A Construction Algorithm of Pareto Optimal Solution Set Based on Queue Principle

Chao Yang, Qin Wu

Institute of Information Technology Engineering, Tianjin University of Technology and Education, Tianjin 300222, China.

Abstract

Aiming at the time and efficiency of constructing the Pareto optimal solution set for multi-objective decision-making problems, a construction algorithm of Pareto optimal solution set based on queue principle is proposed, and the nature of the non-dominated relationship and the correlation of the construction of the Pareto optimal solution set are given Define and prove the theorem. The time complexity of the proposed algorithm is (rmN) and the worst time complexity is rm(2N-2m-1). The structure of the algorithm's construction set in the worst case is derived. The correctness and completeness of the proposed method are demonstrated. The comparison algorithm shows that when the proportion of non-dominated solutions is small (m/N=20%), the algorithm in this paper is better than the fast non-dominated sorting method in comparison times and CPU running time. When the proportion of non-dominated solutions is large (m/N =80%), the faster construction method of the algorithm in this paper and the arena match method are almost the same in comparison times, but they have obvious advantages in CPU running time.

Keywords

Multi-objective Decision-making; Pareto Optimal Solution; Construction Method; Computational Time Complexity.

1. Introduction

Multi-objective decision-making problem (MOP) [1] refers to more than one objective function and simultaneous optimization of each objective with the same decision variable. Due to the contradiction between the goals, it is impossible to obtain a solution under the condition that each goal is optimal. Therefore, the multi-objective decision-making problem finally obtains a set of solutions, namely the Pareto Optimal Set (Pareto Optimal Set) [2]. In actual production and life, the problems we encounter often have multiple attributes. When solving the problems, multiple goals must be optimized at the same time, such as: job shop scheduling, land planning, investment issues, etc. Since the problem of multi-objective decision-making is closer to the actual problem, the problem of multi-objective decision-making has been studied by a large number of scholars since it was proposed.

Multi-objective evolutionary algorithm (MOEA) has significant advantages in solving multi-objective decision-making problems because of its ability to solve highly complex nonlinear problems. According to the basic framework adopted by the algorithm, it can be divided into: the method based on the dominance relationship, the fast non-dominated sorting genetic algorithm (NSGA-2) with elite retention strategy proposed by Deb [3] in 2002. Based on the decomposition method, in 2016, Zheng Jinhua et[4] proposed a preference multi-objective decomposition algorithm (MOEA/DPRE) based on weight iteration. The algorithm uses the weight iteration method to obtain a set of uniform weight vectors. At the same time, it also eliminates the influence of the location information of the decision maker's preference on the performance of the algorithm. Based on the performance index method, in 2016, Li et[5] proposed a multi-index multi-objective optimization random ranking algorithm (SRA). Both algorithms proposed to use two indicators to reflect convergence and diversity and make it A balanced approach to approach the actual Pareto frontier. According to the time complexity of the construction method, the NSGA series proposed by Deb et al.[3,6] in 2000 adopts the hierarchical order construction method, and the time complexity is $O(rN^2)$.

Based on the divide and conquer technology, in 2003 Jensen et al.[7] used recursive ideas to propose an algorithm for constructing non-dominated sets. Its time complexity is $\mathrm{O}(\mathrm{N}(logN)^{r-1})$. When the number r gradually increases to a certain value, the efficiency of the algorithm will continue to decrease. In 2007, Zheng Jinhua et al.[8] proposed a method to construct the multi-objective Pareto optimal solution set based on the natural phenomenon. This method has high efficiency and the time complexity is $\mathrm{O}(\mathrm{rmN})$. In 2008, Fang et al.[9] also proposed a construction algorithm based on the idea of divide and conquer and introduced the concept of dominance tree. The time complexity of the algorithm is $\mathrm{O}(\mathrm{rNlogN})$. In 2016, Wang Fang et al.[10] deduced the transitivity lemma of dominance relations and the non-dominated solution set construction theorem and lemma by analyzing the properties of non-dominated solutions, and proposed the rapid construction of Pareto non-dominated solutions under multi-objective decision-making. Method, and proved the correctness and completeness of the method, the worst-case time complexity is $\mathrm{O}(\mathrm{rm}(4\mathrm{N}-5\mathrm{m}-1)/2)$. For multi-objective decision-making problems with large solution spaces, Wang Yong et al. [11] proposed a Pareto non-dominated solution set construction algorithm based on initial set sorting based on ordered set theory and operation rules in 2018, and experiments proved that it is more efficient than other algorithms , The time complexity in the worst case is $\mathrm{O}((\mathrm{M}-2)N^2)$.

This paper proposes a Pareto optimal solution set construction algorithm based on the line-up method, gives the nature of non-dominated relations and the related definitions and theorems for constructing the Pareto optimal solution set, proves the correctness and completeness of the algorithm, and analyzes The time complexity of the algorithm is $\mathrm{O}(\mathrm{rmN})$, and the structure of the structure set corresponding to the worst time complexity is derived. The experiments are performed on the number of targets of 2, 5, 8, and 10 dimensions in different population sizes and non-dominated solution proportions. In the case of testing, when the proportion of non-dominated solutions is small (m/N=20%), the algorithm in this paper is better than the fast non-dominated sorting method in comparison times and CPU running time. When the proportion of non-dominated solutions is large (m/N=20%), When N=80%), the faster construction method of the algorithm in this paper and the arena match method are almost the same in the number of comparisons, but they have obvious advantages in CPU running time.

2. Related definitions and theorems

Definition 1: Without loss of generality, taking the multi-objective decision-making problem minimization as an example, set a decision vector $\vec{x}=\{x_1,x_2,\dots,x_n\}$, where n is the number of decision variables, there are r objective functions, With k inequality constraints and l equality constraints, the multi-objective decision-making problem can be expressed as:

$$\min \vec{f}(\vec{x})=\{f_1(\vec{x}),f_2(\vec{x}),\dots,f_r(\vec{x})\}$$

$$\text{s.t.}\quad \vec{g}(\vec{x})=\{g_1(\vec{x}),g_2(\vec{x}),\dots,g_k(\vec{x})\}\le 0 \tag{1}$$

$$\vec{h}(\vec{x})=\{h_1(\vec{x}),h_2(\vec{x}),\dots,h_l(\vec{x})\}=0 \tag{2}$$

Where $f_a(\vec{x})$ is the ath objective function, $\mathrm{a}=1,2,\dots,\mathrm{r}$, $g_b(\vec{x})$ is the bth objective function, $\mathrm{b}=1,2,\dots,\mathrm{k}$, $h_c(\vec{x})$ Is the cth objective function, $\mathrm{c}=1,2,\dots,\mathrm{l}$.

Definition 2: The solution that satisfies the constraints of formulas (1) and (2) is a feasible solution for a multi-objective decision-making problem. The set of feasible solutions is called the feasible solution set, which can be expressed as:

$$\mathrm{X}=\{\vec{x_i}|\mathrm{i}=1,2,\dots,\mathrm{N},\vec{g}(\vec{x})\le 0,\vec{h}(\vec{x})=0\}$$

Where $\vec{x_i}$ is the i-th feasible solution, X is the feasible solution set, and N is the number of feasible solutions.

Definition 3: For$x_i,x_j\in X$, if the following conditions are met, then x_i dominates x_j, denoted as $x_i\succ x_j$_j, x_i is the non-dominated solution, and x_j is the dominated solution.

$$\forall \mathrm{k}, f_k(x_i)\le f_k(x_j), k\in\{1,2,\dots,r\}$$

$$\exists k, f_k(x_i) < f_k(x_j), k \in \{1,2,\dots,r\}, i \neq j, x_i, x_j \in X$$

Definition 4: For $x_i, x_j \in X$, if the following conditions are met, then x_i is independent of x_j, or a non-dominant relationship, and the set composed of x_j is called the irrelevant set of x_i.

$$\nexists x_i \succ x_j, \nexists x_j \succ x_i, i \neq j$$

Definition 5: For $x_i \in X$, if the following conditions are met, then x_i is called the non-dominated individual of the feasible solution set X. The set composed of all the non-dominated individuals of the feasible solution set X is called the non-dominated solution set of X, or Pareto The optimal solution set.

$$\nexists x_j \in X, x_j \succ x_i, i \neq j$$

Lemma 1: Transitivity theorem of dominance relations [12], if$x_a, x_b, x_c \in X, x_a \succ x_b, x_b \succ x_c$, then $x_a \succ x_c$。

Proof: According to Definition 4, if $x_a \succ x_b$, there will be $\forall k, f_k(x_a) \leq f_k(x_b), k \in \{1,2,\dots,r\}$, $\exists k, f_k(x_a) < f_k(x_b), k \in \{1,2,\dots,r\}$; if $x_b \succ x_c$, there will be$\forall k, f_k(x_b) \leq f_k(x_c), k \in \{1,2,\dots,r\}$, inequality is transitive, so there must be $\forall k, f_k(x_a) \leq f_k(x_c), k \in \{1,2,\dots,r\}$, $\exists k, f_k(x_a) < f_k(x_c),\ k \in \{1,2,\dots,r\}$, Therefore $x_a \succ x_c$, the certificate is complete.

Theorem 1: The feasible solution set of the multi-objective decision-making problem, if there is a feasible solution that has nothing to do with other feasible solutions, then this solution is a non-dominated solution of the feasible solution set.

Proof: Suppose $x_i \in X$, for $\forall x_j \in X$, i ≠ j, x_i has nothing to do with x_j. According to Definition 4, there must be no $\forall k, f_k(x_i) \geq f_k(x_j), k \in \{1,2,\dots,r\}$, so no feasible solution x_j dominates the feasible solution x_i in the feasible solution set. According to Definition 5, the feasible solution x_i is the non-dominated solution of the feasible solution set X. The proof is complete.

Theorem 2: In the feasible solution set, $x_i \in X$, the solution dominated by the feasible solution x_i must not be a non-dominated solution, and the feasible solution x_i is not necessarily the Pareto optimal solution.

Proof: According to Definition 3, the solution dominated by the feasible solution x_i is the dominating solution, the feasible solution x_i is the non-dominated solution, and x_i may be dominated by other solutions to become the dominating solution. According to Lemma 1 and Theorem 1, the dominance relationship is continuously transmitted, and finally Become the Pareto optimal solution of the feasible solution set.

Theorem 3: If $x_a, x_b, x_c \in X, x_a \succ x_b$, x_b has nothing to do with x_c, there must be no $x_c \succ x_a$.

Proof: The method of proof by contradiction. Suppose there will be $x_c \succ x_a$, and because $x_a \succ x_b$, according to Lemma 1, then $x_c \succ x_b$, which is inconsistent with the conditions x_b and x_c, so the assumption is not true, there must be no $x_c \succ x_a$, the certificate is complete.

Theorem 4: If $x_a, x_b \in X$, Y are irrelevant sets of x_b, and $x_a \succ x_b$, then for $\forall x_c \in Y$, there will be $x_a \succ x_c$ or x_a independent of x_c.

Proof: The method of proof by contradiction. Assuming that x_a does not dominate x_c or x_a has nothing to do with x_c, then the relationship between x_a and x_c is only $x_a \prec x_c$, and because of $x_a \succ x_b$, according to Lemma 1, there will be $x_b \prec x_c$. According to the condition combination definition 4, it can be obtained that x_b has nothing to do with x_c, which is contrary to the hypothesis. Therefore, if the hypothesis does not hold, there will be $x_a \succ x_c$ or x_a has nothing to do with x_c, and the proof is complete.

3. Non-dominated solution set construction algorithm

In the algorithm for solving multi-objective decision-making problems, constructing the Pareto optimal solution set is an indispensable part of the algorithm. The basic idea of the Pareto optimal solution set construction algorithm based on the queuing method is to first line up the construction set to form a team. Each team has one as the team leader. Each team leader is not related to the players in the team. By default, the first team leader is in turn with the team. Comparing other teams, if the first team leader is better than a certain team leader or a certain team leader is better than the first team leader, the two teams are reorganized into a new team; if the two team leaders are irrelevant, the team members who do not meet the conditions will be laid off. After a round of comparison, the first team leader is the non-dominant individual, and the players of the first team leader are queued into the team, and the above process is repeated until the team is empty.

3.1 Algorithm design of line-up method

The algorithm for constructing the Pareto non-dominated set designed in this paper consists of three parts. The first part is responsible for lining up the group. The group is roughly screened during the queuing process. Those who do not meet the conditions will not be formed. After lining up, each team has a captain. The players of each team are not related to the captain; the second part: Responsible for screening non-dominated solutions, select a non-dominated solution once in each cycle (default first team captain), and delete some dominated solutions dominated by the current non-dominated solution; The third part: Responsible for deleting the dominant solution of the current non-dominated solution mixed into the formation. After the third part is completed, because the non-dominant solution (default first team leader) is selected, the first team does not have a leader and needs to line up the players again. This returns to the first part, and the team will be added to the team after being lined up. The related symbols and meanings involved in the construction method are shown in Table 1:

Table 1. Related symbols and meanings of the construction method

symbols	meanings of symbols
Pop	Feasible solution set (population)
Nds	Non-dominated solution set
Unit	The collection that needs to be lined up, initially Pop
Cmps	Compare the set of captains, and store the captains of each team after lining up
Curs	The collection of individuals in the irrelevant set of all captains in Cmps
X	Non-dominant individual, the first team captain in Cmps is defaulted at the beginning
Xurs	The irrelevant set of X, the individual is stored in the irrelevant set after comparison with X
Y	Comparing individuals, the captain in Cmps
Yurs	Irrelevant set of Y
Ds	Domination solution set, used to store the dominated individuals after comparison of two bodies
Wcom	The set to be compared is used to store the set of captains that has nothing to do with the X before being replaced
Wcurs	The collection of individuals in the irrelevant set of all captains in Wcom
W	Comparing individuals, the captain in Wcom
Wurs	Irrelevant set of W
Ncom	Irrelevant set, used to store the captain set irrelevant to the current X
Nurs	The collection of individuals in the irrelevant set of all captains in Ncom
Cxurs	The set to be compared, stores the individuals that are not compared with the current X
Z	Mark the individual
Flag	When replacing X, store the current position of X in the irrelevant set of Y, initially -1

The first part: line up. The first individual who selects the Unit is compared with the other individuals of the Unit in turn. There are three situations when comparing:

(1) The first individual has nothing to do with the comparative individual: the comparative individual enters the irrelevant set of the first individual;

(2) The first individual dominates the comparative individual: the comparative individual enters the Ds, and the first individual enters the Cmps as the team leader;

(3) The first individual is dominated by the comparison individual: the comparison individual is sequentially compared with the individuals in the irrelevant set of the first individual. If the individual in the irrelevant set is dominated, the individual in the irrelevant set enters Ds, otherwise it enters the irrelevant set of the comparative individual, irrelevant set After the comparison is over, the first individual enters the Ds, and the comparison individual enters the Cmps as the team leader.

After the above operation comparison is over, $\text{Unit} = \emptyset$, assign the first captain of Cmps to X, and the number of Cmps individuals is less than 1. At this time, the set of captains Cmps and Curs, and X and Xurs can be obtained, and conclusion 1 is obtained, and formula 3 is established.

$$\text{Pop} = \text{X} + \text{Xurs} + \text{Cmps} + \text{Curs} + \text{Ds} + \text{Nds} \tag{3}$$

Conclusion 1: Cmps is not related to individuals in Curs, and X is not related to individuals in Xurs, which can be proved by definition 4;

The second part: select non-dominated individuals. Use X to compare with the Y of Cmps in turn, there will be the following three situations when comparing:

(1) X dominates Y: Cxurs=Yurs+Cxurs, Y enters Ds;

(2) X is dominated by Y: Cxurs=Cxurs+Xurs, Y replaces X, Xurs=Yurs, Wcom=Wcom+Ncom, Ncom=$\emptyset$, X before replacement enters Ds;

(3) X has nothing to do with Y: X is compared with the individuals in Yurs in turn. Since the non-dominated relationship is not transitive, the comparison results are divided into three cases:

1) If X dominates Yurs individual, then Yurs individual enters Ds;

2) If X has nothing to do with Yurs individual, compare the next Yurs individual;

3) If X is dominated by Yurs individual, then Cxurs=Cxurs+Xurs, Yurs individual replaces X, X enters Ds before replacement, records the current Yurs individual position to Flag, and marks Z=X, Wcom=Wcom+Ncom, Ncom=$\emptyset$. After the Yurs comparison is over, if the replacement of X occurs during the period, Y enters Wcom, otherwise Y enters Ncom.

After the above operation is completed, Cmps=$\emptyset$, we can get the set Wcom that has nothing to do with the captain before the nth replacement, and the set Ncom that has nothing to do with the captain of the current X, as well as the irrelevant sets of the respective captains, the irrelevant set Xurs of X, and the set Xurs with the current X The uncompared set Cxurs, at this time X is a non-dominated solution, and conclusion 2 is obtained.

Conclusion 2: 1) The captain in Wcom must not be able to dominate X, which can be proved by Theorem 3.

2) Individuals in the irrelevant concentration of their respective captains in Wcom are dominated by X or have nothing to do with X, which can be proved by Theorem 4;

3) The individuals in Ncom, Nurs and Xurs have nothing to do with X, which can be proved by definition 4;

4) X is a non-dominated solution of Pop, which can be proved by Definition 5.

At this time, from formula (3), formula (4) must be established:

$$\text{Pop=Wcom+Wcurs+Ncom+Nurs+X+Xurs+Cxurs+Nds+Ds} \tag{4}$$

The third part: select the dominant solution. It can be seen from conclusion 2 that the individuals in Ncom, Nurs and Xurs have nothing to do with X, but the Wcom squadron and its irrelevant set, as well as Cxurs, may have a dominant solution.

To filter Cxurs, the current X is compared with the individuals in Cxurs in turn. There are two situations in the comparison:

(1) X dominates the Cxurs individual, the Cxurs individual enters Ds, and continues to compare the next individual;

(2) If X has nothing to do with the Cxurs individual, then the Cxurs individual enters Xurs and continues to compare with the next individual.

To filter the Wcom squadron leader and its irrelevant set, X is compared with W in Wcom in turn. There are two situations in the comparison:

(1) X has nothing to do with W, and X is compared with the individual in Wurs in turn. If X dominates the Wurs individual, then the Wurs individual enters Ds and compares the next Wurs individual; if X has nothing to do with the Wurs individual, then compares the next Wurs individual;

(2) X dominates W, and X is compared with the individual in Wurs in turn. If X dominates the Wurs individual, then the Wurs individual enters Ds and compares the next Wurs individual; if X has nothing to do with the Wurs individual, then the Wurs individual enters the irrelevant set of X, Continue to compare the next Wurs individual. After the Wurs individual comparison is over, Wurs=∅, W enters Ds.

After the third part is completed, X enters Nds, Unit=Xurs, Cmps = Wcom + Ncom, put the irrelevant sets of each captain in Wcom and Ncom into the captain irrelevant set of Cmps, Wcom =Ncom=Xurs=X=∅ , Get conclusion 3.

Conclusion 3: The individuals in Unit, Cmps and Curs have nothing to do with X, which can be proved by definition 4.

At this time, from formula (4), formula (5) must be established:

$$\mathrm{Pop} = \mathrm{Unit} + \mathrm{Cmps} + \mathrm{Curs} + \mathrm{Nds} + \mathrm{Ds} \tag{5}$$

Proof: When the first part is completed, Pop's individual is divided into five parts, namely X, Xurs, Cmps, Curs, and Ds; the second and third parts operate on X, Xurs, Cmps, and Curs, and the dominant individual enters Ds. The dominant individual enters the Nds, and the X unrelated individuals before entering the Nds are all in Unit, Cmps, and Curs. The other sets Xurs, X, Wcom, Ncom and their respective irrelevant sets are all empty sets, so the equation holds, and the proof is complete. .

Repeat the above part until Cmps, Unit, and Curs are all empty sets. At this time, the individuals in Nds are all non-dominated solutions, and Nds is the Pareto optimal solution set.

3.2 Proof of the correctness and completeness of the algorithm

Proof of the correctness of the algorithm: Prove that the individuals entering the Nds are all non-dominated solutions in the Pop solution space.

Proof: Every individual who enters the Nds enters after the third part is completed. From conclusion 2, X is a non-dominated solution of Pop, and only the second part of the three parts is the operation of the non-dominated solution. Each cycle ends. Later, X is the solution obtained by comparing with all the individuals in Cmps, Curs, and Unit, and is not dominated by the solutions therein. According to the transitivity of the governing relationship in Lemma 1, the individuals entering Ds cannot be dominated either. X, and when the number of cycles is greater than 1, at the beginning of each cycle, Cmps, Curs, and Unit are not related to the individuals who entered Nds in the previous round, and because in the first cycle, the first part is to queue up the Pop solution space , So the individuals entering the Nds are all non-dominated solutions of the Pop solution space. The proof is complete.

Proof of completeness of the algorithm: Prove that all non-dominated solutions in the Pop solution space have entered Nds.

Proof: At the end of the algorithm, Cmps, Unit, and Curs are all empty sets. At this time, according to formula (5), $\mathrm{Pop} = \mathrm{Nds} + \mathrm{Ds}$, individuals in Pop enter Nds and Ds, and individuals in Ds are all dominated by And enter, so all non-dominated solutions in the Pop solution space have entered Nds, and the proof is complete.

3.3 Algorithm time complexity analysis

Suppose a multi-objective decision-making problem has r goals, N feasible solutions, and m non-dominated solutions. According to the method described in 3.1, the algorithm finds a non-dominated solution every time it loops, so it is executed m times. The related symbols and meanings involved in the time complexity analysis are shown in Table 2:

Table 2. Symbols and meanings related to time complexity analysis

symbols	meanings of symbols
np	The total number of constructible individuals, initially N
i	The i-th loop of the algorithm, the maximum value is m
$\mathrm{n}u_1^i$	After the i-th queue, the first individual directly dominates the total number of individuals in the queue of other individuals, except for the captain
$\mathrm{n}u_2^i$	After the i-th queue, the total number of individuals in the queue where the first individual is dominated by other individuals, except for the captain
$\mathrm{n}d_1^i$	During the i-th queue, the total number of individuals directly dominated by the first individual and the first individual dominated into Ds
$\mathrm{n}d_2^i$	The total number of individuals entering Ds due to domination when comparing the individuals in the unrelated set of the first individual during the i-th queue
$\mathrm{n}x_0^i$	After the i-th queue is completed, the number of individuals in Xurs
ncm^i	The number of individuals in Cmps
ncu^i	Number of individuals in Curs
k	The number of times Y replaces x
$\mathrm{n}x_k^i$	Replace the number of individuals in Xurs for the kth cycle in the i-th cycle
$\mathrm{n}x^i$	The number of individuals in the i-th Xurs
$\mathrm{n}part_2^i$	Number of individuals participating in the second part of the i-th comparison
$\mathrm{n}part_3^i$	The number of individuals participating in the third part of the i-th comparison
e_2^i	The number of individuals entering Ds in the second part of the i-th time
e_3^i	The number of individuals entering Ds in the third part of the i-th time
nds	Number of individuals in Nds
num	The total number of calculations of the algorithm
num_1^i	Number of comparisons in the first part
num_2^i	Number of comparisons in the second part
num_3^i	Number of comparisons in the third part
nc^i	The number of individuals in Ncom
nu^i	Number of individuals in Nurs
ncx	Number of individuals in Cxurs

Part 1: In the process of pairwise comparison, it can be seen from 2.1 that the number of comparisons includes, when the first individual dominates other individuals or has nothing to do with other individuals, a comparison is made. When the first individual is dominated by other individuals, the first individual is dominated by other individuals. An individual in the irrelevant set of individuals is compared again, so the number of comparisons in the first part can be expressed by formula (6):

$$num_1^i = (\mathrm{n}u_1^i + \mathrm{n}d_1^i) + 2(\mathrm{n}u_2^i + \mathrm{n}d_2^i) \tag{6}$$

The second part: Except that some individuals entered Ds in the previous i time, and some individuals entered Nds, after the first part is completed, $\mathrm{Unit} = \emptyset$, the number of individuals participating in the second part comparison can be expressed as, by formula (3) Get formula (7),

$$\mathrm{n}part_2^i = \mathrm{np} - \mathrm{nds} - \left(\sum_{i=1}^{i} \mathrm{n}d_1^i + \sum_{i=1}^{i} \mathrm{n}d_2^i + \sum_{i=1}^{i-1} e_2^i + \sum_{i=1}^{i-1} e_3^i\right) \tag{7}$$

It can be seen from 2.1 that, in the comparison process, X has to be compared with the captain in Cmps and the individuals in their respective irrelevant sets. Therefore, the number of comparisons in the second part can be expressed by formula (8):

$$num_2^i = ncm^i + ncu^i \tag{8}$$

The captain in Cmps and the number of individuals in their respective irrelevant sets are expressed by formulas (3) and (7) to obtain formula (9),

$$ncm^i + ncu^i = \mathrm{np} - \mathrm{nds} - \left(\sum_{i=1}^{i} \mathrm{n}d_1^i + \sum_{i=1}^{i} \mathrm{n}d_2^i + \sum_{i=1}^{i-1} e_2^i + \sum_{i=1}^{i-1} e_3^i\right) - \mathrm{n}x_0^i - 1 \tag{9}$$

Putting formula (9) into formula (8) can get formula (10),

$$num_2^i = \mathrm{np} - \mathrm{nds} - \left(\sum_{i=1}^{i} \mathrm{n}d_1^i + \sum_{i=1}^{i} \mathrm{n}d_2^i + \sum_{i=1}^{i-1} e_2^i + \sum_{i=1}^{i-1} e_3^i\right) - \mathrm{n}x_0^i - 1 \tag{10}$$

The third part: the comparison part is mainly to compare X with Wcom, Ncom, and the irrelevant sets of their respective captains and the individuals in Xurs. The number of individuals participating in the third part comparison can be expressed as, by formula (4), the formula (11),

$$\mathrm{n}part_3^i = \mathrm{np} - \mathrm{nds} - \left(\sum_{i=1}^{i} \mathrm{n}d_1^i + \sum_{i=1}^{i} \mathrm{n}d_2^i + \sum_{i=1}^{i} e_2^i + \sum_{i=1}^{i-1} e_3^i\right) - nc^i - nu^i - \mathrm{n}x^i \quad (11)$$

The individuals participating in the comparison in the third part are all compared by X, so the number of comparisons in the third part can be expressed as: formula (12) can be obtained from formula (11),

$$num_3^i = \mathrm{np} - \mathrm{nds} - \left(\sum_{i=1}^{i} \mathrm{n}d_1^i + \sum_{i=1}^{i} \mathrm{n}d_2^i + \sum_{i=1}^{i} e_2^i + \sum_{i=1}^{i-1} e_3^i\right) - nc^i - nu^i - \mathrm{n}x^i - 1 \quad (12)$$

In summary, adding the comparison times of the three parts of the algorithm is the total comparison times of the algorithm. Since the algorithm is executed m times and each individual has r targets, the total calculation times of the algorithm can be calculated by formula (13) Means,

$$\mathrm{num} = \sum_{i=1}^{m} (num_1^i + num_2^i + num_3^i) * r \quad (13)$$

Substituting formula (6) (10) (12) into formula (13), and simplifying formula (14),

$$\mathrm{num} = \sum_{i=1}^{m} (2\mathrm{np} + (\mathrm{nu}_1^i + \mathrm{nd}_1^i) + 2(\mathrm{nu}_2^i + \mathrm{nd}_2^i) - 2\mathrm{nds} - 2\left(\sum_{i=1}^{i} \mathrm{nd}_1^i + \sum_{i=1}^{i} \mathrm{nd}_2^i + \sum_{i=1}^{i-1} e_2^i + \sum_{i=1}^{i-1} e_3^i\right) - e_2^i - nc^i - nu^i - \mathrm{nx}_0^i - \mathrm{n}x^i - 2) * r \quad (14)$$

According to the meanings defined by nds, nds,nd_1^i,nd_2^i,e_2^i,e_3^i,nc^i,nu^i,$\mathrm{n}x^i$and $\mathrm{n}x_0^i$, they are all non-negative integers; because $(\mathrm{nu}_1^i + \mathrm{nd}_1^i + \mathrm{nu}_2^i + \mathrm{nd}_2^i)$ each cycle is an irrelevant set of X, namely $\mathrm{nu}_1^i + \mathrm{nd}_1^i + \mathrm{nu}_2^i + \mathrm{nd}_2^i = nx^{i-1}$, the initial value is N, so there must be $(\mathrm{nu}_1^i + \mathrm{nd}_1^i) + 2(\mathrm{nu}_2^i + \mathrm{nd}_2^i) < 2\mathrm{N}$; and because np is initially N, formula (14) must be less than or equal to $\sum_{i=1}^{m}(4\mathrm{N} - 2) * r < 4\mathrm{rmN}$, that is, the time complexity of the Pareto optimal solution set construction algorithm based on the queuing method is $\mathrm{O(rmN)}$.

Because in general, m<N, this article is better than the fast non-dominated sorting algorithm with time complexity of $\mathrm{O}(\mathrm{r}N^2)$ and recursion with time complexity of $\mathrm{O}(\mathrm{Nlog}^{(r-1)}\mathrm{N})$The method is of the same order as the quick construction method and the ring game rule.

3.4 Worst time complexity analysis

The worst time complexity requirement is equivalent to finding the maximum value of formula (14), that is, finding the maximum value of formula $(2\mathrm{np} + (\mathrm{nu}_1^i + \mathrm{nd}_1^i) + 2(\mathrm{nu}_2^i + \mathrm{nd}_2^i))$, $(2\mathrm{nds} + 2\left(\sum_{i=1}^{i} \mathrm{nd}_1^i + \sum_{i=1}^{i} \mathrm{nd}_2^i + \sum_{i=1}^{i-1} e_2^i + \sum_{i=1}^{i-1} e_3^i\right) + e_2^i + nc^i + nu^i + \mathrm{nx}_0^i + \mathrm{n}x^i)$ minimum value, below Discuss the size of each value.

(1) np is the total number of constructible individuals, initially N, which is N;

(2) nds is the number of individuals in Nds, one individual will enter Nds at the end of each cycle, so nds is equal to the number of cycles, that is, $\mathrm{nds} = \mathrm{m}$;

(3) The minimum value of e_2^i,e_3^i,nc^i and nu^i, according to their definition, the minimum is 0;

(4) The maximum value of $(\mathrm{nu}_1^i + \mathrm{nd}_1^i) + 2(\mathrm{nu}_2^i + \mathrm{nd}_2^i)$, because $\mathrm{nu}_1^i + \mathrm{nd}_1^i + \mathrm{nu}_2^i + \mathrm{nd}_2^i = nx^{i-1}$, according to their respective meanings, when $\mathrm{nu}_1^i = \mathrm{nd}_2^i = 0$ and $\mathrm{nd}_1^i = 1$, $(\mathrm{nu}_1^i + \mathrm{nd}_1^i) + 2(\mathrm{nu}_2^i + \mathrm{nd}_2^i)$ takes the maximum value It is $2\mathrm{nu}_2^i + 1$. At this time, the first individual of Unit enters Ds, the last individual is X, and the rest are in Xurs. Since Cmps is an empty set, it does not enter the second part of the loop, ncx=0, and the third part selects X as a non-dominated individual to enter Nds. According to formula (5), formula (15) can be obtained,

$$\mathrm{Unit} = \mathrm{Pop} - \mathrm{Nds} - \mathrm{Ds} \quad (15)$$

At this time, the individual of Unit is the individual in Xurs at the end of the previous cycle. The number of individuals in Nds is m. Since only one individual enters Ds at a time, the number of individuals in Ds is m. From formula (15), formula (16)),

$$\mathrm{nu}_2^i = \mathrm{N} - 2\mathrm{m} - 2 \quad (16)$$

Therefore, the maximum value of$(\mathrm{nu}_1^i + \mathrm{nd}_1^i) + 2(\mathrm{nu}_2^i + \mathrm{nd}_2^i)$ is $2\mathrm{N} - 4\mathrm{m} - 3$.

5) The minimum value of nx_0^i and nx^i. The meaning of nx_0^i is the number of individuals in Xurs after the i-th queue is completed. When $nu_1^i = nd_2^i = 0$, $nd_1^i = 1$, We can get $nx_0^i = nu_2^i$, because the second part of the loop is not entered, so$nx^i = nx_0^i$.

In summary, when $np = N, nds = m, e_2^i = e_3^i = nc^i = nu^i = nu_1^i = nd_2^i = 0, nd_1^i = 1$, num Take the maximum value, and because X, as a non-dominated solution, enters Nds after the third part is completed, the number of Nds will increase by 1 in the first and next cycle, so $m = i - 1$, put the condition Enter formula (14) and simplify formula (17),

$$\max num = \sum_{i=1}^{m}(2N - 4i + 1) * r = rm(2N - 2m - 1) \tag{17}$$

Therefore, the worst time complexity of constructing the Pareto non-dominated solution set by the queuing method is$rm(2N - 2m - 1)$.

3.5 The structural analysis of the solution to the structure set corresponding to the worst time complexity

From the discussion in 2.4, we can see that the worst time complexity condition is $nd_1^i = 1, nu_2^i = N - 2(i - 1) - 2, e_2^i = e_3^i = nc^i = nu^i = nu_1^i = nd_2^i = 0, nx_0^i = nx^i = nu_2^i$.

In the first part, $nd_1^i = 1$ means that the number of individuals directly dominated by the first individual and the first individual dominated into Ds is 1; $nu_1^i = 0$ means that there is no first individual except for the captain Directly dominate the cohort of other individuals; $nd_2^i = 0$ means that the individual who dominates the first individual does not dominate the unrelated individuals of the first individual; $nu_2^i = N - 2(i - 1) - 2$ means except for the captain , The total number of individuals in the queue where the first individual is dominated by other individuals. If the first cycle $i = 1$, nu_2^i contains N-2 individuals. As we know before, there is one individual as the captain , There is another individual entering Ds, it can be judged that it entered Ds because of the first individual dominated by the following individual. In other words, when i=1, the most complete feasible solution set operation, the number is N, the first individual is compared with the following individuals in turn, when the Nth individual is compared, the first individual is taken by the Nth individual Dominate. At this time, the Nth individual compares the individuals in the first individual's irrelevant set in turn. After the comparison is over, the first individual enters Ds, and the Nth individual serves as the team leader, so the team leader's irrelevant set contains N-2 individuals , When $i \neq 1$, it is an operation on the irrelevant set nx^i of X, and the same is true.

In the second part, $e_2^i = 0$ means that the number of individuals entering Ds due to control in the second part is 0; $nc^i = nu^i = 0$ means that the individuals in Ncom and Nurs are 0. Since there is only one captain assigned to X in the first part, Cmps is an empty set, so it does not enter the second part of the loop.

In the third part, $e_3^i = 0$ means that the number of individuals entering Ds due to the dominance of the third part is 0; $nx_0^i = nx^i = nu_2^i$ means that the irrelevant solutions of X experience the same in the second and three parts.

In summary, the worst-case structure of the distribution structure of the centralized solution is that the first N/2 individuals among the N individuals are all dominated solutions, the following individuals are all non-dominated solutions, and the first individual is dominated by the Nth individual, and Not related to other individuals, the second individual is dominated by the N-1th individual and not related to other individuals, and so on, until the N/2th individual is dominated by the Nmth individual and is not related to other individuals.

For example, the construction set of 14 individuals of the following 2 targets meets the worst-case construction set,

{C1=[1,20], C2=[2,18], C3=[3,16], C4=[4,14], C5=[5,12], C6=[6,10], C7=[7,8], C8=[7,7], C9=[6,9], C10=[5,11], C11=[4,13], C12=[3,15], C13=[2,17], C14=[1,19]}. Among them, the first 7 are dominated solutions, the last 7 are non-dominated solutions, C14 dominates C1 and is not related to other individuals, C13 dominates C2, and is not related to other individuals, and so on, C8 dominates C7.

4. Algorithm performance test experiment

This paper chooses NSGA2's fast non-dominated sorting method (NDS) [3], fast construction method (NTCM) [10], and arena match method (AP) [8] as the comparison algorithm, and verify the effectiveness of the algorithm by comparing the above algorithms. As well as the efficiency of constructing Pareto optimal solution set, the experiments of the above algorithms are carried out in the following environment, operating system: Windows 7 Ultimate, processor: Intel®Core™i5-3470CPU@3.2GHz3.2GHz, RAM: 4.00GB, Platform: Visual Studio 2017, language: C++. At the same time, in order to eliminate the influence of different population distributions on the algorithm, each experiment in this article uses the same population for different algorithms.

In this experiment, there are 4 algorithms to participate in the comparison. The number of targets is 2, 5, 8, and 10 for experiments, corresponding to Figures 1-4. In order to further compare the performance of the algorithms under different proportions of non-dominated solutions in the total scale, each dimension According to the percentage of non-dominated solutions in the total population size (m/N), it is divided into three parts, namely 20%, 50%, and 80%. Each part contains populations of 5 sizes. The four algorithms with the same target number, population size, and proportion of non-dominated solutions are called a set of experiments. In order to eliminate contingency, each set of experiments is performed 30 times, and the experimental results are the average of the 30 experiments, so this article A total of $4 \times 3 \times 5 \times 4 \times 30 = 7200$ experiments were done. In order to accurately verify the efficiency of the algorithm to construct Pareto and avoid the influence of the distribution of different populations on the algorithm, this paper uses the same distribution of populations for each group of experiments.

Through the comparison of Figures 1-4, we can find:

(1) Under the same target number, population size, and proportion of non-dominated solutions, the comparison times of NDS are much higher than other algorithms, while the comparison times of QP, AP and NTCM are basically the same.

(2) Under the same target number, population size, and different proportions of non-dominated solutions, the number of comparisons of NDS is the same, while QP, AP and NTCM are more affected. As the proportion of m/N increases, the number of comparisons increases significantly. This is because the time complexity of NDS is O(rN^2), and the time complexity of other algorithms is $O(rmN)$.

(3) When the number of targets is small (r=2) and the proportion of non-dominated solutions is small (m/N=20%), the running time of each algorithm is similar, but as the proportion of non-dominated solutions increases, AP and NTCM CPU time has increased significantly, and the time is the most. NDS has the least running time and less QP. This is because compared with NDS, QP, AP and NTCM are not only time-consuming to compare the number of times, but also time-consuming to convert between sets. Of course it is also related to the experimental environment.

(4) When the number of targets is large (r>2) and the proportion of non-dominated solutions is small (m/N=20%), NDS has the largest number of comparisons and CPU running time compared to other algorithms, while QP, AP and NTCM are similar in the number of comparisons and CPU running time. With the increase in the proportion of non-dominated solutions, although AP and NTCM are compared less frequently than NDS, the CPU running time of AP and NTCM increases significantly. The proportion of non-dominated solutions increased from 20% to 50%, and the CPU running time of QP increased slightly. When the proportion of non-dominated solutions continued to increase to 80%, the CPU running time of QP dropped again, which was different from the CPU running time of NDS. Within 2-10s.

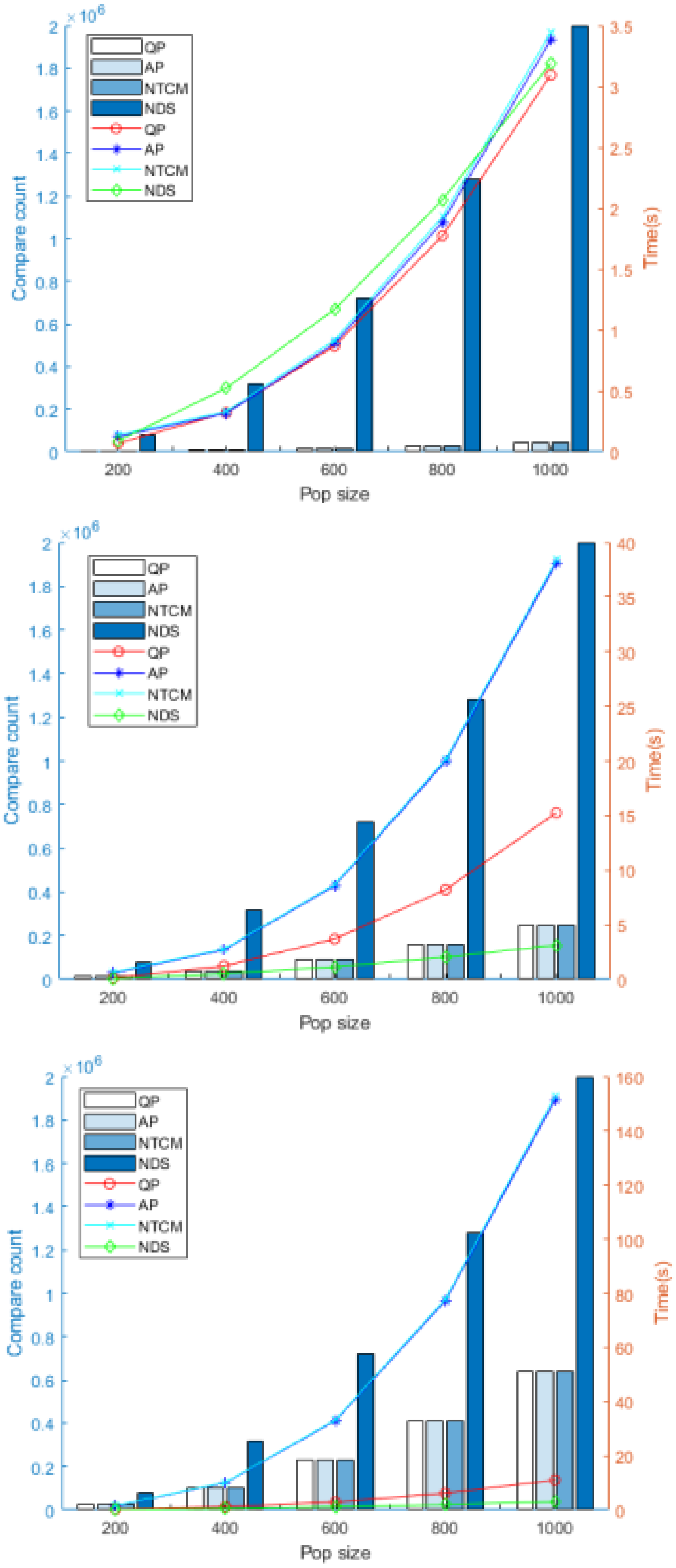

Figure 1. r=2, m/N is from left to right 20%,50%,80%

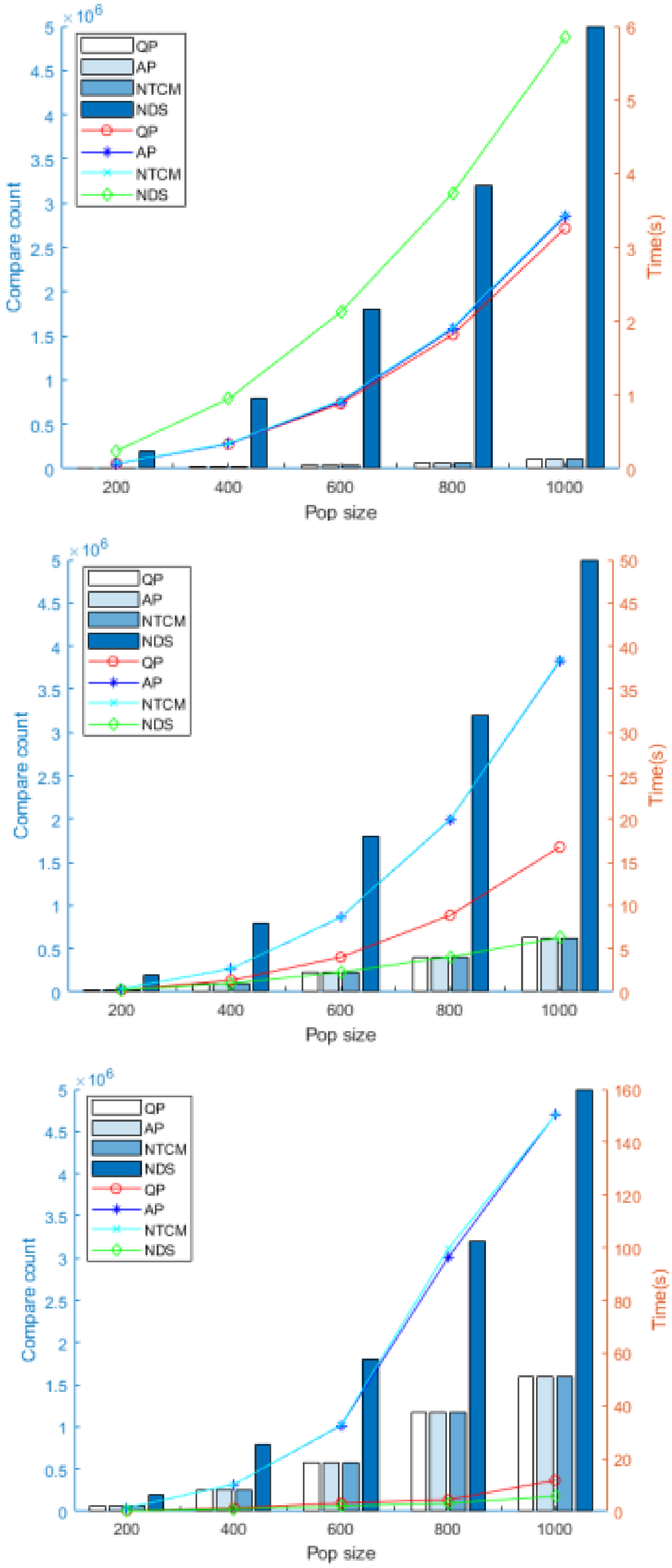

Figure 2. r=5, m/N is from left to right 20%,50%,80%

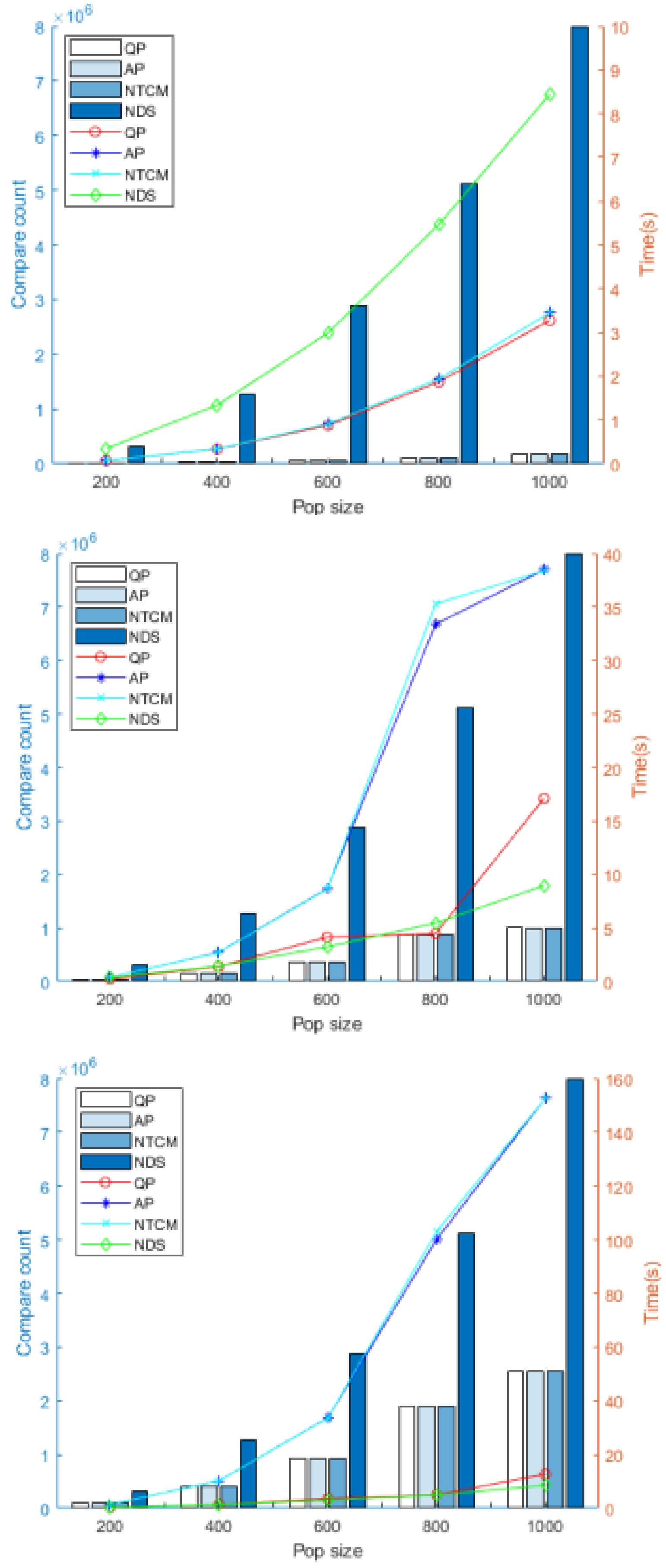

Figure 3. r=8, m/N is from left to right 20%,50%,80%

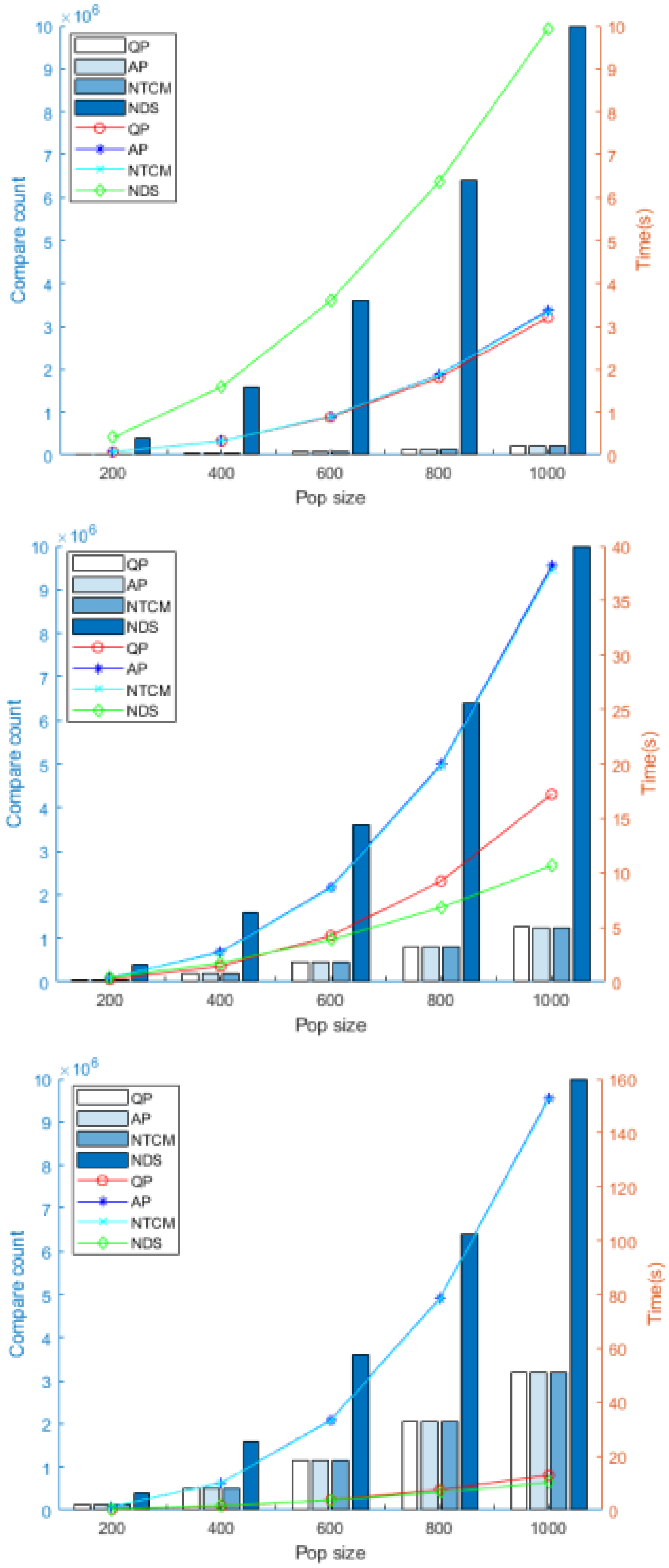

Figure 4. r=10, m/N is from left to right 20%,50%,80%

5. Conclusion

To solve a multi-objective decision problem, it is necessary to construct a Pareto non-dominated solution set. How to efficiently construct a Pareto non-dominated solution set is the most important thing. This paper proposes a Pareto optimal solution set construction algorithm based on the queuing method, and gives the non-dominated solution set at the same time. The nature of the relationship and the relevant definitions and theorems for constructing Pareto optimal solution set and proving it. The time complexity of the proposed algorithm is analyzed as (rmN), and the worst time complexity is deduced as $\mathrm{rm}(2\mathrm{N}-2\mathrm{m}-1)$ and The corresponding structure set structure demonstrates the correctness and completeness of the proposed method. Experiments show that when the number of targets is large (r>2) and the proportion of non-dominated solutions is small (m/N=20%), the comparison times and CPU running time of QP are better than those of NDS, which is almost the same as AP and NTCM. As the proportion of non-dominated solutions increases, the CPU running time of AP and NTCM increases significantly. As the proportion of non-dominated solutions increases, the CPU running time of QP increases first and then decreases. When the proportion of non-dominated solutions increases to 80%, QP and NDS The difference in CPU running time is within 2-10s, but the number of comparisons of QP is lower than that of NDS, and QP is much shorter than the CPU running time of AP and NTCM.

References

[1] SAWARAGIY, NAKAYAMAH, TANINOT. Theory of multi-objective optimization[M]. New York, USA: Academic Press, 1985.

[2] PARETO V. Cours D' economie politique: I, II [M]. Lausanne, Switzerland: F. Rouge, 1896.

[3] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II [J]. IEEE Trans. On Evolutionary Computation, 2002, 6(2): 182-197.

[4] Zheng Jinhua, Yu Guo, Jia Yue. Research on the preference multi-objective decomposition algorithm based on weight iteration to solve the influence of reference points on the algorithm[J]. Chinese Journal of Electronics, 2016,44(01):67-76. (in Chinese)

[5] B. Li, K. Tang, J. Li, X. Yao, Stochastic Ranking Algorithm for Many-Objective Optimization Based on Multiple Indicators [J]. IEEE Trans. Evol. Comput., 2016, 20(6): 924-938.

[6] Deb K, Agrawal S, Pratab A, Meyarivan T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. KanGAL Report, 200001, Kanpur: Indian Institute of Technology, 2000.

[7] Jensen MT. Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms. IEEE Trans. On Evolutionary Computation, 2003, 7(5):503-515.

[8] Zheng Jinhua, Jiang Hao, Kuang Da, Shi Zhongzhi. Method of Constructing Multi-objective Pareto Optimal Solution Set by the Rule of Arena Competition[J]. Journal of Software, 2007(06): 1287-1297.(in Chinese)

[9] Chen Qing, Georganas N D, Petriu E M. Real-time vision-based hand gesture recognition using Haar-like features [C]// Proceedings of Instrumentation and Measurement Technology Conference, 1-3 May, 2007:1-6.

[10] Wang Fang, Rao Yunqing, Tang Qiuhua, He Xiaoxia, Zhang Liping. Fast construction method of Pareto non-dominated solution under multi-objective decision making [J]. Systems Engineering Theory and Practice, 2016, 36(02):454-463.(in Chinese)

[11] Wang Yong, Cheng Jiao, Gao Na, Wang Jing. Pareto non-dominated solution set construction algorithm based on initial set sorting[J]. System Engineering Theory and Practice, 2018, 38(04): 960-970.(in Chinese)

[12] Deb K. Multi-objects evolutionary optimization: Past, present and future [C]// Proceedings of the Fourth International Conferece on Adaptive Computing in Design and Manufacture. London: Springer, 2000: 225-236.