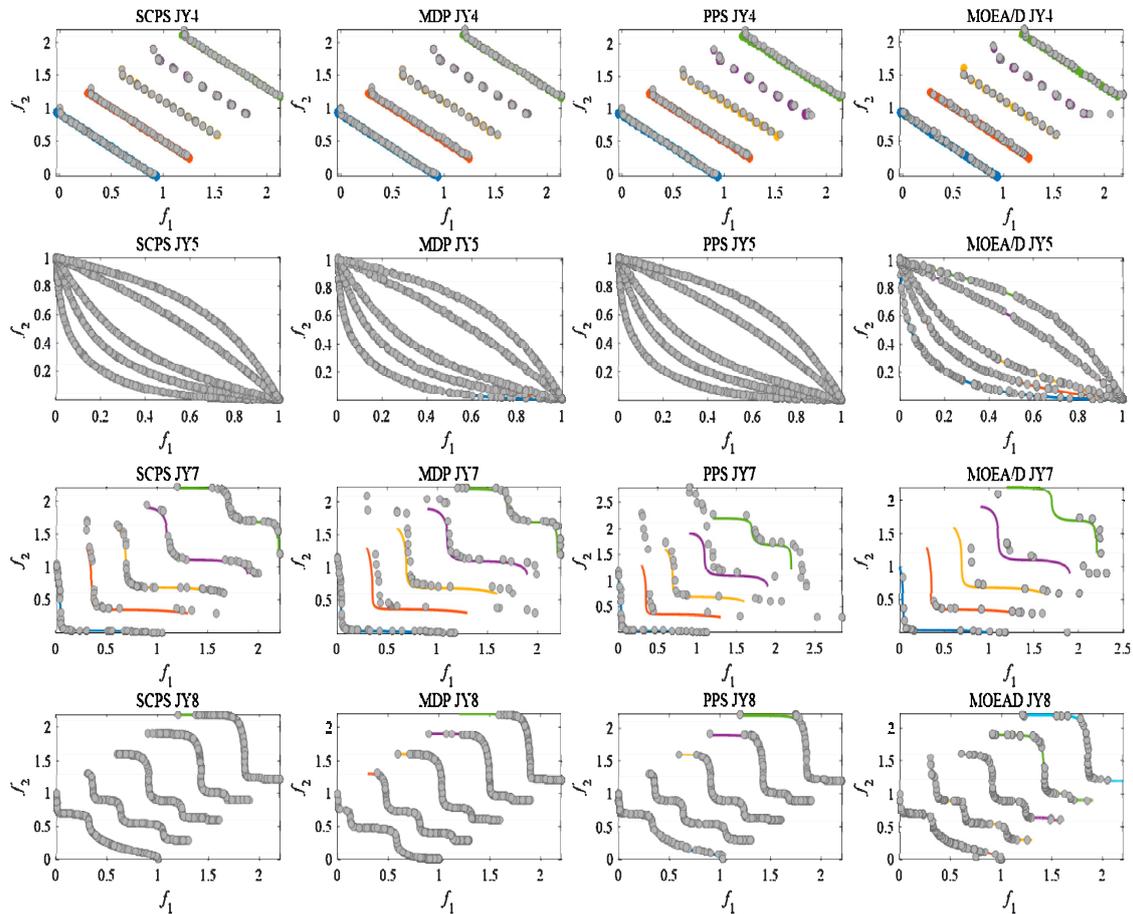


Figure 4. Scatter distribution of prediction population.

The discrete area of JY4 is time-varying. And the concavity of PF in JY5 changes over time. Although the PF changes of these two benchmark problems are different, they are both the unimodal problem. Thus, their results have good distribution and convergence. The PF shape of JY7 and JY8 would be changed all the time, so optimization is rather difficult. The points shown in Figure 4 are the non-dominant solution set of the population, and the number of these points reflects the performance of the algorithm. MDP and PPS have good distribution and convergence at the first moment, but the distribution of non-dominant solution set for these two algorithms is not satisfactory at the next four moments. That is, some points are far from the true PF. In comparison to the other three algorithms, SCPS performs particularly well in distribution and convergence. Moreover, in JY8, all four algorithms have good distribution, but SCPS performs better than the others in convergence.

According to the scatter distribution, SCPS has better searching performance. Its population segment provides an accurate shape change prediction. The cloud prediction strategy is the basic prediction strategy, and extra angle search is the main prediction angle error compensation.

Depending on the results of the IGD mean, IGD change with iterations, and scatter prediction results, the proposed algorithm is stable and has good predictive ability. Its prediction is more reasonable and closer to the population in the last moment.

5. Conclusions

The segment cloud prediction this paper proposed divides the population on the basis of the distance of PF. According to the results of center linear prediction for each population, the center position would be determined. Moreover, the population distribution is predicted by cloud theory. The angle error caused by the linear prediction process is decreased considerably by using extra angle search. The simulation results show that this algorithm has a better convergence and distribution, as well as environmental suitability. However, it cannot be denied that PF segmentation and cloud theory may not perform so well in the face of more difficult dynamic prediction problems, such as a mutation mapping relation between PF and PS. The high-dimensional dynamic multi-objective optimization problem is also an important research target, in which the effectiveness of Euclidean distance is also worth discussing. Future work can focus on these aspects.

Author Contributions: Conceptualization, P.N. and J.G.; methodology, Y.S.; writing—original draft preparation, W.Q.; writing—review and editing, Q.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 61703426) and by the Young Talent Fund of University Association for Science and Technology in Shaanxi, China (Grant No. 20190108).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xia, X.; Gui, L.; Zhan, Z. A multi-swarm particle swarm optimization algorithm based on dynamical topology and purposeful detecting. *Appl. Soft Comput.* **2018**, *67*, 126–140. [[CrossRef](#)]
2. Song, Y.; Fu, Q.; Wang, Y.; Wang, X. Divergence-based cross entropy and uncertainty measures of Atanassov's intuitionistic fuzzy sets with their application in decision making. *Appl. Soft Comput. J.* **2019**, *84*, 105703. [[CrossRef](#)]
3. Zhang, B.; Zhang, M.; Song, Y.; Zhang, L. Combing evidence sources in time domain with decision maker's preference on time sequence. *IEEE Access* **2019**, *7*, 174210–174218. [[CrossRef](#)]
4. Song, Y.; Wang, X.; Quan, W.; Huang, W. A new approach to construct similarity measure for intuitionistic fuzzy sets. *Soft Comput.* **2019**, *23*, 1985–1998. [[CrossRef](#)]
5. Song, Y.; Wang, X.; Zhu, J.; Lei, L. Sensor dynamic reliability evaluation based on evidence and intuitionistic fuzzy sets. *Appl. Intell.* **2018**, *48*, 3950–3962. [[CrossRef](#)]
6. Song, Y.; Wang, X.; Lei, L.; Xue, A. A novel similarity measure on intuitionistic fuzzy sets with its applications. *Appl. Intell.* **2015**, *42*, 252–261. [[CrossRef](#)]
7. Lei, L.; Song, Y.; Luo, X. A New Re-encoding ECOC Using a Reject Option. *Appl. Intell.* [[CrossRef](#)]
8. Soh, H.; Ong, Y.S.; Nguyen, Q.C.; Nguyen, Q.H.; Habibullah, M.S.; Hung, T.; Kuo, J.L. Discovering unique, low-energy pure water isomers: memetic exploration, optimization and landscape analysis. *IEEE Trans. Comput.* **2010**, *14*, 419–437. [[CrossRef](#)]
9. Thammawichai, M.; Kerrigan, E.C. Energy-efficient real-time scheduling for two-type heterogeneous multiprocessors. *Real Time Syst.* **2018**, *54*, 132–165. [[CrossRef](#)]
10. Shen, M.; Zhan, Z.H.; Chen, W.N.; Gong, Y.J.; Zhang, J.; Li, Y. Bi-velocity discrete particle swarm optimization and its application to multicast routing problem in communication networks. *IEEE Trans. Ind. Electron.* **2014**, *61*, 7141–7151. [[CrossRef](#)]
11. Deb, K.; Karthik, S. Dynamic multiobjective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling. In *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization, Matsushima, Japan, 5–8 March 2007*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 803–817.
12. Azevedo, C.R.B.; Araujo, A.F.R. Generalized immigration schemes for dynamic evolutionary multiobjective optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation, New Orleans, LA, USA, 5–8 June 2011*; pp. 2033–2040.
13. Shang, R.H.; Jiao, L.C.; Gong, M.G.; Ma, W.P. An immune clonal algorithm for dynamic multi-objective optimization. *J. Softw.* **2007**, *18*, 2700–2711. [[CrossRef](#)]

14. Kominami, M.; Hamagami, T. A new genetic algorithm with diploid chromosomes by using probability decoding for adaptation to various environments. *Electron. Commun. Jpn.* **2010**, *93*, 38–46. [[CrossRef](#)]
15. Yang, S. On the Design of Diploid Genetic Algorithms for Problem Optimization in Dynamic Environments. In Proceedings of the 2006 Congress on Evolutionary Computation, Vancouver, BC, Canada, 16–21 July 2006; pp. 1362–1369.
16. Liu, M.; Zeng, W.H. Memory enhanced dynamic multi-objective evolutionary algorithm based on decomposition. *J. Softw.* **2013**, *24*, 1571–1588. [[CrossRef](#)]
17. Hatzakis, I.; Wallace, D. Dynamic multi-objective optimization with evolutionary algorithms: A forward-looking approach. In Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, Washington, DC, USA, 8–12 July 2006; pp. 1201–1208.
18. Zhou, A.M.; Jin, Y.C.; Zhang, Q.F. A population prediction strategy for evolutionary dynamic multiobjective optimization. *IEEE Trans. Cybern.* **2014**, *44*, 40–53. [[CrossRef](#)]
19. Wu, Y.; Jin, Y.; Liu, X. A directed search strategy for evolutionary dynamic multiobjective optimization. *Soft Comput.* **2015**, *19*, 3221–3235. [[CrossRef](#)]
20. Rong, M.; Gong, D.; Zhang, Y.; Jin, Y.; Pedrycz, W. Multidirectional prediction approach for dynamic multiobjective optimization problems. In *Intelligent Computing Methodologies, ICIC 2016 Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2016; Volume 9773.
21. Li, Q.; Zou, J.; Yang, S.; Zheng, J.; Ruan, G. A predictive strategy based on special points for evolutionary dynamic multi-objective optimization. *Soft Comput.* **2018**, 1–17. [[CrossRef](#)]
22. Ruan, G.; Yu, G.; Zheng, J.; Zou, J.; Yang, S. The effect of diversity maintenance on prediction in dynamic multi-objective optimization. *Appl. Soft Comput.* **2017**, *56*, 631–647. [[CrossRef](#)]
23. Gee, S.B.; Tan, K.C.; Abbass, H.A. A benchmark test suite for dynamic evolutionary multiobjective optimization. *IEEE Trans. Cybern.* **2017**, *47*, 461–472. [[CrossRef](#)]
24. Deb, K.; Agrawal, S.; Pratap, A.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Comput.* **2002**, *6*, 182–197. [[CrossRef](#)]
25. Jiang, S.; Yang, S. Evolutionary dynamic multiobjective optimization: benchmarks and algorithm comparisons. *IEEE Trans. Cybern.* **2017**, *47*, 198–211. [[CrossRef](#)]
26. Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).