Multi-objective Optimization Based on Chaotic Particle Swarm Optimization

Liansong Xu and Dazhi Pan

***Abstract*—The multi-objective problem is particularly difficult in practical engineering applications, so more and more scholars have studied the problem to find the true Pareto optimal solution. In order to improve the convergence performance of multi-objective optimization algorithm and diversity, this paper proposes a multi-objective optimization algorithm based on chaos particle swarm optimization algorithm: using Logistic mapping sequences in solution in the particle swarm algorithm is updated; introducing the crossover operator of normal distribution to improve the diversity of the population; and using simplified mesh reduction and gene exchange to improve the performance of the algorithm. Compared with the MOPSO, NSGA-II and MOEA/D algorithms, it is shown that the proposed algorithm has good performance and can effectively solve the multi-objective optimization problem.**

***Index Terms*—Multi-objective optimization, logistic mapping, C-MOPSO, crossover operator, Pareto optimal.**

I. Introduction

Particle swarm optimization (PSO) is an intelligent optimization method for simulating group behavior, which is proposed by Kennedy *et al.* in 1995 [1-2]. Its core idea is to promote the development of the whole population through the information sharing mechanism between particles and experience from each other. Particle swarm optimization (PSO) algorithm has been applied in many single objective optimization problems because of its fast convergence and easy implementation, and it has good results. In recent years, many scholars began to study particle swarm optimization (PSO) algorithm to solve multi-objective optimization problems. The representative is: Li Fei *et al*. [3] proposed a multi-objective particle swarm optimization (MOEA/D-DE) algorithm based on decomposition and differential evolution. In the existing multi-objective particle swarm optimization algorithm, we did not take full advantage of the information obtained in the calculation process. In every iteration, we only use Individual optimal and global optimal; At the same time, In the process of optimization, we only exchange information between particles, but the information exchange between non dominated solutions stored in external archives is not fully utilized. Therefore, it is not possible to approach the real Pareto front leading to poor optimization performance.

In scientific research, many engineering application problems can be classified as multi-objective optimization problems, and multiple objectives need to be optimized at the same time. Therefore, the demand for solving multi-objective problems is stronger. However, the multiple goals that need to be optimized are often conflicting, and one optimization of a target will reduce the performance of other targets. The traditional multi-objective optimization method converts multi-objective optimization problems into single objective optimization problems by weighting, but this requires prior knowledge of the problem to be optimized, and whether it can be divided according to these objectives, so this method is difficult to deal with some multi-objective optimization problems [4], [5]. The population based evolutionary algorithm can be used to search multiple solutions in the solution space in parallel, and it is suitable for solving the multi-objective optimization problem [6]. In recent years, many scholars have proposed different improved particle swarm algorithms to solve multi-objective optimization problems. For example, Yang Ning *et al.* [7] put forward a multi-objective particle swarm optimization algorithm based on the multilevel interaction of information, this algorithm can divide the whole optimization process into 3 levels: standard particle swarm optimization layer, particle evolution and learning layer, and file information exchange layer. The 3 levels together improve the convergence and diversity of the algorithm. Li Kewen *et al.* [8] proposed a detection mechanism for the evolutionary state of Multi-objective particle swarm optimization (PSO): Through updating the external Pareto solution set, we detect the evolution state of the algorithm and get feedback information to dynamically adjust the evolution strategy, so that the algorithm can take account of the diversity and convergence of the approximate Pareto front-end in the process of evolution; Yang Jingming *et al.* [9] introduced the redundancy mechanism of external files, and enhanced the diversity of the solution by using its variation and the interference strategy of the population so as to avoid the premature phenomenon of the algorithm. Li Li and Wang Wanliang *et al.* [10] proposed a multi-objective particle swarm optimization (PSO) based on grid ranking, the grid-based ranking mechanism combines the individual dominance information in the entire solution space, and takes advantage of this information to sort. As a result, we gain the merits of the relationship between individuals in the population effectively and efficiently, so the distribution of the solution has been improved well; Yang Jingming *et al.* [11] proposed multi-objective adaptive chaos particle swarm optimization algorithm, applied chaotic sequence to our global optimal selection, and added adaptive selection strategy, and the effect of improvement is also very good.

In order to improve the multi-objective particle swarm

Manuscript received February 8, 2018; revised April 19, 2018.

Liansong Xu is with the School of Mathematics and Information, West Normal University, China (e-mail: 654261644@ qq.com).

Dazhi Pan was with the School of Mathematics and Information, West Normal University, China(e-mail: pdzzj@126.com).

optimization algorithm convergence and diversity, according to the ergodicity of chaotic sequences proposed by Xiong Junhua and Xie Fei [12], a multi-objective optimization algorithm based on Chaotic Particle Swarm Optimization (C-MOPSO) is proposed. Based on particle swarm optimization, the algorithm uses chaotic Logistic sequence to generate initial solution, and adds normal crossover operator into the corresponding particle operation, and makes full use of particle information obtained from each iteration. At the same time, the file reduction strategy is added to reduce the computational complexity. Finally, a comparative analysis with MOEA/D, NSGA-II and MOPSO algorithm shows that the proposed algorithm has good performance and can effectively solve multi-objective optimization problems.

II. Basic Concepts

A. Description of multi-Objective Problem

The multi-objective optimization problem is composed of *D* decision variables, *N* objective functions and *m+n* constraints. The decision variables are function relations with objective functions and constraints. In non-inferior solutions, the decision maker can only choose a non-inferior solution which makes it satisfactory as the final solution according to the requirement of the specific problem. The mathematical form of the multi-objective optimization problem can be described as follows:

$$\min f(x)=[f_1(x), f_2(x),..., f_n(x)], n=1,2,...N \tag{1}$$

$$.st.g_i(x)\le 0, i=1,2,...,m \tag{2}$$

$$h_j(x)=0, j=1,2,...,k \tag{3}$$

$$x=[x_1,x_2,...,x_d] \tag{4}$$

$$x_{d\_\min}\le x_d\le x_{d\_\max}, d=1,2,...,D \tag{5}$$

where *X* is a *D* dimension decision vector, and *Y* is the target vector. *N* to optimize the total number of targets, $g_i(x)\le 0$ is the ith inequality constraint, $h_j(x)=0$ is the jth equality constraint, $f_n(x)$ is the Nth target function; *X* is the decisive space for the decision vector formation, *Y* is the target space formed by the target vector. $g_i(x)\le 0$ and $h_j(x)=0$ determine the feasible domain solution, $x_{d\_\max}$ and $x_{d\_\min}$ are the upper and lower bounds for each vector search. The Pareto [13] optimal solution for solving multi-objective problems is defined as follows:

Pareto dominant: For any two vector $\mu,\ \ v\in\Omega$ is called μ dominated v, or v is dominated by μ, when and only when: $\forall i=1,2,...,M,\mu_i\le v_i$ and $\exists j=1,2,...,M,\mu_j<v_j$;

Pareto optimal solution: $x^*\in Q$ solution of a is called Pareto optimal solution or Pareto non - dominant solution, when and only as $\neg\exists x\in Q: x\succ x^*$;

Pareto optimal solution set: The set PS of all Pareto optimal solutions is called the Pareto optimal solution set PS=$\{x^*\mid\neg\exists x\in Q: x\succ x^*\}$

B. The principle of Particle Swarm Optimization

Similar to other algorithms based on swarm intelligence, particle swarm optimization (PSO) is also through the cooperation and competition among different particles in the group to achieve the search process in the search space to find the optimal location of the problem. The particle swarm algorithm first random initialization a uniform distribution in the optimization space of a given particle (population size is 30), then all the particles according to the two extreme value to update its own speed: one is the individual extreme value(pbest); the other is a group of extreme value (gbest). The mathematical description of the standard PSO, which is widely used at present, is: the total number of particles in the particle swarm is POPSIZE, the dimension of the particle is m, the termination condition of the algorithm (the maximum number of iterations) is $\mathrm{iter}_{\max}$, the flight velocity of the I particle at the t moment and the position in the search space are $v_i(t)=[v_{i1}(t),v_{i2}(t),\ldots,v_{im}(t)]^T$, $x_i(t)=[x_{i1}(t),x_{i2}(t),\ldots,x_{im}(t)]$; The individual extremum and the group extremum of the particle at *t* time are $pbest_i(t)=[p_{i1}(t),p_{i2}(t),...,p_{im}(t)]^T$, $gbest(t)=[g_1,g_2,...,g_m]^T$. All the particles fly in the search space in the following update way to find the optimal solution.

$$v_{i+1}(t+1)=\omega v_i(t)+c_1r_1(pbest_i(t)-x_i(t))+c_2r_2(\mathrm{g}best-x_i(t)) \tag{6}$$

$$x_{i+1}(t+1)=x_i(t)+v_{i+1}(t+1) \tag{7}$$

where ω is the inertia weight coefficient, c_1 and c_2 are learning factors of the algorithm. They affect the "self - learning" and "social learning" ability of the particles. r_1 and r_2 are random numbers between [0,1].

III. Improved Multiobjective Particle Swarm Optimization (PSO) Algorithm

A. Chaos Theory

Chaos is a form of motion in a nonlinear dynamic system, which has the characteristics of randomness, ergodicity and sensitivity to the initial value. Chaos phenomenon is a random and irregular movement occurring in a deterministic system. Its manifestation is uncertainty, but it has the characteristics of unrepeatable and unpredictable. The random motion state obtained by the deterministic equation is called chaos, and the variable which presents the chaotic state is called the chaotic variable. Document [11] proposes a single objective particle swarm optimization algorithm with chaotic sequence, and extends the mutation method to multi-objective particle swarm optimization to enhance its ergodicity in decision space. Logistic mapping is a typical chaotic system, and its formula is as follows:

$$c_d^{t+1}=4\times c_d^t\times(1-c_d^t) \tag{8}$$

In this formula: *t* stands for tth traversal searches, *d* represents variable sequence number, c_d^t doesn't include four fixed points of the chaotic equation.

The chaotic mapping is introduced into the multi-objective particle swarm optimization (PSO), and the initialization method of particle swarm is changed to reduce the uncertainty of the particle swarm. The ergodic characteristics of the chaotic mapping will make the optimal particle swarm go through all the states in the solution space as much as possible, and the diversity of the particle swarm will also increase. In this way, the chaotic traversal search is used after each iteration to avoid the early maturing of the particle swarm, thus improving the global optimization ability of the particle swarm.

In this paper, the initial population is generated: first, we randomly generate an initial solution x1, then generate population size according to our needs, generate N-1 individuals through formula (8), and complete the population generation.

B. Improvement of Particle Weight Updating

The inertia weight determines how much the speed of the previous iteration is, and it is one of the important parameters of the particle swarm optimization. In particle swarm optimization (PSO), the ability of global search and local search can be balanced by adjusting the size of it. The analysis shows that, in the early stage of particle swarm optimization, the selection of the larger inertia weight value can make the algorithm have a strong global search ability; In the latter part of the particle swarm optimization algorithm, the smaller inertia weight value can make the particle converge to the global optimal. In addition, when we update particles every time, we divide the governing relation of particles and add the non-dominated solutions into our file solutions, and the non-dominating solution is the solution of our approximate Pareto frontier, which is the best solution in this generation.

Therefore, the update of our inertia weight is divided into two kinds: A kind of inertia weight to file update solutions, to strengthen the local search ability, so the update formula such as (9); another is non-file update solutions, due to the non-dominated solution is the solution file, in order not to lose the diversity of population, we enhance the global search ability of non-file solution, the inertia weight the update formula is (10).

$$\omega = \frac{\omega_{\max} - (\omega_{\max} - \omega_{\min}) \times \mathrm{iter}}{iter_{\max}} \tag{9}$$

$$\omega = \frac{\omega_{\min} + (\omega_{\max} - \omega_{\min}) \times \mathrm{iter}}{iter_{\max}} \tag{10}$$

where $iter$ is the number of current iterations, and $iter_{\max}$ is the maximum number of iterations; $\omega_{\max}$ and $\omega_{\min}$ are the maximum and minimum of ω.

C. Simplified Mesh Reduction

After the particle update and cross operation, we choose the non-dominant solution as our file solution. With the increase of the number of iterations, the file solution will also increase, the distribution of the solution is more dense and the calculation cost increases. Therefore, we use the grid reduction strategy. The existing grid reduction strategy is triggered when the file solution exceeds the upper limit. Every time a file solution is discarded, it will trigger almost every upper limit in the later period, and the calculation frequency is very high. Once the file solution changes, the grid needs to be maintained so that the grid is likely to need to be rebuilt. Therefore, we improve the grid policy: We delete multiple solutions each time the grid is triggered, which reduces the grid trigger frequency and reduces the computational cost. The concrete steps are as follows:

- Step 1: The maximum minimum value $f_{i,\max}$ on each target is calculated, $f_{i,\min}$, i is the target dimension;
- Step 2: We set up a grid coordinate system. The average of each target is divided into SSS equal points, and the distance between each point of each target is $d = \frac{f_{i,\max} - f_{i,\min}}{sss}$, Grid coordinates are $f_{i,\min} + j \times d, (j = 1, 2, ..., sss)$;
- Step 3: Calculate the grid coordinates for each file solution. Each file solution is assigned to our grid coordinate system(Different file solutions may have the same grid coordinates);
- Step 4: We delete the file solution with the same grid coordinates. If we include extremal solution (boundary solution), we preserve the extreme value solution. Otherwise, we randomly select the same coordinate solution, and all other same coordinate solutions are discarded.

D. Introducing a Normal Distribution Crossover Operator

Because we need to enhance the diversity of the population, we introduce the normal distribution crossover operator (NDX) with a certain probability in the algorithm. In order to avoid the fact that the later algorithm is not close to the real Pareto frontier, we increase the probability of introducing the crossover operator in the later period of the algorithm. We use an increase in the probability of a linear increase in the number of iterations by P_0, such as formula (11). And a random $P \in [0,1]$ is generated, and when the P is less than P_0, the formula(12) is executed.

$$P_0 = k \times \frac{iter}{iter_{\max}} \tag{11}$$

$$\begin{cases} x_{1,i} = \dfrac{p_{1,i} + p_{2,i}}{2} + k' \times \dfrac{(p_{1,i} - p_{2,i}) \,|\, N(0,1) \,|}{2} \\ x_{2,i} = \dfrac{p_{1,i} + p_{2,i}}{2} - k' \times \dfrac{(p_{1,i} - p_{2,i}) \,|\, N(0,1) \,|}{2} \end{cases} \tag{12}$$

where k and k' are two different proportionality coefficients, which are defined by ourselves. $|N(0,1)|$ is a normal distribution random variable; $p_{1,i}$ and $p_{2,i}$ are the two parent of the ith variable; $iter$ is the number of current iterations; $iter_{\max}$ is the maximum number of iterations.

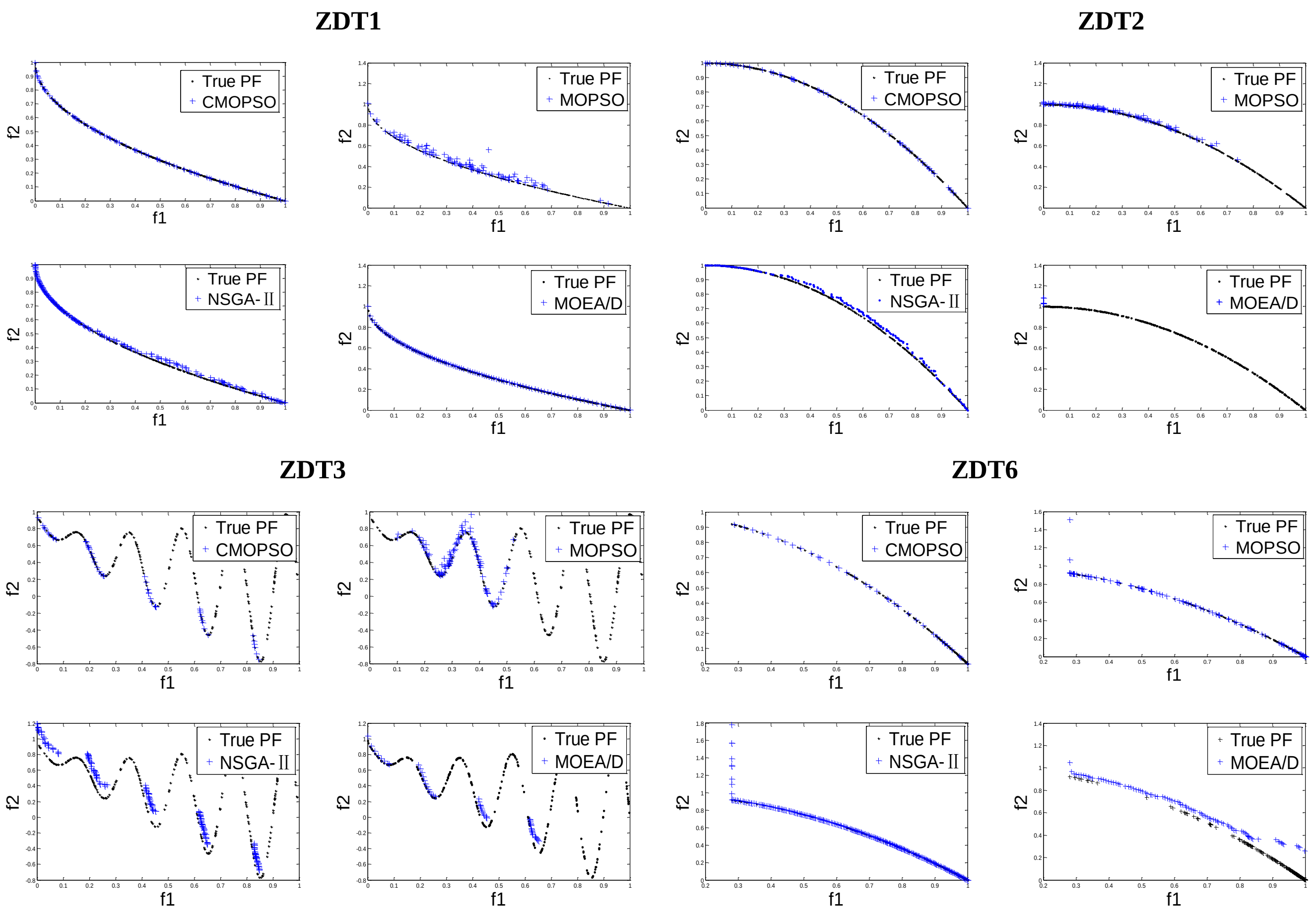
E. Gene Exchange

In order to improve the performance of the algorithm, we introduced the gene exchange in the file learning. We can improve the efficiency of search by exchanging information between the archival solutions of the elite. When all particles fly once, if the file solution has exceeded 100(to ensure sufficient diversity to avoid precocious maturity), if there is a grid reduction strategy, then gene exchange is performed after grid reduction strategy (the file after the grid is reduced is smaller and the solution is more representative). The following steps are as follows:

- Step 1: A random integer m($1 \le n \le d$, d is a decision space dimension. is generated), based on the random number m, to determine the corresponding M decision variable for each file solution.
- Step 2: Based on this decision variable, the decision variables in the last file solution are used in turn to replace the corresponding decision variables in the previous file solution and produce a new solution.
- Step 3: We calculate the corresponding target vector of the new solution. If the new target vector is better than the old target vector, then the decision variable in the previous file solution will use the new value, otherwise the old value will be used. After all the files are updated, delete those dominated files.

F. *Algorithm flow*

- Step 1: Initialization parameters: Initial iteration number $iter=1$, Maximum iteration number $iter_{\max}=1000$, $\omega_{\max}=0.9$, $\omega_{\min}=0.1$, The maximum upper limit of the file solution $\max_{dang}=100$, The learning factor $c_1=0, c_2=2$ of the algorithm, Grid equal number $sss=30$, Population size $N=300$, Cross coefficient of normal distribution $k=0.9$, $k'=1.481$, Coefficient of variation $P=0.05$;
- Step 2: Before the iteration, Produce N particles by the formula (8),and their fitness values are kept in our f matrix;
- Step 3: The initial screening solution added to the file to ensure that the file solution only retained the non-dominant solution(non-inferior solution);
- Step 4: It begins to circulate update iterations and randomly selects a particle in the file solution as a global optimal. In the multi-objective problem, the individual optimality is no longer effective, so we no longer choose the individual optimal;
- Step 5: The multiplicity of the solution is increased by adding random reinitialization before the particle is updated. Setting up P_0=0.05;
- Step 6: According to formula (6) and (7) updating particle velocity and location, we have the particles that exceed the decision space. We reverse the particle velocity and set the location of these particles as boundary particles;
- Step 7: According to the formula (11) and (12), new solutions and files are solved to produce new solutions;
- Step 8: According to every particle we update, if the particle dominates some solutions in the file solution, we delete these solutions and add the new solution to the file. If the particles are dominated by the file solution, we abandon the particle;
- Step 9: According to 2.5, the strategy of gene exchange was carried out to update the file solution;
- Step 10: If $iter<=iter_{\max}$, return to step 4, otherwise jump out of the loop and get the Pareto optimal solution.



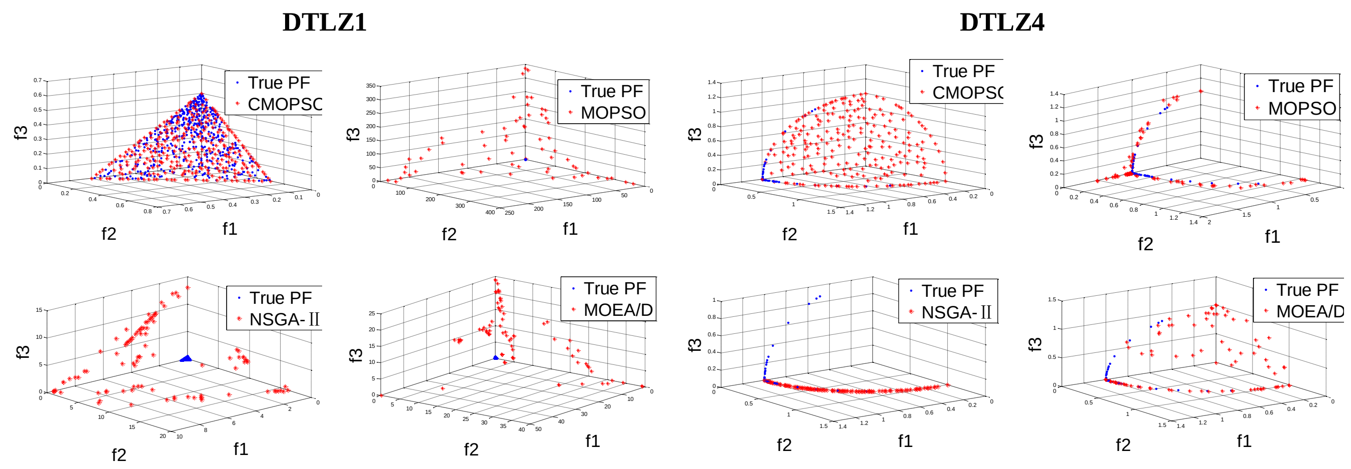


Fig. 1. Comparison of test functions on 4 algorithms.

TABLE I: IGD INDEX STATISTICS

Test function	Comparison method	Algorithm			
		C-MOPSO	MOPSO	NSGA-II	MOEA/D
ZDT1	Mean	**5.289e-04**	8.491e-04	8.407e-04	2.965e-02
	Std	6.254e-05	4.464e-04	1.256e-04	5.412e-04
	p-value	9.607e-29	2.585e-11	7.559e-26	3.383e-52
ZDT2	Mean	**6.962e-04**	7.810E-04	7.458e-03	1.757e-02
	Std	1.652e-04	2.403E-04	8.507e-03	7.056e-04
	p-value	3.255e-20	3.762E-17	4.402e-05	2.829e-42
ZDT3	Mean	1.418e-02	2.401e-02	**1.242e-02**	1.885e-02
	Std	1.776e-04	1.729e-03	1.278e-03	6.247e-03
	p-value	6.266e-57	6.202e-35	1.779e-30	2.685e-16
ZDT6	Mean	**7.833e-04**	3.895e-04	1.921e-02	4.800e-02
	Std	2.619e-04	5.566e-05	2.376e-02	6.866e-04
	p-value	3.392e-16	2.134e-26	1.238e-04	2.888e-55
DTLZ1	Mean	**7.514e-04**	4.628e+00	2.843e-02	3.266e-01
	Std	7.547e-05	5.185e+00	1.117e-01	1.394e-01
	p-value	8.885e-31	3.450e-05	1.740e-01	1.744e-13
DTLZ4	Mean	**4.581e-04**	3.697e-02	8.933e-04	9.625e-04
	Std	1.124e-04	1.807e-02	8.690e-04	8.221e-04
	p-value	8.301e-20	4.722e-12	4.413e-06	5.170e-07

IV. PERFORMANCE TEST AND ANALYSIS OF EXPERIMENTAL RESULTS

A. *Test Functions, Test Environments, and Performance Indicators*

In order to verify the effectiveness of the improved multi-objective particle swarm optimization algorithm, we choose the typical multi-objective test function set ZDT [14] and DTLZ [15], the representative six kinds of multi-objective functions (ZDT1, ZDT2, ZDT3, ZDT6, DTLZ1, ZTLZ4) to carry out experimental simulation comparison. In this paper, Our m=12 and α=100 in DTLZ4.

We use the comprehensive index (Inverted Generational Distance, IGD) [16] to evaluate the proposed algorithm. The test environment is: Windows10, CPU is Intel Xeon-E3-1240v5, 3.5GHZ, and 8G. The test software is MATLAB 2014A. IGD is a metric to measure the distance between the real Pareto frontier and the approximate Pareto frontiers obtained by the algorithm. The lower the IGD value, the better the convergence and diversity of the approximate Pareto fronts obtained by the algorithm, and the closer to the real Pareto frontiers. The calculation formula is as follows:

$$IGD(P,P^*)=\frac{\sum_{i=1}^{|P|}d(P_i,P^*)}{|P|} \qquad (13)$$

Among them: P is an ideal set of uniform sampling, P^* is a set of myopic Pareto solutions obtained through the algorithm, $|P|$ is the size of the population P^*, $d(P_i,P^*)$ is the minimum Euclidean distance between P_i and population P^*.

B. *Results and Analysis of Simulation Experiment*

Three kinds of multi-objective classical algorithms are set up in this paper to analyze the chaos based multi-objective particle swarm optimization (CMOPSO) algorithm. It includes multi-objective particle swarm optimization (MOPSO), non-dominated sorting algorithm NSGA-II [17] and multi objective evolutionary algorithm MOEA/D [18]. The parameters of the algorithm are set to: $iter$=1 , $iter_{\max}=1000$, $\omega_{\max}=0.9$, $\omega_{\min}=0.1$, The maximum upper limit $\max_{\text{dang}}$=100 of the solution, $c_1=0, c_2=2$, $sss=30$, $N=300$, k=0.9 , k'=1.481 , The mutation probability is $P=0.05$, The optimal solution and the real optimal solution are shown in Fig. 1.

From Fig. 1, we can see that under the ZDT series of two test functions, all the results are better than other algorithms. In particular, the three functions of ZDT1, ZDT3 and ZDT6 are almost close to the real Pareto frontiers.

C. *Performance Comparison of Multiobjective Optimization Problems*

The 4 algorithms run 30 times separately, the average value of the IGD performance index (Mean), the standard deviation (Std), as shown in Table I. The blackbody coarse font is the optimal value obtained by each algorithm on the same test problem. The p-value value is the p-value value of the 5%t- test that we have shown the significant level of this algorithm and other contrast algorithms on the same test problem.

From Table I, we can see that after 30 times of variance and double tail detection, the mean and variance of C-MOPSO on two-dimensional multi-target test functions ZDT1, ZDT2 and ZDT6 are superior to those of other algorithms. Due to the dominance relationship between the optimal solutions of our ZDT3 functions, we get a certain gap between the obtained solutions and the optimal solutions. But for the governing solutions, the solution is better than NSGA- II. In 3D, the mean of CMOPSO is better than our other algorithms by comparing the four algorithms of two functions. The following figure is a comparison of the rate of convergence and the size of the IGD index of the various algorithms of the same function at each iteration.

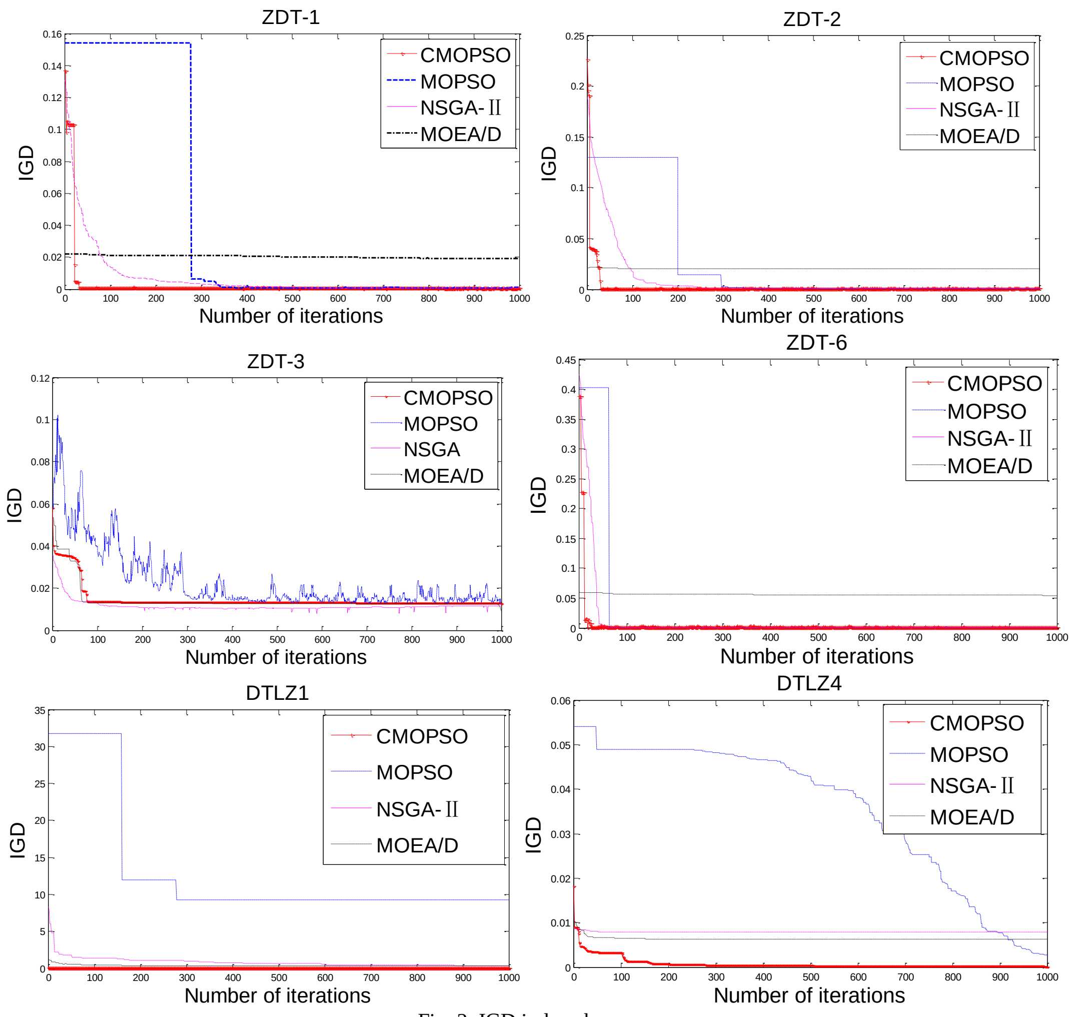


Fig. 2. IGD index changes.

From Fig. 2, we can see that the ZDT3 function is discontinuous and the optimal solution is continuous, but the effect is better. By comparing the convergence rate and effect of the IGD index of the four algorithms on six functions (ZDT1, ZDT2, ZDT3, ZDT6, DTLZ1, DTLZ4), the algorithm proposed in this paper is superior to the other three algorithms.

V. Conclusion

In order to improve the efficiency of solving multi-objective optimization problems, a multi-objective optimization algorithm of chaotic particle swarm optimization is proposed in this paper. In this algorithm, a method of initializing particle swarm based on chaotic sequence is presented to select the global optimal particle randomly and guide the evolution direction of the population. The cross strategy of normal distribution is adopted to increase the diversity of the solution. And we add the grid reduction strategy to reduce the computational cost. Because of better solution in the decision space, there is a good one dimension or multidimensional, so we add the gene exchange to improve the convergence speed. The results of comparison test show that the proposed algorithm can obtain high quality Pareto sets with better convergence, more uniform distribution and wider coverage. The next step is to improve the performance of the algorithm for the multi-objective problem of higher dimension, and apply it to the engineering design in order to realize its social value.

Acknowledgment

Project supported by Natural Science Foundation of China (11371015),Natural Science Foundation of Sichuan Education(18ZA0469), Excellence Research Foundation of

China West Normal University (17YC385) and University Innovation Team of China West Normal University(CXTD2015-4).

References

[1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. the IEEE Int Conf. on Neural Networks. Piscataway*, IEEE, 1995, pp. 1942-1948.

[2] H. M. Xu, *Research on multi-Objective Particle Swarm Optimization Algorithms*, Shanghai: School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, 2013.

[3] F. Li *et al*., "Multi-objective particle swarm optimization algorithm based on decomposition and differential evolution," *Control and Decision*, vol. 32, no. 3, pp. 403-410, 2017.

[4] H. Wang and F. Qian, "Multi objective optimization algorithm for dynamic particle swarm optimization based on congestion and variation," *Control and Decision*, vol. 23, no. 11, pp. 1238-1242, 2008.

[5] A. Kaveh and K. Laknejadi, "A novel hybrid charge system search and particle swarm optimization method for multi-objective optimization," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15475-15488, 2011.

[6] L. B. Zhang *et al*., "Solving multi-objective optimization problem based on particle swarm optimization," *Computer Research and Revelopment*, vol. 41, no. 7, pp. 1286-1291, 2004.

[7] N. Yang *et al*., "Multi-objective particle swarm optimization algorithm based on multilevel information interaction," *Control and Decision*, vol. 31, no. 5, pp. 907-912, 2016.

[8] K. W. Li and Y. Z. Zhang, "Multi-objective particle swarm optimization algorithm based on dynamic adjustment," *Application of Computer System*, vol. 26, no. 7, pp. 161-166, 2017.

[9] J. M. Yang *et al*., "Improved multi-objective particle swarm optimization algorithm based on multiple strategies," *Control and Decision*, vol. 32, no. 3, pp. 435-442, 2017.

[10] L. Li *et al*., "Multi-objective particle swarm optimization based on grid ranking," *Computer Research and Development*, 2017, vol. 54, no. 5, pp. 1012-1023.

[11] J. M. Yang and M. M. Ma, "Multi-objective adaptive chaotic particle swarm optimization algorithm," *Control and Decision*, vol. 30, no. 12, pp. 2169-2170, 2015.

[12] J. H. Xiong and F. Xie, "Improved particle swarm optimization algorithm based on Chaos Theory and hybrid strategy," *Computer Application Technology*, vol. 10, no. 55-56, 2017.

[13] Q. Lin, J. Li *et al.*, "A novel multi-objective particle swarm optimization with multiple search strategies," *European J of Operational Research*, vol. 247, no. 3, pp. 732-744, 2015.

[14] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multi-objective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173-195, 2000.

[15] K. Deb *et al*., "Scalable multi-objective optimization test problems," in *Proc. the Congress on Evolutionary Computation*, Honolulu: IEEE, pp. 825-830, 2002.

[16] F. Neri and C. Cotta, "Memetic algorithms and memetic computing optimization: A literature review," *Swarm and Evolutionary Computation*, vol. 2, pp. 1-14, 2012.

[17] K. Deb *et al.*, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Trans on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.

[18] Q. F. Zhang and H. Li, "MOEA/D: A multi-objective evolutionary algorithm based on decomposition," *IEEE Trans. on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, 2007.

Liansong Xu was born in October 1991, in Mount Emei, Sichuan, China. He graduated from the School of Mathematics and Information, China West Normal University. Now he is a graduate student of the school of mathematics and information at the China West Normal University. He is mainly engaged in the design and research of the algorithm.

Dazhi Pan was born in January 1974 at Mianyang, Sichuan. He received a degree in applied mathematics from China West Normal University, Nanchong, China in June 1996 and his M.S. degree in computer applications from Kunming University of Science and Technology, Kunming, China in June 2003 and Ph.D. degree in petroleum engineering computation from Southwest petroleum University, Chengdu, China in January 2012. He is a professor at the school of mathematics and information China West Normal University, China. His current research interest includes intelligent computing and algorithm analysis and design.