



An improved adaptive NSGA-II with multi-population algorithm

Zhibiao Zhao¹ · Bin Liu¹ · Chunran Zhang¹ · Haoran Liu¹

Published online: 6 September 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

The NSGA-II algorithm uses a single population single crossover operator, which limits the search performance of the algorithm to a certain extent. This paper presents an improved version of the NSGA-II algorithm, named adaptive multi-population NSGA-II (AMP-NSGA-II) that divides the original population into multiple populations and assigns a different crossover operator to each subspecies. It introduces an excellent set of solutions (EXS), which can make the individuals in the EXS set close to the Pareto front and improve the convergence performance of the algorithm. And based on the analysis of the EXS set, the size of each subpopulation can be dynamically adjusted, which can improve the adaptability for different problems. Finally, the computation results on benchmark multi-objective problems show that the proposed AMP-NSGA-II algorithm is effective and is competitive to some state-of-the-art multi-objective evolutionary algorithms in the literature.

Keywords Multi-population · NSGA-II · Multiobjective · Adaptive

1 Introduction

In practical engineering applications, multi-objective optimization is a very important research topic [1]. We often need to use the best possible decision-making problem for multiple targets in a given feasible domain, but the improvement of these targets may come into conflict with each other, a set of objectives must be taken to neutral balance. Multi-objective genetic algorithms have good solvability for such problems [17, 18]. Non-dominated Sorting Genetic Algorithm 2 (NSGA-II) is one of the most representative genetic algorithms. NSGA-II was developed by Deb in 2000, which is NSGA improved non-inferior classification genetic algorithm [14], adopts a better accounting strategy and thus reduces the overall running time of the algorithm.

In recent years, many scholars have proposed improved algorithms for the NSGA-II in order to improve the diversity of the population and the convergence of the algorithm. Some researches improved the individual sequencing of

NSGA-II by ranking individuals in the population as $1 \sim m_r$, and setting the sequence of repeated individuals in the population to $m_r + 1$ to ensure the uniqueness of individuals in the sequence $1 \sim m_r$, improving the diversity of the population and avoiding the population into local optimum [7]. Some researches improved the crossover operation method of NSGA-II genetic algorithm, such as selecting crossover individuals by using the principle of adjacent maximum, that is, selecting the individuals with the largest distance among adjacent individuals for crossover operation, taking advantage of the hybridization to improve the population Distribution at the forefront of Pareto [16]. In some research, the cross-operation of NSGA-II is replaced by an orthogonal array and Taguchi method to optimize the algorithm performance and improve the convergence of the algorithm [11]. The theory of intermittent equilibrium points out that evolution and generation of new species cannot occur in the core area where a major group of species is located. In the core areas where the population is clustered, there is less pressure for biological survival, simple living environment and poor diversity. Therefore, it is extremely difficult to produce a system that can withstand harsh environments and has a strong adaptability of offspring individuals. Based on this, multi-population genetic algorithms have been proposed and are widely developed [10, 13]. Each subpopulation performs independent genetic operations, effectively improving the diversity of the population.

✉ Zhibiao Zhao
15533542910@163.com
Bin Liu
liubin@ysu.edu.cn

¹ School of Information Science and Engineering,
Yanshan University, Qinhuangdao 066004, Hebei, China

In order to further improve the population diversity and search performance of NSGA-II, this paper uses the idea of intermittent equilibrium to establish the genetic operation mode of multi-population and multi-crossover operator, to avoid the population from getting into local optimum. According to the contribution of sub-population to EXS solution set, *Logistic* model is used to adaptively adjust population. Combined with local search method, the algorithm has strong local optimization ability and improves the individual distribution on the Pareto front, and then proposes an adaptive multi-population non-dominated Sorting Genetic Algorithm II (AMP-NSGA-II). This paper describes the operation process of AMP-NSGA-II by introducing the establishment of multi-population.

2 Adaptive multi-population NSGA-II

2.1 Introducing multi-population and crossover operators

Multi-objective optimization usually does not have a solution that can achieve all the objectives simultaneously. Rather, it needs to obtain a Pareto optimal solution set whose optimal solution is theoretically optimal (also referred to as Pareto front-end of the target space) and should have the best possible approximation and uniformity. NSGA-II is an algorithm for solving this multi-objective optimization problem. It conducts the non-dominated sorting of the initial population, then selects, crosses and mutates to obtain a new population, merges the subpopulation with the parent population, and conducts a non-dominated sorting to obtain a non-dominated solution front. In previous NSGA-II, only one population is maintained during the evolution of the population. The disadvantage of this approach is that population diversity may be poor, because for some multi-objective optimization problems with many local optima, populations tend to converge to local optimal locations. This paper decomposes a single population of NSGA-II into multiple subpopulations, each of which is assigned a unique crossover operator. The blend crossover (denoted as $BLX - \alpha$) [5], the simulated binary crossover (denoted as SBX) [3], the simplex crossover (denoted as SPX) [15], and parent centric crossover (denoted as PCX) [2] operator are proved to be suitable for solving different problems of different ability. Therefore, it is reasonable to adjust the scale of each subpopulation adaptively by assigning different crossover operators to each subpopulation and designing an appropriate management strategy.

In this paper, the population is divided into four parts and assigned with above four different crossover operators, respectively. Different crossover operators have different

global search capabilities, different search methods, and different crossover populations. Parents of population I adopt blend crossover ($BLX - \alpha$) [5] operation. The parent individuals are $p^1 = (x_1^1, x_2^1, \dots, x_n^1)$ and $p^2 = (x_1^2, x_2^2, \dots, x_n^2)$ respectively, and the crossover formula is:

$$BLX - \alpha(p_1, p_2) : \begin{cases} y_j^1 = (1 - \beta)x_j^1 + \beta x_j^2 \\ y_j^2 = (1 - \beta)x_j^2 + \beta x_j^1 \end{cases} \quad (1)$$

where $p^2 = (x_1^2, x_2^2, \dots, x_n^2)$ and $\beta = (1 + 2\alpha)U(0, 1) - \alpha$. The individuals generated by the crossover operator have better diversity, improve the individual distribution of the population and have good ability to solve the segmentation function. The crossover operator has an advantage of generating diversity offspring, that allows GA to converge, diversity or adapt to changing fitness landscapes without incurring extra parameters or mechanisms. However, there are limitations to the multi-objective problem with strong correlation between variables, such as epistasis problem. Parents of population II adopt binary crossover (SBX) [3] operation, the parents individuals were $p^1 = (x_1^1, x_2^1, \dots, x_n^1)$ and $p^2 = (x_1^2, x_2^2, \dots, x_n^2)$, the formula for the crossover:

$$SBX(p_1, p_2) : \begin{cases} y_i^1 = 0.5[(1 - \beta)x_i^1 + (1 + \beta)x_i^2] \\ y_i^2 = 0.5[(1 + \beta)x_i^1 + (1 - \beta)x_i^2] \end{cases} \quad (2)$$

where $\beta(u) = \begin{cases} (2u)^{\frac{1}{nc+1}}, u \leq 0.5 \\ (2(1 - u))^{\frac{1}{nc+1}}, \text{others} \end{cases}$, and u is a uniformly distributed random number in interval $[0, 1]$. The crossover operator can cross two parents to obtain any child individuals and has good solving ability for multi-objective optimization problems with globally optimal upper and lower bounds with narrow global domains. Parents of population III adopt simplex crossover (SPX) [15] operations, and their parents were $p^1 = (x_1^1, x_2^1, \dots, x_n^1)$, $p^2 = (x_1^2, x_2^2, \dots, x_n^2)$ and $p^3 = (x_1^3, x_2^3, \dots, x_n^3)$ respectively. The crossover formula was:

$$SPX : \begin{cases} y_i^1 = (1 + \varepsilon)(x_i^1 - \bar{x}) \\ y_i^2 = (1 + \varepsilon)(x_i^2 - \bar{x}) \\ y_i^3 = (1 + \varepsilon)(x_i^3 - \bar{x}) \end{cases} \quad (3)$$

where \bar{x} denote the center of the selected individual. The crossover operator can balance the development of the population and explore the performance, so that the formation of the offspring of individuals from the coordinate system independent. Can perform well in multimodal functions and low-dimensional functions with three parents or high-dimensional functions with four parents. However, it is disadvantageous to solve the test function composed of strong correlation. Parents of population IV adopt the parent centric crossover (PCX) [2] operation and the parents are $p^1 = (x_1^1, x_2^1, \dots, x_n^1)$,

$p^2 = (x_1^2, x_2^2, \dots, x_n^2)$ and $p^3 = (x_1^3, x_2^3, \dots, x_n^3)$. The crossover formula is:

$$PCX : y^p = p^p + \omega_\zeta d^p + \sum_{i=1, i \neq p}^{n_u} \omega_\eta \bar{D} e^i \quad (4)$$

where $e^i = d^i / |d^i|$; $d^p = p^p - g$, g is the average vector of n_u parent individuals; D is the mean of the vertical distance to the vector d^p for the remaining $n_u - 1$ individuals. The crossover operator uses an adaptive method to cross-operate and The sub-individuals generated with the crossover operator are not close to the center of the selected parent but close to the parent.

The main process of genetic algorithm shown in Fig. 1, including multi-population genetic manipulation, updating EXS solution set and adjusting the size of the population adaptively and other parts.

In Fig. 1, $P_{i,t}$ is the t -generation parent population of i ; $Q_{i,t}$ is the t -generation sub-population of i . The four sub-populations assigned four different crossover operators. In multi-populations genetic manipulation phase, each subpopulation uses NSGA-II algorithm to perform non-inferior sorting and crowd-distance sorting operations respectively. F_j is the set of individuals with rank j in population $R_{i,t} = P_{i,t} \cup Q_{i,t}$; and $P'_{i,t+1}$ is the $t + 1$ -generation parent population dynamically adjusted by population i . From a global perspective, the application of cross-operation can find some good individuals. In the EXS solution collection update phase, the EXS is updated according to the fitness of individuals, and the dominant individuals in the subpopulations are collected in the EXS solution set. Through the proposed local search method, the crossover operator of the dominant population is used to further increase the search ability of the algorithm and improve the Pareto frontier EXS solution set distribution. In order to get the population closer to the Pareto frontier, individual selection of crossover operations are randomly selected from the population $P_{i,t} (i = 1, 2, 3, 4), Q_{i,t} (i =$

1, 2, 3, 4) and the EXS set, respectively, to ensure that new individuals crossing the generation continually approximate the optimal solution set. In adjusting population size phase, the contribution of different populations to the EXS is counted, and the population with a large contribution is considered to be more suitable for solving the target problem. Therefore, the number of the individuals is appropriately increased, otherwise it is decreased. The crossover operator is:

$$Crossoveroperator(k) \begin{cases} BLX - \alpha(p_1, p_2), & k = 1 \\ SBX(p_1, p_2), & k = 2 \\ SPX(p_1, p_2, p_3), & k = 3 \\ PCX(p_1, p_2, p_3), & k = 4 \end{cases} \quad (5)$$

Where k is serial number of four populations. Population $P_{i,t} (i = 1, 2, 3, 4)$ were crossed and mutated to obtain population $Q_{i,t}$, and population $R_{i,t} = P_{i,t} \cup Q_{i,t}$ was ranked non-inferior and crowded by distance $P_{1,t+1}, P_{2,t+1}, P_{3,t+1}, P_{4,t+1}$. In order to improve the local search ability of the population, this algorithm preserves the mutation operator of the original NSGA-II to adjust some gene values in individual chromosome strings. From a local point of view, the individual is more approximate to the optimal solution set and the genetic algorithm local search Sufficient ability to maintain the diversity of the population, to prevent premature phenomenon. When the algorithm introduces the above crossover operator, the various subpopulation can make up each other and coordinate with each other so that the algorithm can find effective crossover operator for different multi-objective optimization problems, cross better individuals and enhance the adaptation of the algorithm Sexual and population diversity.

The NP individuals are divided into 4 populations $P_{1,1}, P_{2,1}, P_{3,1}$ and $P_{4,1}$, and the initial size of each population is $NP/4$, respectively. The population was initialized using the method in [8]. From the initial four sub-populations $P_{i,1} (i = 1, 2, 3, 4)$, four progeny populations $Q_{i,1}$ are generated through the selection, crossover and mutation operation of the binary tournament respectively. By sorting the $R_{i,1} = P_{i,1} \cup Q_{i,1}$ populations non-inferiorly, Rank1 individuals b_i in $R_{i,1}$ populations are added to the EXS solution sets so that the initial EXS solution has a much denser distribution on the Pareto frontier. EXS solution set initialization process is shown in Fig. 2. If $|EXS| = \sum_{i=1}^4 b_i > N_{EXS}$, the non-inferior ordering and crowding distance ranking of the EXS solution sets are performed and remove the bad solution from the EXS solution so that $|EXS| = N_{EXS}$. Or if $|EXS| = \sum_{i=1}^4 b_i < N_{EXS}$, it will not be processed. When the EXS solution set is updated, the EXS solution sets the number of individual to N_{EXS} . The process of updating is to replace the inferior solution in

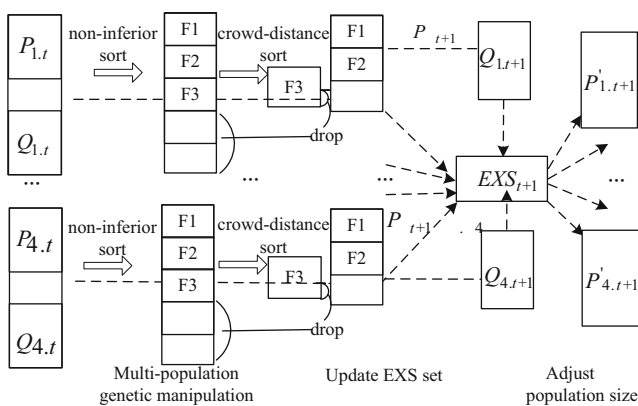


Fig. 1 The main procedure of the AMP-NSGA-II

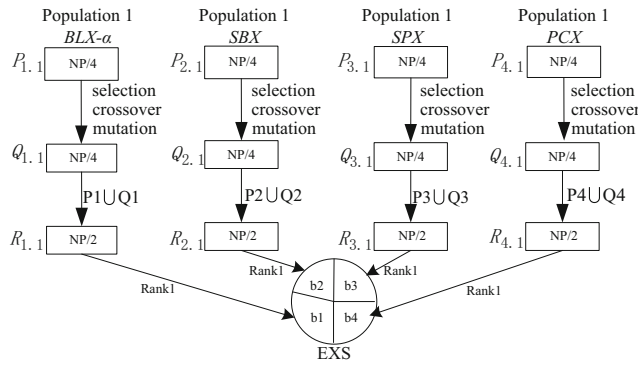


Fig. 2 The excellent solution set initial process

the population of the dominant individual in the population, so that the EXS collection is constantly approaching Pareto optimal solution set. The update steps for the EXS collection are as follows:

- step 1 If the solution in the population $i(i = 1, 2, 3, 4)$ is better than the solution in the EXS solution set, the solution in the EXS solution set is removed and the dominant individuals corresponding to the population i are added to the solution set.
- step 3 If the solution in the population is worse than the solution in the EXS solution, do not deal with it;
- step 3 If the number of individuals in the EXS solution set exceeds the set value N_{EXS} , the bad solution is removed according to the non-inferior sorting and crowding distance sorting method of NSGA-II.

2.2 Local search algorithm

To improve the individual distribution of EXS solution sets, the algorithm process is divided into two stages. The first phase is EXS-pop-Update, which updates the EXS solution via populations $Q_{1,t+1}, Q_{2,t+1}, Q_{3,t+1}$ and $Q_{4,t+1}$. This method can search and exploit the population $P_{1,t+1}, P_{2,t+1}, P_{3,t+1}$ and $P_{4,t+1}$ well and find out more excellent individuals to improve the distribution of EXS solution sets on the Pareto frontier. The second phase is EXS-self-Update, which is a set of self-updating for the EXS solution. Through the EXS solution of the inter-personal cross-operation, local search the EXS solution, mining EXS solution sets the potential of the optimal solution individual. Based on the analysis of the update process of EXS solution by $P_{i,t+1}(i = 1, 2, 3, 4)$, the contribution of each subpopulation to EXS solution is calculated and the crossover operator in this phase is determined. The specific process shown in Table 1, where j represents the serial number of the four populations, N_{EXS} represents the number of individuals in the EXS. The individuals with the greatest contribution and the individuals in EXS are randomly selected to cross-mutate and generate sub-individuals, and the disadvantaged individuals in the EXS are replaced by the dominant individuals.

2.3 Adaptive adjustment of population size

The distribution of crossover operators in each subpopulation leads to different distribution of individuals in each

Table 1 The excellent solution set updated by itself

1	The estimated contribution of $P_{i,t+1}(i = 1, 2, 3, 4)$ to the update of EXS solution is c_i ;
2	Determine the population that contributes most to the EXS $j: \left\{ j \mid c_j = \max_{i=1,2,3,4} \{c_i\} \right\}$;
3	Calculate the crossover operator for the EXS-self-Update operation $Crossoveroperator(j)$;
4	If $j \leq 2$:
5	For $k: = 1$ to $(N_{EXS}/2)$:
6	Two crossover individuals p_1 and p_2 are randomly selected from the EXS solution set;
7	For individuals p_1 and p_2 , crossover operations are performed with $Crossoveroperator(j)$ crossover operators to generate offspring individuals q_1 and q_2 ;
8	Update the EXS solution set with individuals q_1 and q_2 ;
9	End for
10	End if
11	If $j \geq 3$:
12	For $k: = 1$ to $(N_{EXS}/3)$:
13	Three crossover individuals, $p_1, p_2,$ and p_3 are randomly selected from the EXS solution set
14	For individuals p_1, p_2 and p_3 , crossover operations are performed with $Crossoveroperator(j)$ crossover operators to generate offspring individuals q_1, q_2 and q_3 ;
15	Update the EXS solution set with individuals q_1, q_2 and q_3 ;
16	End for
17	End if

population when solving different multi-objective problems. Sub-population $P_{i,t+1}$ ($i = 1, 2, 3, 4$) differ in their contribution to the EXS solution set during the updating of the EXS solution set. Under the condition of keeping the total number of individuals NP unchanged, by adjusting the size of each population, the density of individuals in front of Pareto is increased, the number of population with fewer density in Pareto frontier is reduced, and each subpopulation can be realized with different adaptive optimization problems, the optimal solution set can be found as soon as possible for different multi-objective optimization problems, which effectively improves the adaptability of the population. Select two populations with large contribution to increase the individual number of the population. With the gradual increase of population size, the individual number of each population can not exceed the maximum population ps_{max} because of limited resources and the number of increase individuals variable n_{inc}^i should be gradually reduced. Select two small contribution populations to reduce the number of selected individuals, with the gradual reduction of population size, it should ensure that the number of individuals in the sub-population not less than the population minimum ps_{min} . Using the discrete time *Logistic* model to adjust the number of individual population, the number of increase individuals and reduce individuals are calculated by (6) and (7) respectively:

$$n_{inc}^i = \beta_1 \cdot ps_g^i \cdot \left(1 - \frac{ps_g^i}{ps_{max}}\right) \tag{6}$$

$$n_{dec}^i = \beta_2 \cdot ps_g^i \cdot \left(1 - \frac{ps_g^i}{ps_{min}}\right) \tag{7}$$

Where ps_g^i denotes the size of the g -th population of population i , $\left(1 - \frac{ps_g^i}{ps_{max}}\right)$ and $\left(1 - \frac{ps_g^i}{ps_{min}}\right)$ are the density dependences, ps_{max} and ps_{min} are the maximum and minimum number of individuals we have set. The *Logistic* models show density dependence, meaning the per capita population growth rates decline as the population density increases. In addition, the negative density dependence $\left(1 - \frac{ps_g^i}{ps_{min}}\right)$ gradually decreases the decreasing individuals to ps_{min} . The two selected subpopulations with large (or small) contributions have some differences in updating the number of EXS solutions. Therefore, this paper proposes the following adjustment strategy:

The number of individuals updating the four population-based EXS solution sets is c_1^i , c_2^j , c_3^k , and c_4^l , where i, j, k, l indicates the population label, here c_1^i means the contribution of population i to update the EXS solution set and the subscript 1 means it is the largest among the four

populations. The size of sub-population i, j, k, l are denote by (8)-(11) respectively.

$$Sub_i = ps_g^i + \frac{c_1^i + c_4^l}{\sum_{t=1}^4 c_t} \cdot n_{inc}^i \tag{8}$$

$$Sub_j = ps_g^j + \frac{(c_2^j + c_3^k) / 2}{\sum_{t=1}^4 c_t} \cdot n_{inc}^j \tag{9}$$

$$Sub_k = ps_g^k + \frac{(c_2^j + c_3^k) / 2}{\sum_{t=1}^4 c_t} \cdot n_{dec}^k \tag{10}$$

$$Sub_l = ps_g^l + \frac{c_1^i + c_4^l}{\sum_{t=1}^4 c_t} \cdot n_{dec}^l \tag{11}$$

To ensure that the number of individuals in each subpopulation is not less than ps_{min} and not large than ps_{max} , constraint the number of individuals Sub_d with (12). Properly reduce/increase the individuals of the largest/smallest subpopulations, leaving the total number of individuals unchanged.

$$Sub_d = \begin{cases} ps_{min}, & Sub_d < ps_{min} \ (d = 1, 2, 3, 4) \\ Sub_d, & others \\ ps_{max}, & Sub_d > ps_{max} \ (d = 1, 2, 3, 4) \end{cases} \tag{12}$$

After determining the size of each subpopulation Sub_d ($d = 1, 2, 3, 4$), non-destructive sorting of each population, and then proceed as follows:

- Case 1 If the number of population d Sub_d is smaller than the number of original population, delete it from the worst one in the population until the number of population reaches Sub_d .
- Case 2 If the number of population d Sub_d is greater than the number of original population, do:
 - step 1 Pick an individual randomly from the population d ;
 - step 2 Pick an individual (or two individuals randomly from the EXS solution set if the population d is assigned an SPX or PCX crossover operator);
 - step 3 For the two individuals (or three) obtained, new solutions are added to the population d by cross-operation of crossover operators assigned to the population;
 - step 4 If the number of population d reaches Sub_d , stop. Otherwise, skip to step1.

3 Experimentation

This section is devoted to presenting the experiments performed in this work. The performance of the algorithm is analyzed using 12 standard test problems, including the dual-objective test problems ZDT [19]: ZDT1, ZDT2, ZDT3, ZDT4, ZDT6 and three-target test problems DTLZ [4]: DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5, DTLZ6, DTLZ7. *GD* (General Distance) to measure whether the updated EXS solution set converges to the true Pareto optimal solution set. *GD* represents the distance between the obtained EXS solution set and the true Pareto optimal solution set, the smaller its value is, The optimal solution set converges to the true Pareto optimal solution set. It can be calculated by (13):

$$GD = \sqrt{\frac{\sum_{i=1}^{|EXS|} d_i^2}{|EXS|}} \tag{13}$$

where $|EXS|$ is the number of individuals in the EXS population; d_i is the Euclidean distance between the nearest individuals in individuals $i \in EXS$ and P^* in the target space. I_h (Hypervolume) index evaluation method is a solution set comprehensive quality evaluation method, which can simultaneously evaluate the convergence, uniformity and universality of the solution set. The larger the value of I_h , the comprehensive performance of the optimal solution set obtained The better. In this paper, the objective

function is normalized in the range of [0,1]. The standardization formula is:

$$f'_j(x) = (f_j(x) - f_j^{\min}) / (f_j^{\max} - f_j^{\min}) \tag{14}$$

where $f_j(x)$ is the j -th objective function value of solution x ; f_j^{\min} (f_j^{\max}) is the minimum value (maximum value) of the 6th target value in the solution set obtained by genetic algorithm, and the reference point is set as (1.0, 1.0).

According to the flow of the algorithm, it can be seen that the updating process of the EXS set is to replace the more dominant individuals in the four populations into the EXS set. Therefore, the result of $EXS(t+1)$ ($t+1$ means the $t+1$ generation of evolution) is definitely better than $EXS(t)$, that is, there is no individual in $EXS(t)$ can dominate any individual in $EXS(t+1)$, so the algorithm is definitely convergent. Figure 3 shows EXS individuals initial states and its evolution process at generations 50, 200, 500 when the algorithm solves the ZDT1 problem.

Table 2 shows the comparison between the convergence index (*GD*) and the comprehensive performance index (I_h) of AMP-NSGA-II algorithm and NSGA-II algorithm on 12 test problems and verify whether there is a significant difference between the two algorithms in *GD* and I_h through independent-samples t-test by SPSS (Statistical Product and Service Solutions) software. The resulting values are the median (x_m) and the interquartile range (*IRQ*) of the results obtained by running 100 times. Better values are in bold style. As is shown, the AMP-NSGA-II algorithm obtained 12 superior *GD* values out of 12

Fig. 3 AMP-NSGA-II convergence process

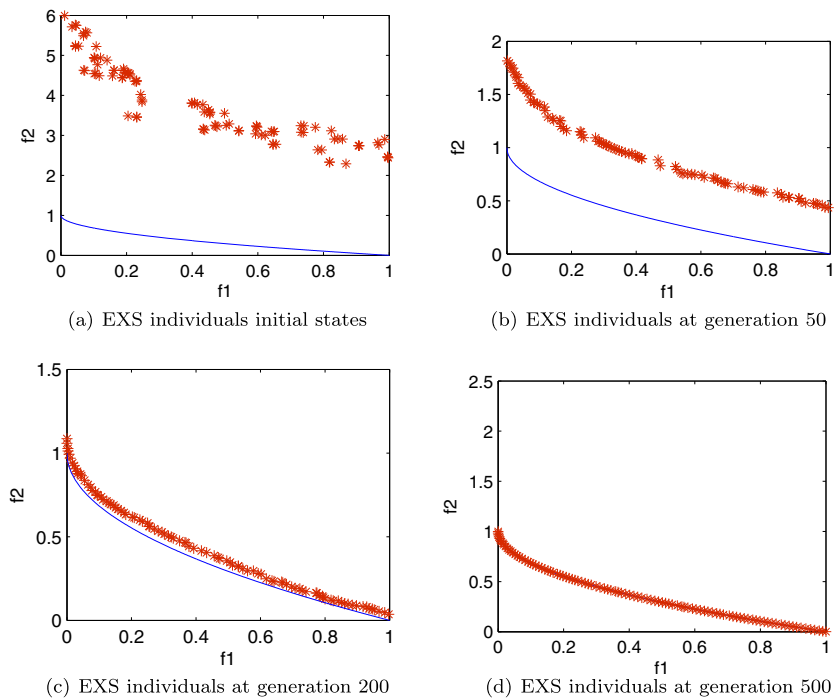


Table 2 Experimental results comparison between AMP-NSGA-II and NSGA-II algorithm

Problems	GD				SD	I_h				SD
	AMP-NSGA-II		NSGA-II			AMP-NSGA-II		NSGA-II		
	x_m	IRQ	x_m	IRQ		x_m	IRQ	x_m	IRQ	
ZDT1	1.410e-04	3.196e-05	4.586e-04	5.976e-05	Y	6.619e-01	4.164e-05	6.551e-01	9.119e-04	Y
ZDT2	4.791e-05	4.814e-06	5.225e-04	9.523e-05	Y	3.287e-01	4.087e-05	3.206e-01	1.362e-03	Y
ZDT3	3.348e-05	3.048e-06	1.799e-04	3.695e-05	Y	5.876e-01	2.071e-05	5.835e-01	8.462e-04	Y
ZDT4	1.034e-04	4.488e-05	1.606e-04	5.000e-05	Y	6.620e-01	3.278e-05	6.595e-01	9.191e-04	Y
ZDT6	8.413e-03	0.715e-03	9.627e-02	1.357e-02	Y	4.013e-01	6.867e-05	6.587e-01	0.000e+00	Y
DTLZ1	5.654e-04	3.060e-05	7.416e-04	2.683e-04	Y	7.662e-01	6.529e-03	7.623e-01	7.801e-03	Y
DTLZ2	6.399e-04	1.681e-04	1.366e-03	2.414e-04	Y	3.867e-01	6.033e-03	3.754e-01	9.428e-03	Y
DTLZ3	1.102e-03	9.371e-05	4.420e-03	1.156e-02	Y	3.872e-01	6.354e-03	3.342e-01	3.843e-02	Y
DTLZ4	4.628e-03	3.684e-04	5.137e-03	3.862e-04	Y	3.839e-01	5.823e-03	3.759e-01	8.662e-03	Y
DTLZ5	2.482e-04	3.115e-05	3.684e-04	7.521e-05	Y	9.400e-02	4.973e-05	9.288e-02	2.400e-04	Y
DTLZ6	5.766e-04	3.630e-05	6.314e-04	2.361e-03	Y	9.494e-02	3.682e-05	9.355e-02	1.857e-02	Y
DTLZ7	2.062e-03	8.198e-04	2.665e-03	7.560e-04	Y	2.940e-01	4.417e-03	2.841e-01	7.348e-03	Y

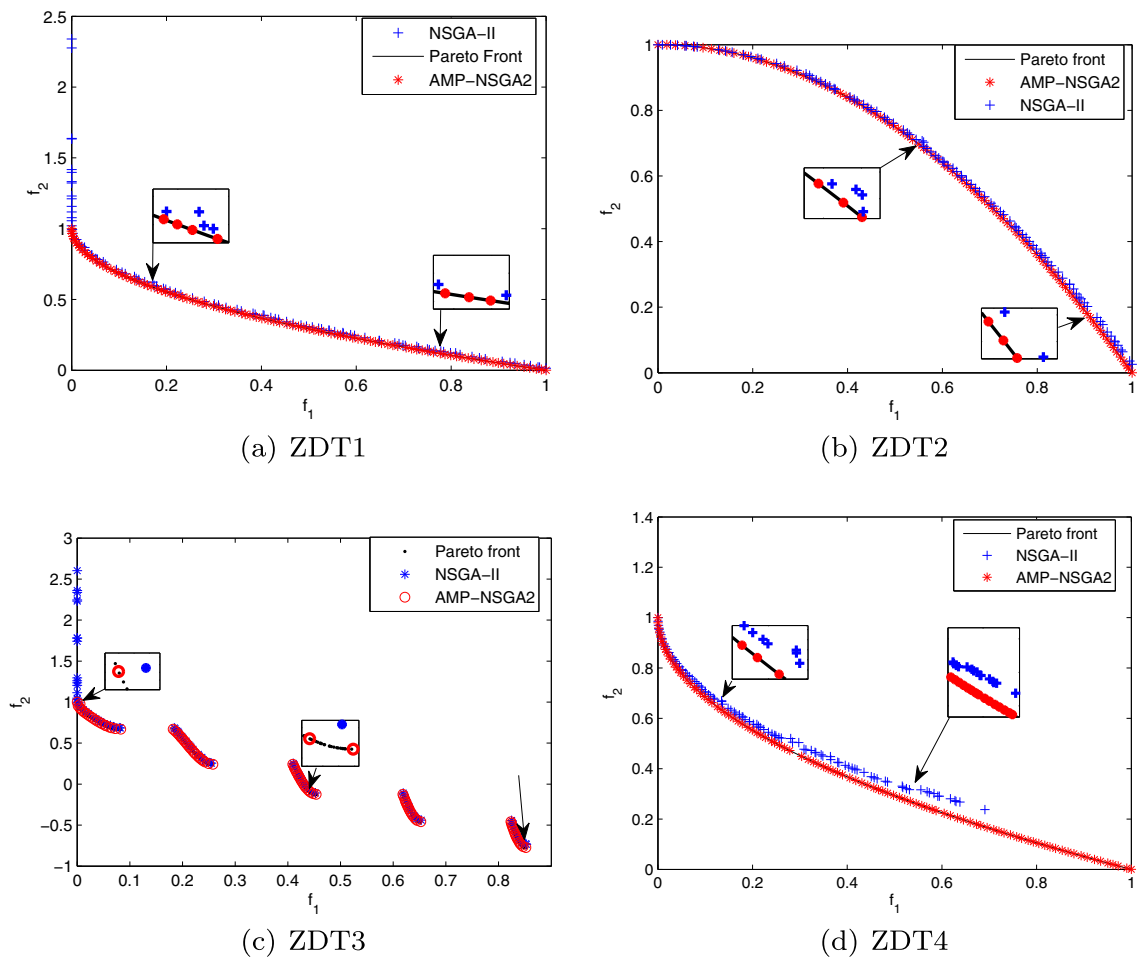


Fig. 4 Comparison of AMP-NSGA-II and NSGA-II experimental results

test problems and obtained 11 superior I_h values out of 12 test problems, except for the ZDT6. This shows that the algorithm has a better adaptability than NSGA-II. Independent-samples t-test showed significant differences (SD) between the two genetic algorithms. Therefore, the proposed AMP-NSGA-II genetic algorithm is superior to the NSGA-II genetic algorithm in terms of convergence and overall performance. A graphical overview of 2 bi-objective problems is given in Fig. 4. As is shown, on the ZDT1 and ZDT3 test problem, a large number of individuals in NSGA-II fall into $f_1 = 0$ and the individuals are unevenly distributed, indicating that the NSGA-II algorithm is easy to fall into local optimum when solving the test problem and individuals in AMP-NSGA-II were uniformly and accurately distributed on the Pareto front. On the ZDT2 problem, AMP-NSGA-II distributed more evenly than NSGA-II, with less deviation from the Pareto front. On the ZDT4 problem, NSGA-II individuals are concentrated in the range of $[0, 0.7]$. As f_1 increases, individuals gradually diverge from the Pareto front and is divergent. The Pareto front obtained by NSGA-II and AMP-NSGA-II of the three-objective problem DTLZ4 is shown in Fig. 5. As is shown, in the results of NSGA-II, the distribution of

individuals is disorganized, with obvious overlap between individuals. In contrast, the distribution of individuals in AMP-NSGA-II is closer to the Pareto optimal solution and the individual diversity is better. Therefore AMP-NSGA-II distributes more evenly than NSGA-II individuals and has better convergence performance, which proves that the improved method in this paper is effective.

In order to test the competitiveness of proposed AMP-NSGA-II, it was compared with some state-of-the-art multi-objective evolutionary algorithms (MOEAs) such as AbYSS [8], GWASFGA [12], SMPSO [9] and DMOPSO [6]. Source codes of these algorithms are available through the website: <https://jmetal.github.io/jMetal/> and their parameters are set as suggested by the original. Tables 3 and 4 show the performance comparison over GD and I_h performance on 12 test problem. As is shown in Table 3, in terms of GD metric in 12 test problems, AMP-NSGA-II obtained 4 best result, including 1 best result of the bi-objective problem and 3 best results of the three-objective problems. It is worth mentioning that DMOPSO obtained 7 best result in GD metric, but in terms of I_h metric, as is shown in Table 4, DMOPSO obtained none best result, and AMP-NSGA-II algorithm obtained 5 best result.

Fig. 5 Comparison of AMP-NSGA-II and NSGA-II experimental results in DTLZ4 problem

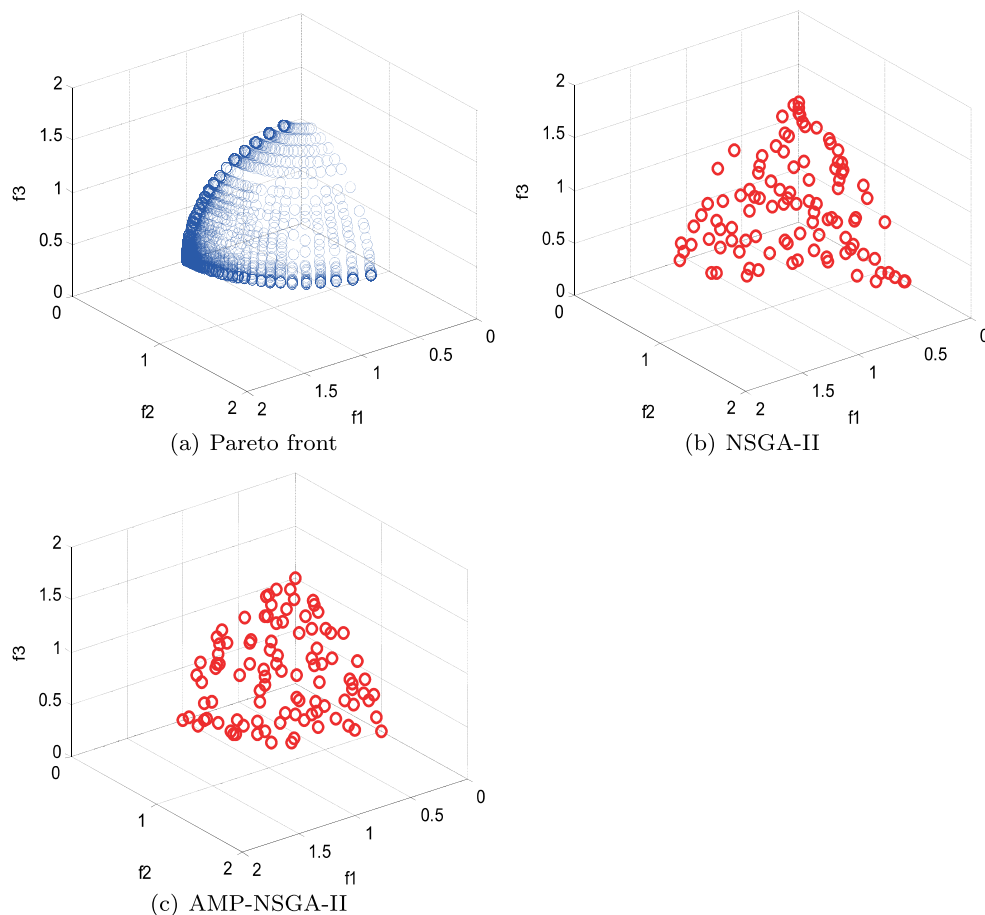


Fig. 6 Comparison experimental results in DTLZ7 problem

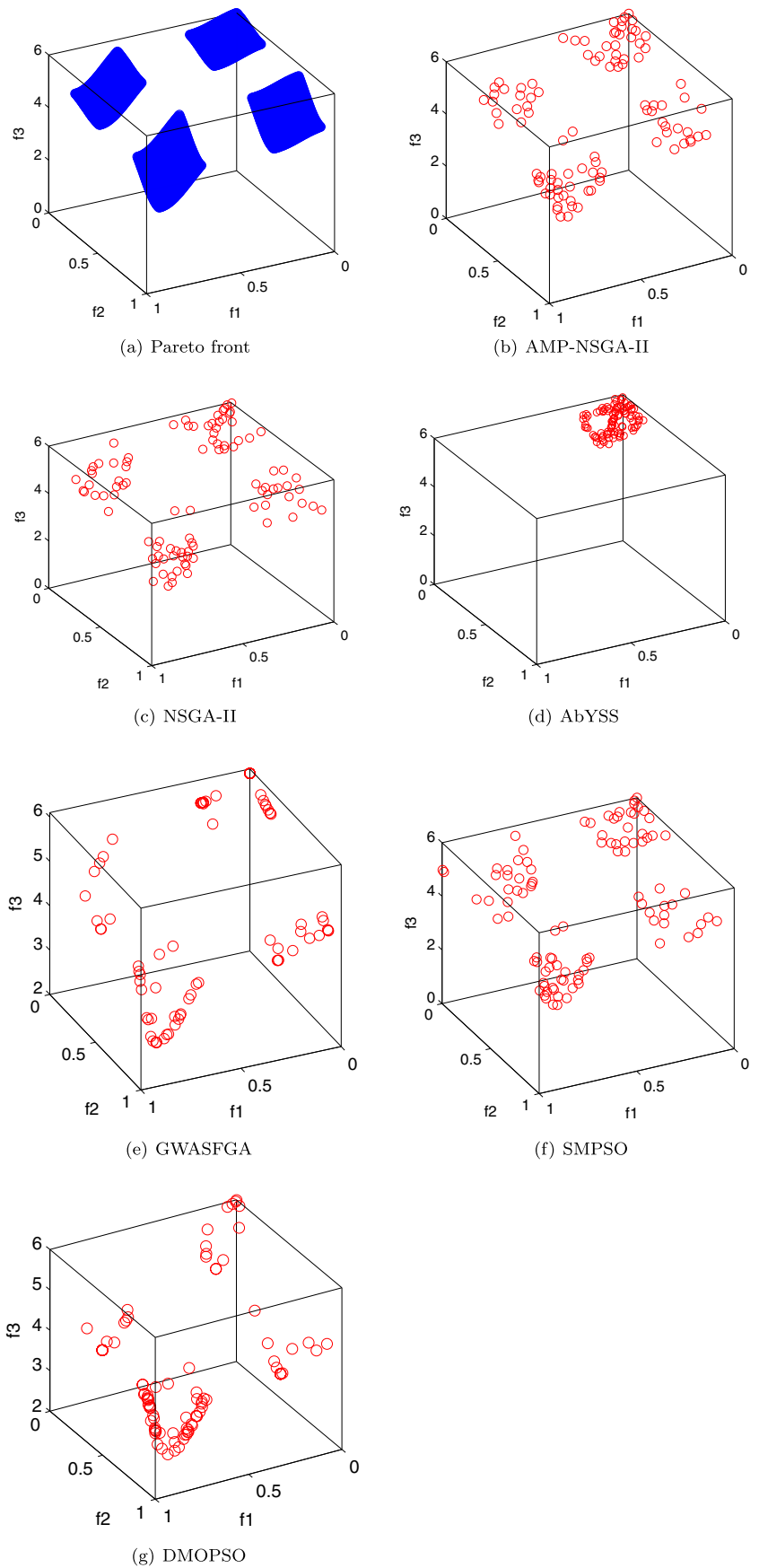


Table 3 Experimental results in term of GD between AMP-NSGA-II and NSGA-II algorithm

GD	AMP-NSGA-II		AbYSS		GWASFGA		SMP SO		DMOP SO	
	x_m	IRQ	x_m	IRQ	x_m	IRQ	x_m	IRQ	x_m	IRQ
ZDT1	1.410e-04	3.20e-05	1.525e-04	2.14e-05	2.624e-04	3.19e-05	1.088e-04	4.29e-05	9.784e-05	2.27e-05
ZDT2	4.791e-05	4.81e-06	5.223e-05	1.15e-05	9.461e-05	2.76e-06	4.740e-05	2.68e-06	4.658e-05	1.69e-06
ZDT3	3.348e-05	3.05e-06	3.393e-05	3.42e-06	2.467e-04	2.15e-06	3.609e-05	2.72e-06	1.609e-04	3.43e-06
ZDT4	1.034e-04	4.49e-05	2.316e-04	1.02e-04	1.408e-03	1.33e-05	1.212e-04	4.16e-05	9.473e-05	2.68e-06
ZDT6	8.413e-03	0.72e-03	7.475e-05	8.61e-06	1.037e-03	8.12e-06	5.031e-05	8.38e-06	3.527e-05	7.62e-06
DTLZ1	5.654e-04	3.06e-05	6.034e-04	8.94e-05	6.295e-04	2.61e-04	5.191e-03	1.38e-03	4.654e-04	7.76e-06
DTLZ2	6.399e-04	1.68e-04	6.841e-04	3.74e-05	8.780e-04	1.60e-04	3.393e-03	6.47e-04	8.730e-04	3.43e-05
DTLZ3	1.102e-03	9.37e-05	4.091e-03	1.02e-04	9.826e-03	3.12e-03	3.360e-03	1.16e-03	2.171e-02	7.36e-03
DTLZ4	4.628e-03	3.68e-04	4.881e-03	3.15e-04	6.283e-03	2.49e-04	5.745e-03	5.06e-04	8.716e-03	4.61e-04
DTLZ5	2.482e-04	3.16e-05	2.539e-04	3.14e-05	2.581e-04	6.27e-05	2.611e-04	3.35e-05	2.344e-04	3.31e-05
DTLZ6	5.766e-04	3.63e-05	9.373e-03	7.67e-03	6.951e-02	7.12e-05	5.664e-04	3.34e-05	5.347e-04	3.29e-05
DTLZ7	2.062e-03	8.20e-04	1.287e-03	1.06e-03	2.601e-03	5.11e-04	4.675e-03	1.07e-03	2.725e-03	1.04e-03

GWASF GA obtained 5 best I_h value, but the convergence of the algorithm is limited and didn't get a the superior GD value. SMP SO and AbYSS obtained 1 best I_h result and 1 best GD result, respectively, that show a little inferior performance than AMP-NSGA-II, GWASF GA and DMOP SO. Pareto front for each algorithm with the best GD metric for problem DTLZ7 is plotted in Fig. 6. As is shown, though the GD result obtained by AbYSS is the best one, the distribution of individuals are trapped in the first block of the true Pareto front, which is not in line with our expectation. The other five algorithms have similar performance in Fig. 6, but the individuals obtained by DMOP SO have a high overlap in the fourth

part of the pareto front and GWASF GA (80 individuals) is more biased toward the boundary of the pareto front. The individuals obtained by AMP-NSGA-II distributes visually more evenly than NSGA-II and SMP SO. In pairwise comparison from Tables 3 and 4, our AMP-NSGA-II obtained 10 better GD results and 8 not less I_h results than the AbYSS, 11 better GD results and 7 not less I_h results than GWASF GA, 8 better GD results and 11 not less I_h results than SMP SO, 5 better GD results, 11 not less I_h results than DMOP SO. Through the comparison with other four MOEAs, AMP-NSGA-II can be found to have superior results for most of the test problems.

Table 4 Experimental results in term of I_h between AMP-NSGA-II and NSGA-II algorithm

I_h	AMP-NSGA-II		AbYSS		GWASF GA		SMP SO		DMOP SO	
	x_m	IRQ	x_m	IRQ	x_m	IRQ	x_m	IRQ	x_m	IRQ
ZDT1	6.619e-01	4.16e-05	6.620e-01	7.17e-05	6.626e-01	4.59e-05	6.620e-01	4.58e-05	6.615e-01	4.69e-05
ZDT2	3.287e-01	4.09e-05	3.287e-01	7.60e-05	3.287e-01	2.42e-06	3.287e-01	3.72e-05	3.284e-01	3.73e-05
ZDT3	5.876e-01	2.07e-05	5.876e-01	2.40e-05	5.158e-01	2.71e-06	5.875e-01	7.42e-05	5.137e-01	8.77e-05
ZDT4	6.620e-01	3.28e-05	6.599e-01	2.32e-03	6.625e-01	1.03e-04	6.618e-01	1.00e-04	6.609e-01	8.99e-05
ZDT6	4.013e-01	6.87e-05	4.006e-01	1.43e-04	4.019e-01	2.17e-04	4.013e-01	6.40e-05	4.013e-01	4.38e-05
DTLZ1	7.662e-01	6.53e-03	7.612e-01	8.17e-03	7.384e-01	1.36e-03	7.401e-01	9.50e-03	7.409e-01	6.06e-03
DTLZ2	3.867e-01	6.03e-03	3.847e-01	7.12e-03	3.924e-01	4.58e-04	3.548e-01	7.30e-03	3.678e-01	8.22e-03
DTLZ3	3.872e-01	6.35e-03	3.417e-01	4.18e-02	1.872e-01	4.81e-04	3.561e-01	1.45e-02	3.409e-01	9.67e-03
DTLZ4	3.839e-01	5.82e-03	3.867e-01	5.25e-03	3.822e-01	6.22e-05	3.632e-01	8.70e-03	2.729e-01	7.54e-03
DTLZ5	9.400e-02	4.97e-05	9.401e-02	3.00e-05	9.425e-02	8.04e-06	9.390e-02	6.70e-05	9.203e-02	4.67e-05
DTLZ6	9.494e-02	3.68e-05	4.883e-02	1.90e-02	9.472e-02	2.44e-03	9.490e-02	4.58e-05	9.312e-02	4.75e-05
DTLZ7	2.940e-01	4.42e-03	2.604e-01	0.00e+00	2.441e-01	1.17e-03	2.764e-01	6.20e-03	2.583e-01	4.84e-03

4 Conclusion

In this paper, a multi-objective optimization algorithm for multi-population and multi-crossover operator mode is constructed. Single population in NSGA-II is divided into four subpopulations to improve the diversity of the population. According to the different contribution of sub-population to EXS solution set, *Logistic* model is used to realize adaptive adjustment of sub-population. The update of EXS solution set is divided into pop-Update and self-Update two stages, thus the local search program of the algorithm is given. The paper gives a scheme to adjust the size of various groups, so that the population quantity is related to the contribution of EXS solution set. The proposed algorithm is tested by 12 standard test functions, the experimental results show that:

- 1) The comparison with the original NSGA-II shows that the improvement of the proposed strategy is positive and the t-test shows that the performance of the algorithm is significantly different and has been qualitatively improved.
- 2) The addition of multiple populations increases the diversity of the population and makes the distribution of individuals more even and overcome local optimality effectively.
- 3) The adaptive scheme combined with local search algorithm allocate more evolution chances to certain sub-populations with more appropriate crossover operators, making the algorithm guarantee a robust performance for different kinds of multi-objective problems.
- 4) Under the combination of multiple partial changes, the performance of the proposed algorithm is comparative or superior to some state-of-the-art MOEAs for the ZDT and DTLZ series of problems.

References

1. Cai Y, Wang J (2015) Differential evolution with hybrid linkage crossover. *Inf Sci* 320:244–287
2. Deb K, Anand A, Joshi D (2002a) A computationally efficient evolutionary algorithm for real-parameter optimization. *Evol Comput* 10(4):371–395
3. Deb K, Pratap A, Agarwal S, Meyarivan T (2002b) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
4. Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multiobjective optimization. In: Abraham A, Jain L, Goldberg R (eds) *Evolutionary multiobjective optimization*. Advanced information and knowledge processing. Springer, London, pp 105–145. <https://doi.org/10.1007/1-84628-137-76>
5. Eshelman LJ, Schaffer JD (1993) Real-coded genetic algorithms and interval-schemata. In: *Foundations of genetic algorithms*, vol 2. Elsevier, pp 187–202
6. Lee KB, Kim JH (2014) DMOPSO: Dual multi-objective particle swarm optimization. In: *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, pp 3096–3102
7. Li AD, He Z, Zhang Y (2016) Bi-objective variable selection for key quality characteristics selection based on a modified NSGA-II and the ideal point method. *Comput Ind* 82:95–103
8. Nebro AJ, Luna F, Alba E, Dorronsoro B, Durillo JJ, Beham A (2008) AbYSS: Adapting scatter search to multiobjective optimization. *IEEE Trans Evol Comput* 12(4):439–457
9. Nebro AJ, Durillo JJ, Garcia-Nieto J, Coello CC, Luna F, Alba E (2009) SMPSO: A new pso-based metaheuristic for multi-objective optimization. In: *2009. mcdm'09. ieee symposium on Computational intelligence in multi-criteria decision-making*. IEEE, pp 66–73
10. Pourvaziri H, Naderi B (2014) A hybrid multi-population genetic algorithm for the dynamic facility layout problem. *Appl Soft Comput* 24:457–469
11. Qiao S, Dai X, Liu Z, Huang J, Zhu G (2012) Improving the optimization performance of NSGA-II algorithm by experiment design methods. In: *2012 IEEE International Conference on Computational intelligence for measurement systems and applications (CIMSA)*. IEEE, pp 82–85
12. Saborido R, Ruiz AB, Luque M (2017) Global WASF-GA: an evolutionary algorithm in multiobjective optimization to approximate the whole pareto optimal front. *Evol Comput* 25(2):309–349
13. Toledo CFM, Franca PM, Morabito R, Kimms A (2009) Multi-population genetic algorithm to solve the synchronized and integrated two-level lot sizing and scheduling problem. *Int J Prod Res* 47(11):3097–3119
14. Tran KD (2009) An improved non-dominated sorting genetic algorithm-ii (ANSGA-II) with adaptable parameters. *Int J Intell Syst Technol Appl* 7(4):347–369
15. Tsutsui S, Yamamura M, Higuchi T (1999) Multi-parent recombination with simplex crossover in real coded genetic algorithms. In: *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation*. Morgan Kaufmann Publishers Inc., vol 1, pp 657–664
16. Yijie S, Gongzhang S (2008) Improved NSGA-II multi-objective genetic algorithm based on hybridization-encouraged mechanism. *Chin J Aeronaut* 21(6):540–549
17. Zhong YB, Xiang Y, Liu HL (2014) A multi-objective artificial bee colony algorithm based on division of the searching space. *Appl Intell* 41(4):987–1011
18. Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang Q (2011) Multiobjective evolutionary algorithms: a survey of the state of the art. *Swarm Evol Comput* 1(1):32–49
19. Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol Comput* 8(2):173–195



Zhibiao Zhao received M.S. degree in School of Electrical Engineering from Yanshan University, China, in 2015. He has been worked toward the Ph.D. degree in the School of Information Science and Engineering, Yanshan University since 2015. His current research interests include multi-objective optimization and machine learning.



Chunran Zhang received M.S. degree in School of Electrical Engineering from Yanshan University, China in 2018. His research interests include artificial intelligence and multi-objective optimization.



Bin Liu is currently a professor and Ph.D. supervisor at School of Information Science and Engineering, Yanshan University, China. His research interests include artificial intelligence and multi-objective optimization.



Haoran Liu received Ph.D. degree in control engineering from INSA de Rouen, France. Now he works as professor and Ph.D. supervisor in School of Information Science and Engineering, Yanshan University, China. His research interests include multi-objective optimization, data processing.