An Evolutionary Dynamic Multi-objective Optimization Algorithm Based on Center-point Prediction and Sub-population Autonomous Guidance

1st Jianwei Zhou
School of Information Engineering
Xiangtan University
Xiangtan 411105, China
Email: 1450376419@qq.com

2nd Juan Zou
School of Information Engineering
Xiangtan University
Xiangtan 411105, China
Email: zoujuan@xtu.edu.cn

3rd Shengxiang Yang
School of Information Engineering
Xiangtan University
Xiangtan 411105, China
School of Computer Science and Informatics
De Montfort University
Leicester LE1 9BH, U.K.
Email: syang@dmu.ac.uk

4th Gan Ruan
School of Information Engineering
Xiangtan University
Xiangtan 411105, China
Email: ruangan199332@gmail.com

5th Junwei Ou
School of Information Engineering
Xiangtan University
Xiangtan 411105, China
Email: 435727748@qq.com

6th Jinhua Zheng
School of Information Engineering
Xiangtan University
Xiangtan 411105, China
Email: jhzheng@xtu.edu.cn

***Abstract*—Dynamic multi-objective optimization problems (DMOPs) provide a challenge in that objectives conflict each other and change over time. In this paper, a hybrid approach based on prediction and autonomous guidance is proposed, which responds the environmental changes by generating a new population. According to the position of historical population, a part of the population is generated by predicting roughly and quickly. In addition, another part of the population is generated by autonomous guidance. A sub-population from current population evolves several generations independently, which guides the current population into the promising area. Compared with other three algorithms on a series of benchmark problems, the proposed algorithm is competitive in convergence and diversity. Empirical results indicate its superiority in dealing with dynamic environments.**

***Index Terms*—Dynamic multi-objective optimization, autonomous guidance**

I. Introduction

Many real-world optimization problems involve several objectives which conflict with each other and change over time [1]. This kind of optimization problems is usually known as dynamic multi-objective optimization problems (DMOPs) [1], [2]. Evolutionary algorithms are useful in solving DMOPs as they are inspired by evolution [3]. When dealing with DMOPs, the optimization algorithm must be able not only to evolve a set of convergent and diverse solution, but also to track the moving Pareto-optimal front (POF) and Pareto-optimal set (POS). Without loss of generality, a typical DMOP in this paper can be described as follows [1]:

$$\begin{cases} min \quad F(x,t) = (f_1(x,t), f_2(x,t), ..., f_M(x,t)) \\ s.t. \quad g(x,t) \leq 0, \quad h(x,t) = 0, \quad x \in \Omega_x \end{cases} \tag{1}$$

where f is the set of objectives at time t, M is the number of objectives, g and h are the inequality and equality constraints, and $\Omega_x \subseteq R^n$ is the decision space.

In recent years, evolutionary algorithms are widely introduced to solve the multi-objective optimization problem because of its high efficiency [4], [5], [6]. Based on this, evolutionary algorithms also have been introduced for DMOPs [7], [8], [9]. Jiang et al. [10] classified dynamic multi-objective evolutionary algorithms (DMOEAs) into two categories based on the way of handling environmental changes, namely, convergence-based approaches and diversity-based approaches.

Convergence-based approaches particular emphasize the convergence, which try to exploit past information for better tracking performance mainly by using prediction strategy and memory strategy. Some DMOEAs use prediction approaches to speed up convergence by guiding the population to evolve towards the POF after environmental changes. For example, Hatzakis and Wallace [11] proposed feed-forward prediction strategy (FPS) to estimate the POF. Zhou et al. [12] proposed a forecasting approach called population prediction strategy (PPS). More recently, in 2016, Muruganantham et al. [13] proposed a new prediction model called Kalman filter prediction. Those prediction models work at particular problem.

Usually, prediction models are too weak to deal with ruleless problem. Besides, memory scheme is another kind of common convergence-based approach [14], [15]. This kind of approach works by storing relevant information of historic POS into a memory. Goh et al. [16] proposed a coevolutionary multi-objective algorithm based on competitive and cooperative mechanisms to solve DMOPs. What's more, several memory-based strategies have been presented in Wang [17] which reuses the information from memory pool to conduct a new population while the changes happen.

Diversity-based approaches particular put emphasis on the diversity, which try to make the the population keep uniform distribution and expand the search area. Many methods based on diversity have been proposed [18], such as diversity introduction after a change, diversity maintenance throughout the run, and self-adaptive methods. Deb et al. [19] extends NSGA-II to deal with DMOPs by introducing diversity. This algorithm replaces $\zeta\%$ of the new population with new randomly produced solutions. In addition to randomly producing solutions, multiple population method is another normal diversity-based approach. Maintaining diversity by multiple population methods is helpful for exploring promising search area. For example, in R. Shang et al. [20] and C. Goh et al. [21], multiple population method is introduced to coevolutionary algorithm. Proper diversity is helpful for exploring promising search regions, but too much diversity may causes evolutionary stagnation. There are many other strategies to solving DMOPs [22], [23], [24], [7]. For example, Farina et al. [1] proposed an adapted static MOO algorithm for DMOPs.

In this paper, a hybrid approach based on center-point prediction and sub-population autonomous guidance(CPSAG) is proposed, which responds to environmental changes by generating a new population. According to the position of historical population, a part of the population is generated by predicting roughly and quickly. In addition, another part of the population is generated by autonomous guidance. A sub-population from current population evolves several generations independently, which guides the current population into the promising area.

The rest of this paper is organized as follows. Section II describes the proposed CPSAG in detail. Experimental studies and analysis on test problems are carried out in Section III. Finally, some conclusions are drawn in Section IV.

II. Algorithm Design

The flowchart of DMOEA framework is illustrated in algorithm 1. In general, the DMOEA starts with random initialization of population, and then executes optimization operators which include evaluation, selection, crossover, and mutation. If the stopping criteria are not met, the detection of environmental changes is executed all the time. The response will be executed only if the environment has changed, and the optimization operators are executed along with the detection of environmental changes.

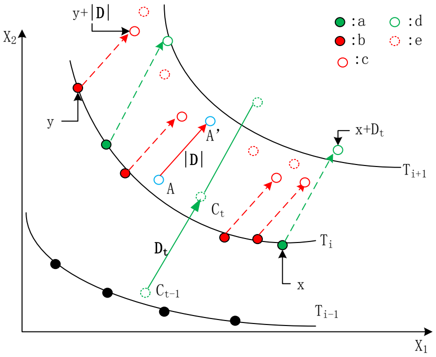

Fig. 1. Illustration of the proposed CPSAG.

Algorithm 1 Framework of DMOEA

```
input: N(population size)
output: $P(population)$
1: initpop();
2: optimize the MOP by using an MOEA
3: while stopping criterion not met do
4:     if change detected then
5:         response();
6:     else
7:         optimize the MOP by using an MOEA
8:     end if
9: end while
```

In the past, researchers have done a lot of work to solve DMOPs. Usually, a change is detected by reevaluating individuals or checking population statistical information [25], [26], [27]. We use the former way to detect the environmental changes. In fact, when the environment does not change, a DMOP is usually seen as a static multi-objective optimization problem. In this paper, RM-MEDA [28] is introduced as an MOEA to optimize the problems. In this paper, we mainly focus on the response to environmental changes of the **line 5** in algorithm 1.

In this section, we focus on how to respond to environmental changes by using center-point prediction and sub-population autonomous guidance strategy. It should be noted that we are not prepared to emphasize the accuracy of forecast when using the center-point prediction strategy. We are more inclined to track the trend of change because actually it is difficult to forecast a set of exact solution, and then, sub-population autonomous guidance strategy is introduced which helps to find the promising area independently. As Fig.1 shows, in the decision space, the sub-population labeled as 'd' is generated by center-point prediction strategy and another sub-population labeled as 'c' is generated by autonomous guidance strategy. It should be noted that the individuals are selected by non-dominated sorting in the Algorithm 2 and Algorithm 3. The following steps detail the algorithm.

A. Center-point Prediction

Because DMOP constantly changes all the time, it is reasonable to mine useful knowledge from the historical data. As shown in algorithm 2, we forecast the optimal solution

Algorithm 2 Center-point Prediction

```
input: $P_t$(population before environment changed at time t), $P_{t-1}$
output: $P_a(subpopulation)$
$C_t \leftarrow$ the center point of $P_t$
$C_{t-1} \leftarrow$ the center point of $P_{t-1}$
$M \leftarrow$ Q individuals selected from $P_t$
for $i = 1$ to $M.size()$ do
    $P_a \leftarrow X_i = M_i + C_t - C_{t-1}$
end for
```

at the next moment by using the the former two generations. Fig. 2 illustrates that, T_{i-1} and T_i represent the former two generations. C_t represents the centroid of POS at the time t and $|P|$ represents the size of population P , then C_t can be computed by

$$C_t = \frac{1}{|P|} \sum_{x \in P} x \tag{2}$$

What needs to be stressed again is that we do not intend intended to forecast the population accurately and overnight at the next moment, in fact, it is impossible to do it. As algorithm 2 shows, a part of the population is generated by predicting roughly and quickly, and this is based on the fact that $C_t - C_{t-1}$ represents the direction of evolution. And a new solution X' can be generated by

$$X' = X + D_t \tag{3}$$

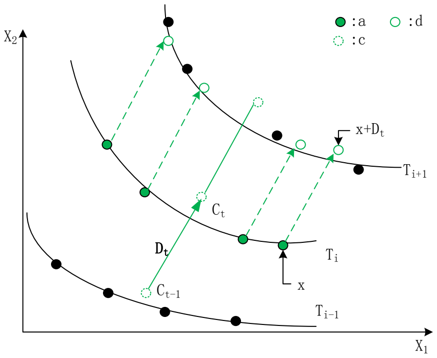

Fig. 2. Illustration of the center-point prediction.

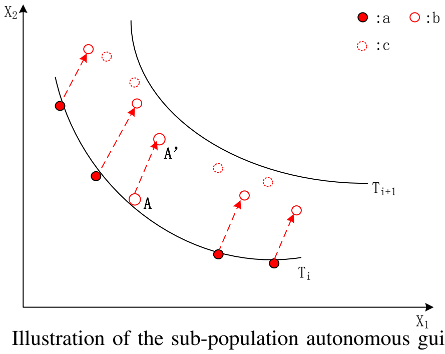

Fig. 3. Illustration of the sub-population autonomous guidance.

Algorithm 3 Autonomous Guidance

```
input: $P_t$(population before environment changed at time t)
output: $P_b, P_c, P_d(subpopulation)$
 1: $R_0 \leftarrow$ R individuals selected from $P_t$
 2: $C_0 \leftarrow$ the center point of $R_0$
 3: $R_1 \leftarrow$ execute EC operation k generations on $R_0$
 4: $C_1 \leftarrow$ the center point of $R_1$
 5: $N \leftarrow$ R individuals selected from $P_t$
 6: for $i = 1$ to $N.size()$ do
 7:     $P_b \leftarrow X_i = N_i + C_1 - C_0$
 8: end for
 9: $P_c \leftarrow R_0$
10: $P_d \leftarrow R_1$
```

B. Sub-population Autonomous Guidance

As mentioned above, just a part of the population is generated by predicting roughly and quickly. In order to compensate for this deficiency, a new and important strategy is proposed. As the algorithm 3 shows, after environmental changes being detected, some solutions have been selected to archive from the current population, in which center point is C_0, the center point is defined as (2). Then, the sub-population (consisting of solutions from the archive) evolves k generations and then gains a new sub-population where center point is labeled as C_1. In fact, C_1 - C_0 represents the direction of evolution by sub-population autonomous guidance but not prediction. Based on this, a part of the new population is generated according to Line 7. Because the sub-population R is evolving autonomously based on some solutions of current population, R will be retained to the next generation. It is also necessary to retain some individuals to the next generation.

Fig. 3 illustrates that, the environmental changes have been detected at time T_i. And, the population at time T_i is labeled as current population. The sub-population labeled 'a' is selected from current population by means of non-dominated sorting, in which center point is labeled as 'A'. The selected sub-population evolves k generations autonomously. Thereafter, we get a new sub-population labeled 'b' as shown in Fig. 2. The center point of this new sub-population is labeled as A'. This paper holds that the obtained sub-population is located in a promising area. So the evolutionary direction of sub-population can be used as the evolutionary direction of whole population.

C. Boundary Check

The boundary check in the proposed algorithm is applied to check whether the generated solution is within the given boundary of the decision space. The procedure of the boundary check is described in detail as follows:

$$y_i = \begin{cases} x_i & if \quad l_i < x_i < u_i \\ u_i - 0.1 * (u_i - C_i) & if \quad x > u_i \\ l_i + 0.1 * (C_i - l_i) & if \quad x < l_i \end{cases} \tag{4}$$

where C_i is the ith dimension of the population' center point, i = 1,..., n, n is the dimension of the test problems' decision space. l_i is the minimum boundary of the ith dimension, correspondingly, u_i is the maximum boundary of the ith dimension.

III. EXPERIMENTAL STUDIES

A. Test Problems

In this paper, three test benchmark suites, including FDA [1], dMOP [16], and F [12], are used to assess proposed algorithm. According to the changes of PSt and PFt, the benchmark problems are classified into four categories. Type I: the PS changes and the PF remains invariant. Type II: both PS and PF changes. Type III: the PF changes and ps remains invariant. Type IV: both PS and PF remain invariant.Usually, the fourth type is considered to be static MO optimization problem, other than dynamic problem.

TABLE I
THE POPULATION SIZE

Type	benchmark problems
I	FDA1, FDA4, DMOP3
II	FDA3, DMOP2, F5, F6, F7, F8, F9, F10
III	DMOP1, FDA2

B. Performance Metric

- Inverted generational distance (IGD)
  The IGD metric proposed by Sierra and Coello [29] measures both the convergence and diversity of the obtained solutions. It is mathematically given by

$$IGD(A, P^*) = \frac{\sum_{v \in p^*} d(v, A)}{|P^*|} \tag{5}$$

  Where A is the obtained PF, P_* is a set of uniformly distributed points along the optimal PF in the objective space and $d(v, A)$ is the minimum Euclidean distance between v and the points in A. The lower the IGD is, the better the convergence performance is shown.

- Maximum spread (MS)
  The MS metric proposed by Goh [30] measures to what extent the obtain POF covers the optimal POF, and is defined as follows:

$$MS(A, P^*) = \sqrt{\frac{\sum_{j=1}^{M} (\frac{min(A_{j,u}, P^*_{j,u}) - max(A_{j,u}, P^*_{j,l})}{P^*_{j,u} - P^*_{j,l}})^2}{M}} \tag{6}$$

  where $A_{j,u}$ and $A_{j,l}$ are the maximum and the minimum value of the j-th objective in the obtained PF. $P^*_{j,u}$ and $P^*_{j,l}$ are the maximum and the minimum value of the j-th objective in the optimal PF.

- Spacing metric (Sp)
  Schott [31] developed this kind of metric with regard to the distribution of the discovered Pareto front. Schott's spacing metric measures how evenly the members in POF are distributed, and is computed as

$$Sp = \sqrt{\frac{1}{|A| - 1} \sum_{i=1}^{|A|} (\overline{d} - d_i)^2} \tag{7}$$

  where A is the approximate Pareto optimal front and d_i is the Euclidean distance between the ith member in POF and its nearest member in POF and $\overline{d}$ is the average value of d_i.

C. Compared Algorithms and Parameter Settings

- Compared Algorithms
  In this paper, the proposed CPSAG is compared to three other existing strategies: population prediction strategy (PPS), feed-forward prediction strategy (FPS) and randomly initialize strategy (RIS). RM-MEDA introduced in this paper is chosen as the MOEA optimizer.

  FPS: FPS [11] records the boundary points of the POF and predicts the location of the whole population when the next environmental change occurs by autoregressive (AR) model. The re-initialized population in FPS is composed of three parts: the non-dominated solution set, the dominated solution set and the predicted solution set.

  PPS: The optimal solution set of PPS [12] is divided into two parts: the population center and manifold. Based on the archived population centers over a number of continuous time series, PPS adopts a univariate autoregression (AR) model to predict the next population center. Similarly, previous manifolds are used to predict the next manifold. The initial population is completely generated through the predicted center and manifold when there is an environmental change.

  RIS: Randomly initialize strategy (RIS) [12], which randomly generates a new population in the feasible region of the decision space.

- Parameter Settings
  The parameter settings for the problems and algorithms are as follows. The severity and frequency of the environmental changes are $n_T = 10$, $\tau_T = 30$. The dimensions of the test problems' decision space are all set to be n = 20. The AR(p) model order is set to be p = 3 and the length of history mean point series is set to be M = 23 for PPS and FPS. For FPS, the number of cluster was set 5 and the probability in the prediction model was 0.9. The population size is set to be N = 100 for all test problems. To make sure the population size is 100 in every generation, generally, the sub-population sizes of two strategy are set Q =40, R = 20. We run each algorithm 20 times for each test problem independently. Then each

TABLE II
MEAN AND STANDARD DEVIATION OF IGD VALUES FOR FOUR ALGORITHMS ON TEST PROBLEMS OVER 20 RUNS

PROBLEM	CPSAG	PPS	FPS	RIS
FDA1	**9.40E-03(8.14E-04)**	5.82E-02(2.59E-02)	2.64E-02(4.04E-03)	1.13E+00(4.92E-02)
FDA2	2.56E-02(1.04E-04)	2.56E-02(1.29E-04)	**2.55E-02(8.06E-05)**	5.25E-02(5.99E-04)
FDA3	**2.58E-01(9.16E-03)**	3.77E-01(3.39E-02)	2.83E-01(5.98E-03)	1.79E+00(7.08E-02)
FDA4	**1.09E-01(1.96E-03)**	1.19E-01(2.53E-03)	1.26E-01(2.46E-03)	3.67E-01(8.16E-03)
DMOP1	**9.80E-02(3.62E-03)**	1.54E-01(1.34E-01)	1.01E-01(4.67E-03)	5.62E-01(1.79E-02)
DMOP2	**9.50E-02(1.31E-03)**	1.65E-01(5.49E-02)	1.18E-01(9.53E-03)	1.50E+00(2.83E-02)
DMOP3	**9.32E-03(7.30E-04)**	4.63E-02(2.69E-02)	2.60E-02(4.45E-03)	1.12E+00(2.77E-02)
F5	**1.97E-01(2.75E-02)**	2.13E-01(5.01E-02)	2.54E-01(4.09E-02)	1.04E+00(3.41E-02)
F6	**2.11E-01(1.24E-02)**	2.45E-01(2.05E-02)	2.63E-01(2.33E-02)	4.96E-01(1.25E-02)
F7	**1.92E-01(4.18E-03)**	1.97E-01(9.35E-03)	2.04E-01(7.04E-03)	5.65E-01(1.38E-02)
F8	3.09E-01(9.32E-02)	**2.72E-01(7.88E-02)**	2.97E-01(5.99E-02)	1.06E+00(2.69E-02)
F9	8.57E-01(4.40E-01)	3.67E-01(5.04E-02)	**3.46E-01(2.56E-02)**	9.36E-01(2.73E-02)
F10	**1.32E-01(3.17E-03)**	1.41E-01(5.97E-03)	1.39E-01(3.63E-03)	7.01E-01(2.34E-02)

simulation experiment run for 3600 generations and 120 environmental changes were tracked for all algorithms.

D. Results and discussion

1) Results on FDA and DMOP problems: Comparison results of CPSAG with other three DEMOAs in terms of IGD, MS and Sp values are presented in Tables II-IV . The best metric values are highlighted in bold face. As the results show, CPSAG performs better than other three DEMOAs on most problems for the metric values.

It is clear from Table II that CPSAG obtains the best results on the majority of the tested FDA and dMOP instances. Apart from FDA2, CPSAG performs better than PPS, FPS and RIS on the other tested FDA and dMOP instances for IGD values, which indicates that CPSAG maintains better convergence over the three others compared algorithms in most case. It is worth noting that CPSAG significantly performs better than the other three algorithms on the tested dMOP instances.

As shown in Table III, CPSAG performs better than FPS and RIS on all FDAs and DMOPs in terms of MS. Apart from FDA2, CPSAG also performs better than PPS on the other tested FDA and dMOP instances for MS value.

From Table IV, CPSAG performs better than the other algorithms on FDA1, FDA2, DMOP1 and DMOP3 in terms of SP. but CPSAG is a little worse than FPS on FDA3. Compared to pps, CPSAG shows worse performance.

2) Results on F5-F10 problems: In this section, We test CPSAG and the other three algorithms on F5-F10, which have nonlinear correlation between decision variables. It can be observed from Table II that CPSAG obtains the best results ont the majority of the tested F instances and mainly loses on F8 and F9, which also indicates that CPSAG maintains better convergence in most case. Table III presents the MS metric obtained by four algorithms on the tested F instances. Apart from F9, CPSAG performs better than PPS on the other tested F instances for MS value. And CPSAG performs better than FPS on provided problems except F8. It is clear from Table IV that CPSAG significantly performs better than the other three algorithms on F5, F6, F7, F9 and F10 but just lose on F8.

3) Discussion : The previous experimental comparison has shown that CPSAG is capable of solving a wide range of DMOPs. On the one hand, in some patterns of changes, the expected set of solutions can be found easily. So, center-point prediction plays an important role in this situation. On the other hand, the autonomous guidance strategy can overcome shortages of the former strategy in other cases. Because the autonomous guidance strategy finds the expected solution by autonomic evolution regardless of the fact that some environment changes irregularly. The autonomous guidance strategy has some drawbacks too. For example, CPSAG performs worse on F7 and F8. One of the main reasons is the lack of search capability. Further, the individuals that participate in autonomic evolution are not enough.

In order to investigate the influence of the generation in autonomous evolution, several comparative experiments are carried out. The experiments are conducted on FDA1-FDA5, dMOP1-dMOP3, and F5-F10. The relevant parameter settings are presented in subsection C. The generations of autonomous evolution are set to 5, 10, 20 and 30 respectively. From table V, when k$\geq$10, The indicators are not very different from each other for most case. What needs to be explained is the k is set to 10 when CPSAG compared with other three algorithms.

IV. CONCLUSION

This paper has proposed the CPSAG algorithm for dynamic multi-objective optimization problems. CPSAG aims to respond to environmental changes. To attain this goal, the center-point prediction and autonomous guidance strategy are introduced into the proposed method. From comparison results, the proposed algorithm is competitive in almost test problems. In the future, it is a subject remaining to be researched that the influence of the number of sub-population.

ACKNOWLEDGMENT

This work was supported by the research projects: the National Natural Science Foundation of China under Grant Nos. 61502408, 61673331, 61379062 and 61403326, the Education Department Major Project of Hunan Province under Grant No. 17A212, the CERNET Innovation Project under Grant No. NGII20150302, the Natural Science Foundation

TABLE III
MEAN AND STANDARD DEVIATION OF MS VALUES FOR FOUR ALGORITHMS ON TEST PROBLEMS OVER 20 RUNS

PROBLEM	CPSAG	PPS	FPS	RIS
FDA1	**9.95E-01(7.73E-04)**	9.78E-01(6.00E-03)	9.85E-01(2.48E-03)	7.03E-01(5.20E-03)
FDA2	7.61E-01(8.32E-04)	**7.62E-01(9.98E-04)**	7.61E-01(6.63E-04)	8.75E-01(1.83E-03)
FDA3	**8.71E-01(6.88E-03)**	7.32E-01(1.74E-02)	8.05E-01(5.53E-03)	5.77E-01(7.21E-03)
FDA4	**0.99999(2.12E-06)**	0.99986(1.65E-05)	0.99998(5.14E-06)	0.99984(1.35E-05)
DMOP1	**9.91E-01(4.42E-03)**	9.66E-01(5.04E-02)	9.87E-01(5.96E-03)	3.19E-01(1.24E-02)
DMOP2	**9.90E-01(1.96E-03)**	9.54E-01(2.90E-02)	9.71E-01(7.06E-03)	5.43E-01(1.24E-02)
DMOP3	**9.95E-01(7.10E-04)**	9.80E-01(8.94E-03)	9.85E-01(2.64E-03)	7.03E-01(7.25E-03)
F5	**9.76E-01(1.39E-02)**	9.75E-01(1.45E-02)	9.59E-01(1.43E-02)	6.79E-01(1.67E-02)
F6	**9.68E-01(1.20E-02)**	9.66E-01(8.30E-03)	9.50E-01(1.27E-02)	7.98E-01(1.09E-02)
F7	**9.86E-01(5.08E-03)**	9.81E-01(6.18E-03)	9.75E-01(8.45E-03)	7.82E-01(1.00E-02)
F8	9.10E-01(3.63E-02)	9.37E-01(2.53E-02)	**9.45E-01(1.93E-02)**	6.79E-01(1.19E-02)
F9	7.07E-01(9.32E-02)	**8.72E-01(1.65E-02)**	8.26E-01(2.02E-02)	6.96E-01(1.82E-02)
F10	**0.999997**(3.10E-06)	0.999754(4.94E-04)	0.999994(2.78E-06)	0.99958(6.77E-05)

TABLE IV
MEAN AND STANDARD DEVIATION OF SP VALUES FOR FOUR ALGORITHMS ON TEST PROBLEMS OVER 20 RUNS

PROBLEM	CPSAG	PPS	FPS	RIS
FDA1	**4.46E-03(5.59E-04)**	5.34E-03(5.72E-04)	5.35E-03(3.61E-04)	4.22E-02(3.31E-03)
FDA2	**4.88E-03(9.44E-05)**	4.91E-03(9.90E-05)	5.07E-03(1.31E-04)	1.02E-02(3.63E-04)
FDA3	2.12E-02(1.29E-03)	7.16E-03(5.22E-04)	**6.40E-03(1.60E-04)**	4.02E-02(3.99E-03)
FDA4	7.20E-02(7.02E-03)	**4.92E-02(1.25E-03)**	7.30E-02(5.32E-03)	1.22E-01(3.17E-03)
DMOP1	**3.35E-03(2.95E-04)**	3.51E-03(6.30E-04)	4.05E-03(2.03E-03)	4.29E-02(4.94E-03)
DMOP2	6.09E-03(7.01E-04)	**4.55E-03(4.21E-04)**	5.46E-03(3.74E-04)	3.11E-02(2.26E-03)
DMOP3	**4.71E-03(4.80E-04)**	5.18E-03(5.79E-04)	5.45E-03(2.84E-04)	4.10E-02(2.91E-03)
F5	**6.74E-03(3.31E-03)**	1.22E-02(7.93E-03)	3.27E-02(1.22E-02)	9.38E-02(7.38E-03)
F6	**7.63E-03(2.68E-03)**	1.45E-02(3.53E-03)	2.69E-02(6.45E-03)	6.44E-02(8.53E-03)
F7	**6.10E-03(1.60E-03)**	8.02E-03(1.93E-03)	1.82E-02(6.07E-03)	6.87E-02(6.71E-03)
F8	2.57E-02(1.24E-02)	**2.42E-02(8.85E-03)**	4.22E-02(1.43E-02)	9.77E-02(6.51E-03)
F9	**3.79E-02(1.53E-02)**	3.83E-02(5.39E-03)	4.50E-02(7.16E-03)	9.31E-02(8.02E-03)
F10	**9.03E-02(1.01E-02)**	1.13E-01(1.41E-02)	1.04E-01(1.31E-02)	5.82E-01(3.73E-02)

TABLE V
MEAN OF THE IGD, SP AND MS VALUES FOR THE CPSAG WITH VARYING EVOLUTIONARY GENERATIONS K, WHERE THE AVERAGE OVER 20 INDEPENDENT RUNS IS SHOWN

Problems	IGD				Sp				MS			
	k=5	k=10	k=20	k=30	k=5	k=10	k=20	k=30	k=5	k=10	k=20	k=30
FDA1	1.08E-02	9.40E-03	9.46E-03	9.13E-03	5.10E-03	4.46E-03	4.59E-03	4.70E-03	9.94E-01	9.95E-01	9.95E-01	9.96E-01
FDA2	2.56E-02	2.56E-02	2.56E-02	2.56E-02	4.93E-03	4.88E-03	4.87E-03	4.83E-03	7.61E-01	7.61E-01	7.61E-01	7.61E-01
FDA3	2.64E-01	2.58E-01	2.55E-01	2.49E-01	2.14E-02	2.12E-02	2.09E-02	2.16E-02	8.66E-01	8.71E-01	8.72E-01	8.77E-01
FDA4	1.12E-01	1.09E-01	1.09E-01	1.09E-01	7.65E-02	7.20E-02	7.15E-02	6.71E-02	1.00E+00	1.00E+00	1.00E+00	1.00E+00
DMOP1	1.00E-01	9.80E-02	9.94E-02	1.05E-01	3.40E-03	3.35E-03	3.44E-03	3.54E-03	9.87E-01	9.91E-01	9.88E-01	9.80E-01
DMOP2	1.10E-01	9.50E-02	9.58E-02	9.49E-02	6.45E-03	6.09E-03	6.17E-03	6.06E-03	9.80E-01	9.90E-01	9.89E-01	9.91E-01
DMOP3	1.05E-02	9.32E-03	9.51E-03	9.15E-03	4.77E-03	4.71E-03	4.71E-03	4.69E-03	9.95E-01	9.95E-01	9.95E-01	9.95E-01
F5	2.25E-01	1.97E-01	2.16E-01	2.01E-01	1.01E-02	6.74E-03	1.07E-02	8.78E-03	9.60E-01	9.76E-01	9.68E-01	9.77E-01
F6	2.15E-01	2.11E-01	2.09E-01	2.13E-01	9.56E-03	7.63E-03	8.14E-03	8.56E-03	9.65E-01	9.68E-01	9.73E-01	9.72E-01
F7	1.95E-01	1.92E-01	1.95E-01	1.90E-01	5.65E-03	6.10E-03	6.13E-03	5.57E-03	9.80E-01	9.86E-01	9.83E-01	9.86E-01
F8	3.04E-01	3.09E-01	3.43E-01	3.51E-01	2.77E-02	2.57E-02	2.63E-02	3.17E-02	9.17E-01	9.10E-01	9.10E-01	9.01E-01
F9	8.76E-01	8.57E-01	9.77E-01	9.36E-01	3.35E-02	3.79E-02	4.01E-02	3.92E-02	7.07E-01	7.07E-01	6.78E-01	7.09E-01
F10	1.35E-01	1.32E-01	1.31E-01	1.30E-01	1.01E-01	9.03E-02	8.93E-02	8.32E-02	1.00E+00	1.00E+00	1.00E+00	1.00E+00

of Hunan Province under Grant No. 14JJ2072, the Science and Technology Plan Project of Hunan Province under Grant No. 2016TP1020, the Provinces and Cities Joint Foundation Project under Grant No. 2017JJ4001.

REFERENCES

[1] K. D. M. Farina, P. Amato, Dynamic Multiobjective Optimization Problems: Test Cases, Approximation, and Applications., IEEE Press, 2004.

[2] C. K. Goh, K. C. Tan, Evolutionary Multi-objective Optimization in Uncertain Environments - Issues and Algorithms, Springer Berlin Heidelberg, 2009.

[3] M. Camara, J. Ortega, F. J. Toro, Parallel processing for multi-objective optimization in dynamic environments, Parallel and Distributed Processing Symposium (2007) 1–8.

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

[5] Q. Zhang, H. Li, Moea/d: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731.

[6] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints, IEEE Transactions on Evolutionary

Computation 18 (4) (2014) 577–601.

[7] J. Wei, Y. Wang, Hyper rectangle search based particle swarm algorithm for dynamic constrained multi-objective optimization problems, in: Evolutionary Computation, 2012, pp. 1–8.

[8] A. Birashk, J. K. Kordestani, M. R. Meybodi, Cellular teaching-learning-based optimization approach for dynamic multi-objective problems, Knowledge-Based Systems 141.

[9] M. Mavrovouniotis, C. Li, S. Yang, A survey of swarm intelligence for dynamic optimization: algorithms and applications, Swarm Evolutionary Computation 33 (2017) 1–17.

[10] S. Jiang, S. Yang, A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization, IEEE Transactions on Evolutionary Computation PP (99) (2016) 1–1.

[11] I. Hatzakis, D. Wallace, Dynamic multi-objective optimization with evolutionary algorithms:a forward-looking approach, in: Conference on Genetic and Evolutionary Computation, 2006, pp. 1201–1208.

[12] A. Zhou, Y. Jin, Q. Zhang, A population prediction strategy for evolutionary dynamic multiobjective optimization, IEEE Transactions on Cybernetics 44 (1) (2013) 40–53.

[13] A. Muruganantham, K. C. Tan, P. Vadakkepat, Evolutionary dynamic multiobjective optimization via kalman filter prediction, IEEE Trans Cybern 46 (12) (2016) 2862.

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

[15] M. Liu, Memory enhanced dynamic multi-objective evolutionary algorithm based on decomposition, Journal of Software 24 (7) (2013) 1571–1588.

[16] C. K. Goh, K. C. Tan, A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization, IEEE Press, 2009.

[17] Y. Wang, B. Li, Investigation of memory-based multi-objective optimization evolutionary algorithm in dynamic environment, in: Eleventh Conference on Congress on Evolutionary Computation, 2009, pp. 630–637.

[18] R. Gan, G. Yu, J. Zheng, J. Zou, S. Yang, The effect of diversity maintenance on prediction in dynamic multi-objective optimization, Applied Soft Computing.

[19] K. Deb, B. R. N. Udaya, S. Karthik, Dynamic Multi-objective Optimization and Decision-Making Using Modified NSGA-II: A Case Study on Hydro-thermal Power Scheduling, DBLP, 2007.

[20] R. Shang, Y. Wang, J. Wang, L. Jiao, S. Wang, L. Qi, A multi-population cooperative coevolutionary algorithm for multi-objective capacitated arc routing problem, Information Sciences 277 (2) (2014) 609–642.

[21] N. Al-Messabi, Y. Li, I. El-Amin, C. Goh, Forecasting of photovoltaic power yield using dynamic neural networks, International Joint Conference on Neural Networks 20 (2012) 1–5.

[22] A. Dłaz-Manrłquez, G. T. Pulido, J. G. Ramłrez-Torres, Handling dynamic multiobjective problems with particle swarm optimization, in: Icaart 2010 - Proceedings of the International Conference on Agents and Artificial Intelligence, Volume 1 - Artificial Intelligence, Valencia, Spain, January, 2010, pp. 337–342.

[23] C. R. B. Azevedo, A. F. R. Arajo, Generalized immigration schemes for dynamic evolutionary multiobjective optimization, in: IEEE Congress on Evolutionary Computation, Cec 2011, New Orleans, La, Usa, 5-8 June, 2011, pp. 2033–2040.

[24] M. Helbig, Solving dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation, Management of Boundary Constraint Violations.

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

[26] J. G. Shouyong Jiang, Marcus Kaiser, N. Krasnogor, Less detectable environmental changes in dynamic multiobjective optimisation, Conference:GECCO2018.

[27] S. Jiang, S. Yang, A framework of scalable dynamic test problems for dynamic multi-objective optimization, in: IEEE Symposium on Computational Intelligence in Dynamic Uncertain Environments, 2014, pp. 32–39.

[28] Q. Zhang, A. Zhou, Y. Jin, RM-MEDA: A Regularity Model-Based Multiobjective Estimation of Distribution Algorithm, IEEE Press, 2008.

[29] C. A. C. Coello, M. R. Sierra, A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm, in: Mexican International Conference on Artificial Intelligence, 2004, pp. 688–697.

[30] C. K. Goh, K. C. Tan, An investigation on noisy environments in evolutionary multiobjective optimization, IEEE Transactions on Evolutionary Computation 11 (3) (2007) 354–381.

[31] J. R. Schott, Fault tolerant design using single and multicriteria genetic algorithm optimization, Cellular Immunology 37 (1) (1995) 1C13.