Design and Implementation of Improved Multi-objective Genetic Algorithm Based on Uniform Distribution

Lianshuan Shi and Shuangyu Duan(✉)

School of Information Technology and Engineering, Tianjin University of Technology and Education, Tianjin 300222, China
shilianshuan@sina.com

Abstract. In production and daily life, with the development and progress of various technologies, many practical problems gradually transform from the form of a single goal to the form of multiple goals. In single objective optimization problems, only finding the optimal solution that satisfies the conditions for a single objective is considered, while multi-objective optimization problems are looking for the optimal solution set that satisfies multiple objectives simultaneously under given conditions When solving multi-objective optimization problems, the NSGA II algorithm is generally used When using NSGA II algorithm to solve multi-objective optimization problems, there are some problems such as slow Rate of convergence, weak stability, and easy to fall into local optimization. An improved algorithm was proposed to address this issue. Introducing interval uniform distribution module during population initialization, dynamically adjusting crossover and mutation probabilities during the algorithm process. The improved multi-objective genetic algorithm based on uniform distribution was applied to classic examples and compared with the application results of NSGA II algorithm and particle swarm optimization algorithm. The experimental results show that the improved multi-objective genetic algorithm performs best in terms of uniformity and convergence, followed by the NSGA II algorithm, and the particle swarm algorithm has the worst results in comparative experiments.

Keywords: multi-objective optimization · genetic algorithm · adaptive strategy · uniform distributed

1 Introduction

Multi objective programming problems are widely present in engineering and practical life [1–3], and genetic algorithms have shown their advantages in solving multi-objective optimization problems. Genetic algorithm performs genetic operations on all individuals in the population to obtain the optimal solution, with strong global optimization ability. Srinivas and Deb [4] proposed a non dominated sorting genetic algorithm, namely the NSGA algorithm. However, due to the high time complexity of the non dominated

Y. Cao and X. Shao (Eds.): DIONE 2023, LNICST 515, pp. 120–134, 2025.
https://doi.org/10.1007/978-3-031-80713-8_9

sorting genetic algorithm during operation, when the population is large, this sorting will consume a lot of time, thereby reducing search efficiency. To address the above shortcomings, Deb et al. [5] improved the non dominated sorting genetic algorithm and proposed the non dominated sorting genetic algorithm with elite strategy, namely NSGA II algorithm. The algorithm incorporates an elite retention strategy to retain some excellent individuals, improving the accuracy of the algorithm. Additionally, by introducing a crowding comparison operator, the diversity of the population is enhanced. The traditional NSGA II algorithm still has some problems such as being prone to falling into local optima and weak stability. Chen Jie et al. [5] proposed an improvement strategy that uses sorting fitness strategy, on-demand layering strategy, and other methods to improve the convergence speed of the algorithm. Zhang Yi et al. [6] proposed an orthogonal design NSGA II algorithm and applied it to the problem of automotive brakes. Through experiments, it proved that a good Pareto optimal solution set was obtained. Furtuna et al. [7] designed an application software based on the elite strategy's non dominated sorting genetic algorithm (NSGA II algorithm) to solve the optimization problem in the synthesis process of polysiloxane.

2 Improved Multi-objective NSGA II Algorithm

This article proposes an improved multi-objective genetic algorithm based on uniform distribution, based on the NSGA II algorithm. When initializing the population, the algorithm adds a uniform partitioning module to make the distribution of the initial population more uniform and diverse. By dynamically adjusting the crossover and mutation probabilities, the convergence of the obtained solution set is better. Using multi-objective functions for simulation experiments to verify the improvement effect, the results show that the improved multi-objective genetic algorithm based on uniform distribution can effectively solve the problems existing in the traditional NSGA II algorithm, resulting in significant improvements in accuracy, convergence, and uniformity of the algorithm.

2.1 Introduction of Uniformly Distributed Modules

Random initialization methods are generally used during population initialization, and populations using random initialization methods may experience uneven distribution of initialization populations. In order to improve the distribution of the initial population and avoid the algorithm falling into local optima due to uneven distribution of initial individuals, an interval uniform distribution module is added during population initialization. The initial population of the genetic algorithm is evenly divided into several regions, and random methods are used in each small region to ensure that there is a distribution of random individuals in each small region.

This method evenly blocks the target interval of the population, where the population distribution in each region can be represented by Eq. 1.

$$\mathrm{D} = \left(\mathrm{D}_{1,}\mathrm{D}_{2,} \cdots \mathrm{D}_{\mathrm{n}}\right) \tag{1}$$

In the equation, D represents the set of all regions in the target interval of the population, n represents the number of small regions divided into, and D_{n} represents the nth interval.

Assuming that the initial population of the traditional NSGA II algorithm has 150 individuals, if the target interval is divided into 15 blocks, in order to maintain a constant number of initial populations, it is specified that the number of individuals randomly initialized within each interval is 10. This ensures both the uniformity of the population distribution interval and the randomness of the individuals. As shown in Fig. 1.

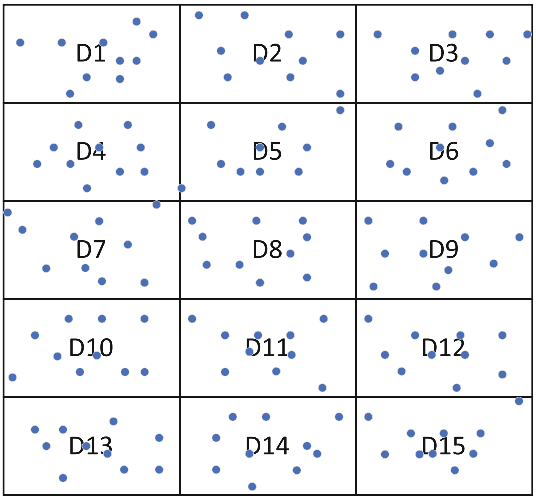


Fig. 1. Initial population distribution after uniform partition

2.2 Dynamic Adjustment of Crossover Probability and Mutation Probability

In the process of population evolution, if the value of the crossover probability P_c is set to be larger, the speed of generating new individuals will be faster, but at the same time, individuals with higher fitness will be replaced quickly, and excellent individuals cannot be retained; If the value of the crossover probability P_c is set to a small value, it will lead to the algorithm reaching the convergence state too early and unable to find the true optimal solution of the algorithm. Therefore, this article proposes a dynamic method for adjusting the crossover probability. In the initial stage of population iteration, in order to improve population richness, a larger crossover probability is adopted to avoid the algorithm falling into local optima; In the later stage of population iteration, in order to improve the convergence of the algorithm, the crossover probability value should be appropriately reduced. To address the above issues, the following improvements are made to the crossover operator, as shown in Eq. 2.

$$P_c = \frac{3}{20}\sin\left(\frac{n}{N} \times \pi + \frac{\pi}{2}\right) + \frac{3}{4} \tag{2}$$

In the equation, P_c represents the value of the crossover probability, n represents the current algebra, and N represents the total number of iterations.

At the beginning of population iteration, in order to maintain population richness, the mutation probability is appropriately reduced to maintain the acquisition of excellent individuals; When the population evolves to the later stage, increasing the probability of population variation introduces new individuals, making the diversity of the population more abundant. The mutation operator makes the following improvements, as shown in Eq. 3.

$$P_m = \frac{1}{40}\sin\left(\frac{n}{N} \times \pi - \frac{\pi}{2}\right) + \frac{3}{40} \tag{3}$$

In the equation, P_m represents the value of mutation probability, n represents the current algebra, and N represents the total number of iterations.

3 Implementation Process of Improved NSGA II Algorithm

(1) Initialize the relevant parameters of the genetic algorithm: the population size is N1, the maximum number of iterations is set to T1, and the evolutionary algebra counter is represented by n;
(2) Initialize the population using a uniform partitioning module;
(3) After non-dominated sorting, the first generation subgroup is generated through improved selection, crossover, and mutation operators using the NSGA - II algorithm;
(4) Determine whether to generate a new parent population. If so, perform the improved selection, crossover, and mutation operator process. Otherwise, perform non-dominated sorting and crowding calculation, select suitable individuals to form a new parent population, and then perform the improved selection, crossover, and mutation operator process;
(5) Determine whether the conditions for program completion are met. If the conditions are met, the algorithm ends. Otherwise, continue with step (4) of the loop.

4 Test Function and Evaluation Indicators of the Improved NSGA II Algorithm

4.1 Test Functions

This article selects ZDT1, ZDT2, ZDT3, and ZDT6 from the standard test functions ZDT, running in the following environment: operating system: Windows10, processor: Intel ® Core (TM) i5-4210UCPU @ 1.70 GHz 2.40 GHz, installed memory: 8.00 GB, software: MATLAB R2019a, programming language: MATLAB language.

4.2 Evaluation Indicators of the Algorithm

This article uses the Inverse Generational Distance (IGD) evaluation index as the evaluation index of the algorithm. The smaller the IGD value, the better the convergence and distribution of the algorithm. The formula is shown in Eq. 1.

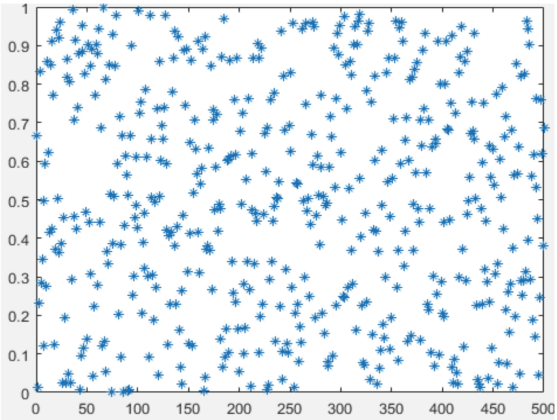
$$\mathrm{IGD}(P, Q) = \frac{\sum_{v \in P} d(v, Q)}{|P|} \tag{4}$$

Among them, P is the set of points distributed on the real Pareto surface, Q is the obtained Pareto optimal solution set, d(v, Q) is the minimum Euclidean distance from individual v in set P to set Q, and |P| is the number of point sets distributed on the real Pareto surface. From the above equation, it can be seen that if the algorithm has good convergence performance, the value of d(v, Q) is relatively small, which is used to determine the convergence of the algorithm; If the algorithm has poor distribution performance, the individuals in the population will gather together and the corresponding values will d(v, Q) increase to judge the distribution of the algorithm.

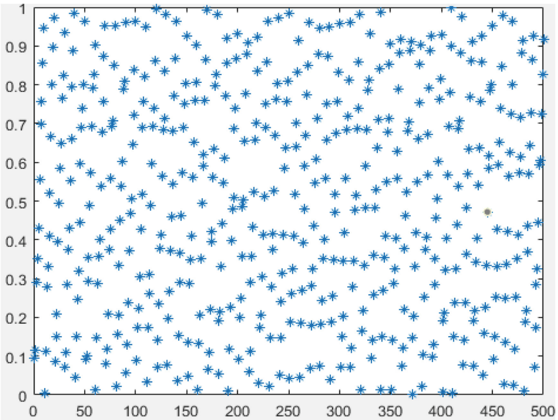
5 Test Interval Uniform Distribution Module

5.1 Testing of Individual Distribution of Initial Population

To demonstrate the effectiveness of introducing the interval uniform distribution module, the population initialization of the NSGA II algorithm was improved. In each test function, the population size and iteration number of the algorithm were set to 500. Figures 4, 5, 6, 7 show the distribution of individuals after population initialization before and after the improvement. Through the comparison between (a) and (b) figures, it can be intuitively seen that before improving the population initialization method, as shown in (a) figure in Fig. 2, although the individual distribution of the ZDT1 test function is much better than expected, there are still some areas where clustering occurs, and some areas are empty sets. After introducing the interval uniform distribution module, as shown in (b) figure in Fig. 4, the distribution of individual population is relatively uniform, Ensure that there are individuals present in each 0.1 interval in the vertical axis. For functions ZDT2, ZDT3, and ZDT6, the effectiveness of introducing interval uniform distribution modules was also demonstrated through image comparison, which improved the diversity of the initial population (Fig. 3).



(a)

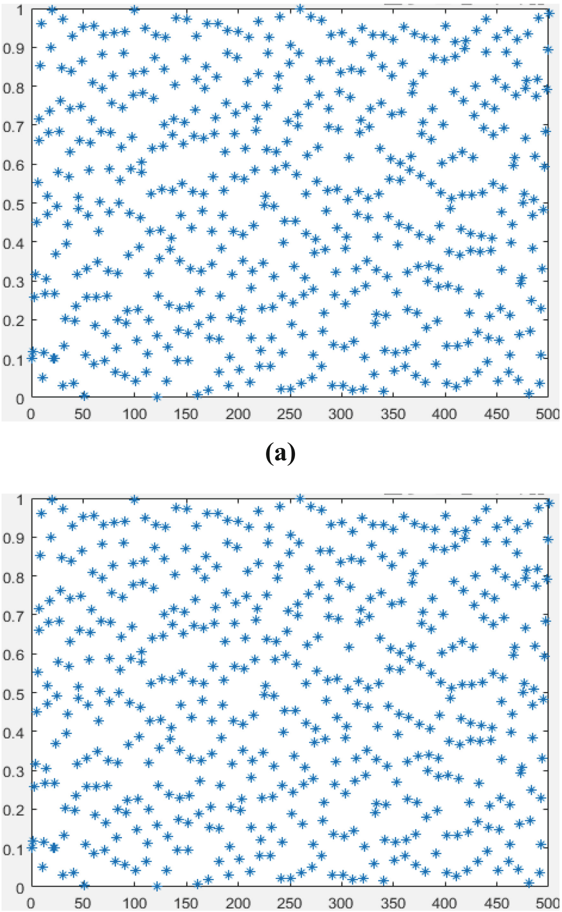


(b)

Fig. 2 (a). Distribution of ZDT1 random initialization (b). Initial distribution of ZDT1 interval uniform distribution

5.2 Testing and Analysis of Experimental Results

In order to observe the effectiveness of the improved algorithm more intuitively, 500 individuals were selected for testing and images of the experimental results were drawn in this section of the experiment, as shown in Figs. 6, 7, 8, 9 . (a) group showed the test results of the NSGA II algorithm, and (b) group showed the test results of the introduction of a uniform interval module. It can be clearly seen from Fig. 6 that (b) is closer to the real Pareto optimal solution frontier compared to (a), and the continuity is better than (a). From Fig. 7, it can be observed that the Pareto optimal leading edge of the two images is relatively close. Compared with the latter half of the images in (a) and (b), (b) is clearly closer to the real Pareto optimal leading edge than (a). For Fig. 8 (b), the graph is also closer to the real Pareto optimal solution frontier than (a), but there is little difference

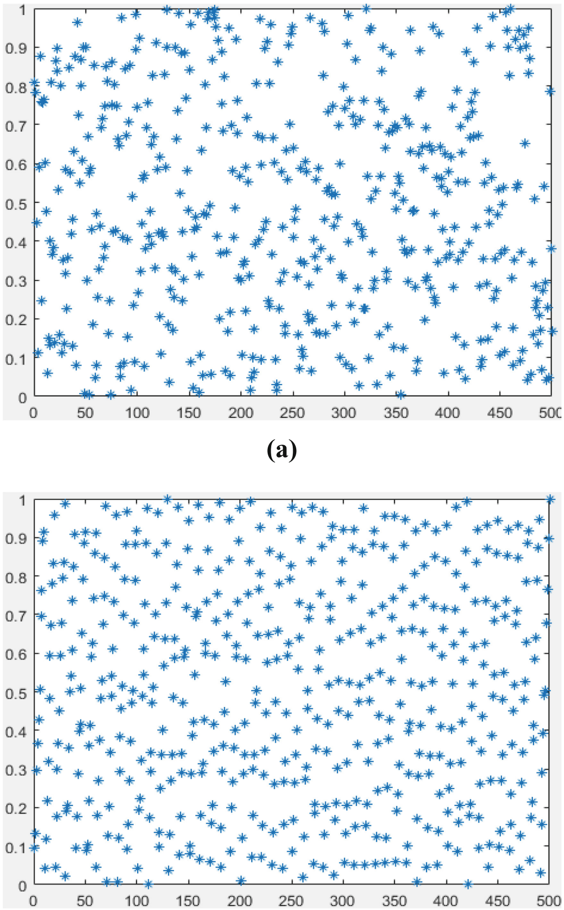


(a)

(b)

Fig. 3 (a). Distribution of ZDT2 random initialization (b). Initial distribution of ZDT2 interval uniform distribution

in continuity and uniformity between the two. Observing Fig. 9, by comparing the two ends of the image, it can be seen that (b) is much lower than (a), indicating that (b) is closer to the real Pareto optimal solution front, and (a) there are many discrete points in the front section of the image, but there is not much difference in the uniformity of the rear section between the two. From the overall image perspective, the algorithm that introduces a uniform interval module has more advantages in continuity than the NSGA - II algorithm, and is more effective in searching for the forefront of Pareto optimal solutions.



(a)

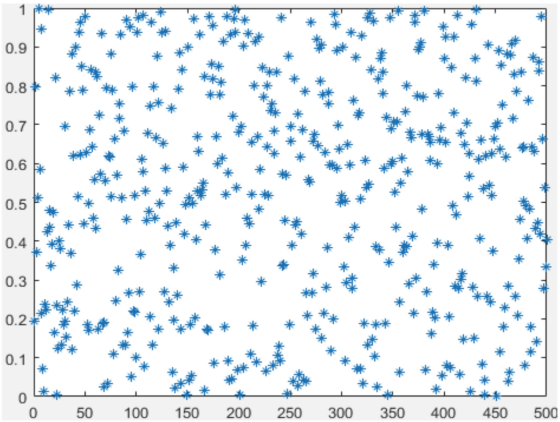
(b)

Fig. 4 (a). Distribution of ZDT3 random initialization (b). Initial distribution of ZDT3 interval uniform distribution

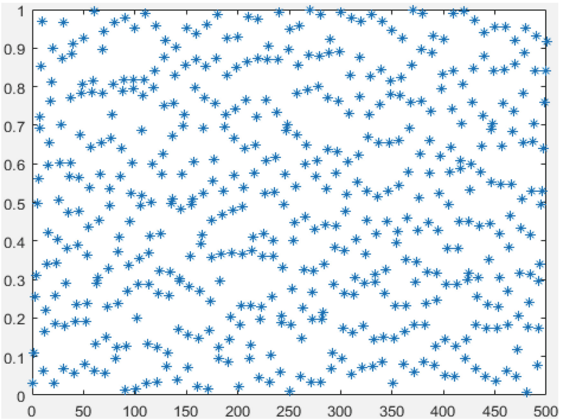
6 Simulation Experiment and Result Analysis of Improved NSGA II Algorithm

6.1 Parameter Settings

Test the improved algorithm using standard test functions and compare it with NSGA - II algorithm and particle swarm optimization algorithm. In Chapter 2, the relevant theoretical knowledge of NSGA II algorithm and particle swarm optimization algorithm has been introduced. Here, the relevant parameters of the NSGA II algorithm are set as follows: the crossover probability is specified as 0.7, and the mutation probability is specified as 0.04; Set the relevant parameters of the particle swarm optimization algorithm to: reset the inertia weight to 0.6, and set the learning factors c1 and c2 to 2;



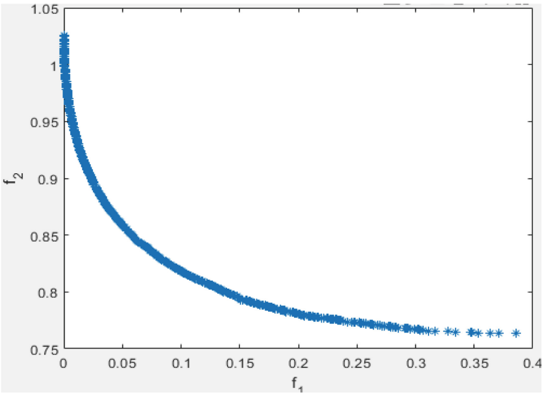
(a)



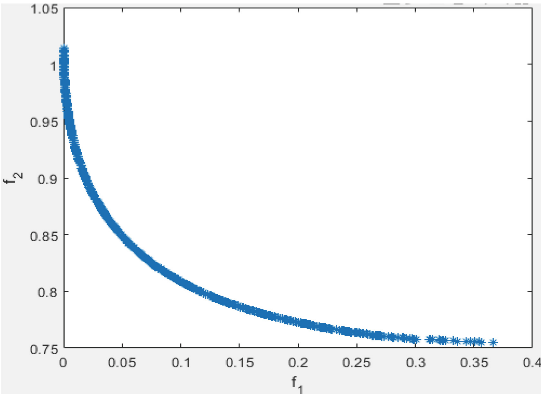
(b)

Fig. 5 (a). Distribution of ZDT6 random initialization (b). Initial distribution of ZDT6 interval uniform distribution

The improved NSGA II algorithm divides the initial population module into 10 equal intervals, and the crossover and mutation probabilities are calculated according to the formula mentioned above. Set the population size to 300 and the evolutionary algebra to 250.



(a)



(b)

Fig. 6 (a). Test results of ZDT1 NSGA - II (b). Test results of ZDT1 introducing uniform interval module

6.2 Simulation Experiments and Result Analysis

In order to measure the performance of the improved algorithm, the commonly used inverse generation distance evaluation index for multi-objective optimization problems is introduced, and the improved algorithm is abbreviated as UDMGA. For the three algorithms to be tested, each experiment was independently run 20 times, and the IGD values of each algorithm during the ZDT function testing were recorded. The optimal, worst, average, and standard deviations were retained, and the results are shown in Table 1.

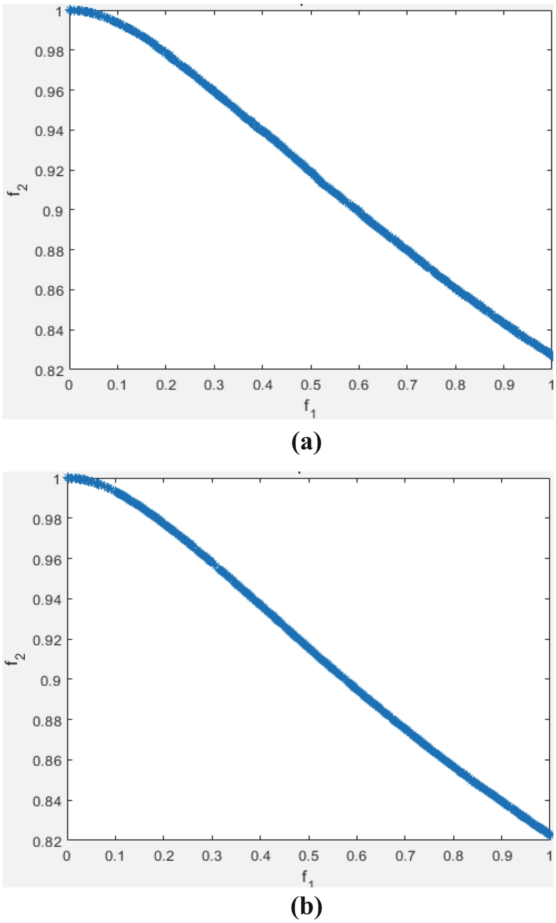
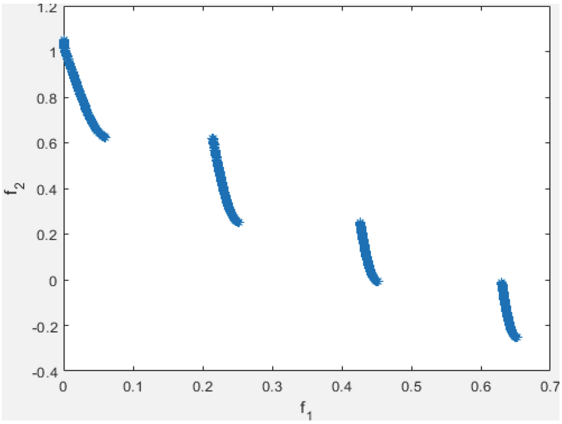
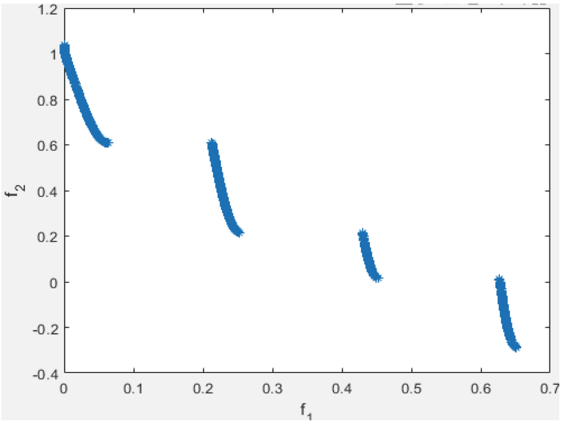


Fig. 7 (a). Test results of ZDT2 NSGA - II (b). Test results of ZDT2 introducing uniform interval module

From Table 1, it can be seen that for ZDT1, the UDMGA algorithm is superior to the NSGA II algorithm and PSO algorithm in terms of worst case, average, and standard deviation. The NSGA II algorithm is the best in terms of optimal values, followed by the UDMGA algorithm, and the PSO algorithm is the worst. For ZDT2, the UDMGA algorithm outperforms the NSGA II algorithm and PSO algorithm in terms of worst case, mean, and standard deviation. The NSGA II algorithm performs best in terms of optimal values, followed by the UDMGA algorithm, PSO algorithm, and NSGA II algorithm is still better in the other three values. For the piecewise function ZDT3, the PSO algorithm still performs the worst on these four values, while the NSGA - II algorithm only performs better on the optimal value than the UDMGA algorithm, while the other three values are not as good as the UDMGA algorithm. In the test function ZDT6, the UDMGA algorithm outperforms the NSGA II algorithm and PSO algorithm in terms of optimal value, worst case, average value, and standard deviation. The NSGA II algorithm ranks



(a)

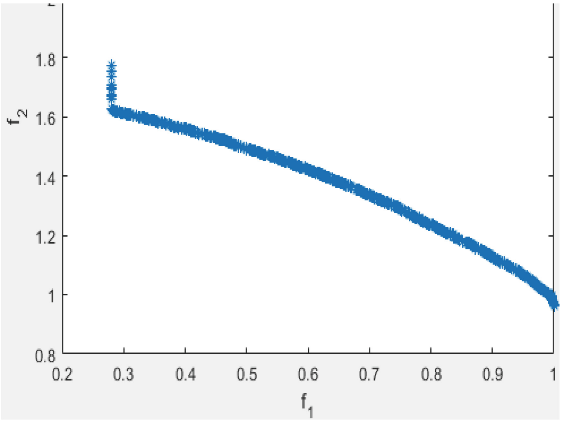


(b)

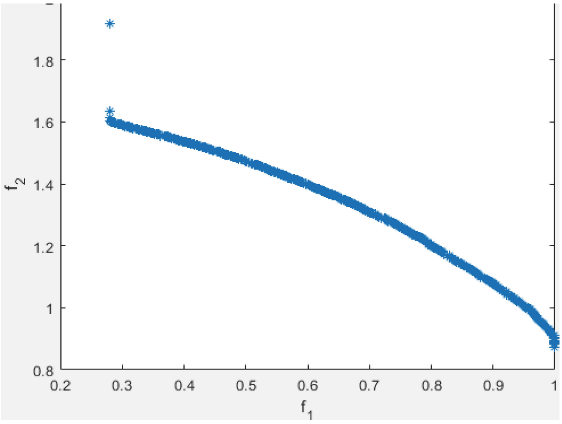
Fig. 8 (a). Test results of ZDT3 introducing uniform interval module (b). Test results of ZDT3 introducing uniform interval module

second, while the PSO algorithm still ranks last. In summary, the UDMGA algorithm is only slightly inferior to the NSGA - II algorithm in terms of optimal values. Overall, this algorithm has more advantages, while the PSO algorithm performs poorly, proving that the improved algorithm has better convergence and distribution.

Based on the above situation, the convergence of the three algorithms is relatively good, but in terms of distribution, the PSO algorithm performs the worst, followed by the NSGA - II algorithm. The UDMGA algorithm performs the best among the four test functions, so the improved UDMGA algorithm has a significant improvement in distribution.



(a)



(b)

Fig. 9 (a). Test results of ZDT6 NSGA - II (b).Test results of ZDT6 introducing uniform interval module

Table 1. IGD operation results of UDMGA, NSGAII and PSO for four test functions

Test Function	Algorithm	Optimal Value	Worst Value	Mean Value	Standard Deviation
ZDT1	NSGAII	1.66848E-03	2.02309E-03	1.84955E-03	9.56934E-5
	PSO	1.11614E-02	1.77895E-02	1.41833E-02	1.92180E-3
	UDMGA	1.67339E-03	1.88940E-03	1.77618E-03	6.36025E-5

(*continued*)

Table 1. (*continued*)

Test Function	Algorithm	Optimal Value	Worst Value	Mean Value	Standard Deviation
ZDT2	NSGAII	1.69419E-03	2.04676E-03	1.85805E-03	8.55208E-5
	PSO	6.70733E-03	1.32055E-02	8.84498E-03	1.52467E-3
	UDMGA	1.70135E-03	1.99306E-03	1.81299E-03	7.15675E-5
ZDT3	NSGAII	1.71340E-03	2.30127E-03	2.02904E-03	1.40988E-4
	PSO	2.76112E-03	3.73975E-03	3.23168E-03	2.62997E-4
	UDMGA	1.73292E-03	2.11475E-03	1.93608E-03	1.07047E-4
ZDT6	NSGAII	1.09542E-03	2.77198E-03	1.52374E-03	4.61499E-4
	PSO	1.89115E-03	6.28160E-03	3.70850E-03	1.25625E-3
	UDMGA	1.08459E-03	2.14103E-03	1.34914E-03	2.39895E-4

7 Summary

This article mainly improves on the standard NSGA - II algorithm. Firstly, in the initialization stage, in order to avoid the algorithm falling into local optima due to uneven distribution of initial individuals in the solution space, an interval uniform distribution module is introduced. In genetic operations, the crossover probability and mutation probability are dynamically adjusted according to the changes in the iteration process. Experimental verification and analysis are conducted using the ZDT standard test function. The experiments show that the improved algorithm obtains a more uniform distribution of the solution set and better convergence.

References

1. Guo, X., Ji, M., Zhao, Z., Wen, D., Zhang, W.: Global path planning and multi objective path control for unmanned surface vehicle based on modified particle swarm optimization (PSO) algorithm. Ocean Eng. **216** (2020)
2. Hu, L., et al.: A multiobjective optimization approach for COLREGs compliant path planning of autonomous surface vehicles verified on networked bridge simulators. IEEE Trans. Intell. Transp. Syst. **21**(3) (2020)
3. Guan, Z.: Preliminary study on multi objective evolutionary algorithms. J. Shijiazhuang School Econ. (02), 125–128+149 (2002)
4. Srinivas, N., Deb, K.: Multiobjective optimization using non optimized sorting in genetic algorithms. Evol. Comput. **2**(3) (1994)
5. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and efficient multi-objective genetic algorithm: NSGA II. IEEE Trans. Evol. Comput. **6**(2) (2002)
6. Chen, J., Xiong, S., Lin, W.: Research on improvement strategies of NSGA - II algorithm. Comput. Eng. Appl. **47**(19), 42–45 (2011)
7. Zhang, Y., Chen, P., Wan, X.: Multi-objective optimization of brakes based on orthogonal design NSGA - II algorithm. Comput. Appl. Res. **33**(03), 720–725 (2016)

8. Furtuna, R., Curteanu, S., Leon, F.: An elitist non dominated sorting genetic algorithm enhanced with a neural network applied to the multi objective optimization of a polysiloxane synthesis process. Eng. Appl. Artistic Intell. **24**(5) (2011)
9. Huang, Y., Yao, X., Shen, H.: The application of entropy enhanced chaotic particle swarm optimization algorithm in workshop scheduling. Modular Mach. Tools Autom. Process. Technol. (09), 152–155 (2018)
10. Wang, L., Deng, J., Wang, S.: Overview of research on distributed shop floor scheduling optimization algorithms. Control Decision Making, **31**(01), 1–11 (2016)
11. Jiangang, P., Mingzhou, L., Mingxin, Z., Zhang, X., Ge, M.: Evolutionary multi-objective flexible job shop scheduling based on improved non dominated sorting cloud model. J. Mech. Eng. **50**(12), 198–205 (2014)
12. Chaoyong, Z., Xing, D., Wang, X., Li, X., Liu, Q.: Multi objective flexible job shop scheduling based on improved non dominated sorting genetic algorithm. J. Mech. Eng. **46**(11), 156–164 (2010)
13. Wei, W., Tan, J., Feng, Y., Zhang, R.: Research on multi-objective optimization methods for flexible job shop scheduling problems. Comput. Integr. Manufact. Syst. **15**(08), 1592–1598 (2009)
14. Dai, M., Tang, D., Giret, A., Salido, M.A.: Multi objective optimization for energy efficient flexible job shop scheduling problem with transportation constraints. Robot. Comput. Integr. Manufact. **59** (2019)
15. Shao, Z., Pi, D., Shao, W.: A multi objective discrete invasive weed optimization for multi objective blocking flow shop scheduling problem. Expert Syst. Appl. **113**(C) (2018)