A New Immune Clone Algorithm to solve the constrained optimization problems

LIANG ZHOU, JIANGUO ZHENG
Glorious Sun School of Business & Management
Donghua University
1882 West Yan'an Rd., Shanghai, 200051
P. R. CHINA
homzhou@163.com

Abstract: - In recent years, the constrained optimization problems have become a hot topic among the interest of scholars. In this paper, a new improved artificial immune algorithm is proposed and then used for solving constrained optimizations problems (COPs). This algorithm will treat these COPs as multi-objective optimization problems, and it is based on the concept of Pareto optimization to solve COPs. The mechanism of clone is imported into this new immune algorithm, at the same time, the new improved immune algorithm consists some new concepts, such as linear non-equilibrium recombination operator and preference difference, which can build an efficient immune model for solving this kind of multi-object problems. Finally, simulation on some test functions show that the new immune clone algorithm can obtain better results compared with the existing algorithms.

Key-Words: - constrained optimization, Multi-object optimization, linear non-equilibrium recombination operator, immune, clone, preference difference, Pareto optimization

1 Introduction

Biological immune system is a very complex, high parallel and self-adaptive system. It can identify, eliminate the foreign matter of antigen, and this biological immune system has the ability of studying, remembering and self-adaptive at the same time, which can maintain the balance within the organic environment. In recent years the artificial immune system (AIS) which is put forward based on the theory of biological immune system [1].

AIS has also been widely used in many fields and demonstrated its superior performance [2]. These include the field of constrained optimization [3] or multi-object optimization [4]. For example, Shang ronghua et al. proposed immune clonal multi-objective optimization algorithm, which can treats constrained optimization as a multi-objective optimization with two objectives [3]. Yang dongdong et al proposed a new preference rank immune memory clone selection algorithm to solve the problem of multi-objective optimization with a large number of objectives [5]. In the same period, Shang ronghua et al. pointed out that the new algorithm based on the immune clonal theory can be used to test the complex multi-objective problems, and there are much better performance in both the convergence and diversity [6].

Generally speaking, the traditional AIS can be used to solve the optimization problems. However, COPs have their own characteristics, and it is not appropriate to apply AIS to solve the COPs directly. In this paper, based on the theory of biological immune system, an improved immune clone algorithm (IICA) is proposed in order to solve the COPs. This paper will combine with the thoughts in literature [3], and transform the COPs into multi-object optimization problems (MOPs). Then the new algorithm will use some new mechanisms, such as clone, linear non-equilibrium recombination operator and preference difference operation. These operations will help to improve the performance for solving the COPs. The numerical results show that this new algorithm is more robust and efficient. The chapters are arranged as following: chapter 2 will introduce the preliminaries about COPs and the basic theories of AIS, chapter 3 proposes an improved immune clone algorithm, which can be used to solve the COPs effectively, chapter 4 give the simulation results and chapter 5 is the conclusion of the paper.

2 Preliminaries

2.1 Formulation of the constrained optimization problems

Constrained Optimization Problems are common mathematical programming problems in the actual application. Without loss of generality, a non-linear programming problem can be described as following model[7].

$$\min \quad y = f(x) \qquad (1)$$

s.t.

$$g(x) = (g_1(x), g_2(x), \ldots\ldots, g_l(x)) \le 0$$

$$h(x) = (h_{l+1}(x), h_{l+2}(x), \ldots\ldots, h_m(x)) = 0$$

where.

$$x = (x_1, x_2, \ldots\ldots, x_n) \in X \subseteq R^n$$

$$X = \{(x_1, x_2, \ldots\ldots, x_n) \mid l_i \le x_i \le u_i\}$$

$$l = (l_1, l_2, \ldots\ldots, l_n),$$

$$u = (u_1, u_2, \ldots\ldots, u_n)$$

Where x is n-dimensional decision vector; y is the object function. l and u are the lower limit and top limit of decision vector, respectively.

As for the COPs, its feasible space X_f is a set with the decision variables which can satisfy all the restrained conditions. And its global optimum can be described as follows.

If x^* is a global optimum solution, if and only if $x^* \in X_f$, and there was no $x \in X_f$ which satisfy $f(x) \le f(x^*)$.

Generally speaking, how to deal with the constrained conditions is a key operation in model (1). The existing methods are mainly penalty function method, remaining viable solution method, Ray-Tai-Seow method and so on [8,9]. To take penalty function method for an example, it is the most popular method; it allows the individuals in the population to violate the constrained conditions to a certain extent. However, these individuals will be punished in order to reduce the possibility which the individuals are selected. Although this method is very simple, in the practical operation, how to choose a proper penalty factor is very difficult.

In recent years, the new method which transforms the constrained conditions into multi-object optimization problems begins to attract scholars' attention. At the same time, artificial immune system can succeed in work out the multi-object optimization problems [10]. In this paper, based on the previous corresponding investigations in literature [3], combined with the characters of COPs, a new improved immune algorithm is used to deal with these COPs.

Firstly, the constrained conditions in the object function will be processed as follows.

$$F_j(x) = \begin{cases} [\max[\{0, g_j(x)], 1 \le j \le l \\ |h_j(x)|, l+1 \le j \le m \end{cases}$$

Then we can have a new function:

$$F(x) = \sum_{j=1}^{m} F_j(x) \qquad (2)$$

Where formula (2) transform the constrained into a object function $F(x)$, then a new object optimization problem is consisted of two object function. It can be described as following model.

$$\min \quad y = (f(x), F(x)) = (f_1(x), f_2(x)) \qquad (3)$$

where.

$$x = (x_1, x_2, \ldots\ldots, x_n) \in X \subseteq R^n$$

$$X = \{(x_1, x_2, \ldots\ldots, x_n) \mid l_i \le x_i \le u_i\}$$

$$l = (l_1, l_2, \ldots\ldots, l_n),$$

$$u = (u_1, u_2, \ldots\ldots, u_n)$$

It is also important to note that there are lots of differences between traditional MOPs and common single-object optimization problems (SOPs). For example, as for the SOPs, its best solution is a global optimum, so when the algorithm is used to deal with SOPs, the key operation is to maintain the diversity of the whole population in order to avoid prematurity. However, when the algorithm is used to deal with MOPs, the most important step is to find a Pareto frontier. Therefore, in this paper, in order to deal with this new multi object optimization problem, some basic definition about MOPs is introduced as follows according to the reference [11].

Definition 1 (Pareto better solution): as for the MOPs, there are two feasible solutions, x_A and x_B, it can define that x_A is Pareto better solution compared with x_B, if and only if :

$$\forall i = 1,2,\ldots,m, f_i(x_A) \le f_i(x_B) \ \wedge$$

$$\exists j = 1,2,\ldots,m, f_j(x_A) < f_j(x_B)$$

And it can be called: $x_A \succ x_B$, or x_A control x_B.

Definition 2 (Pareto optimization solution): there is a solution x^*, which is called Pareto optimization solution (or non-control solution), if and only if: $\neg\exists\, x \in X : x \succ x^*$.

Definition 3 (Pareto optimization solution set): all the Pareto optimization solution consist the Pareto optimization solution set, defined as follows: $P^* = \{x^* \mid \neg\exists x \in X : x \succ x^*\}$

Definition 4 (Pareto frontier): as for Pareto optimization solution set P^*, all the Pareto optimization solution form an objective vector, which can be seen as a Pareto frontier. And it is defined as PF^*.

Thus, from afore-mentioned definitions, we can see that as for the traditional MOPs, the solving process is to find a Pareto optimization solution set, which can be trending Pareto frontier. This process is very complex and we can not find a best solution as the global optima. However, the COPs are different from traditional MOPs, and COPs can degenerate into SOPs in the feasible solution space, so its best solution is also a global optimum. Therefore, there are not fully equivalent between traditional MOPs and the new multi-object optimization problem which is described in model (3) in this paper.

2.2 The theory about artificial immune algorithm

Biological immune system is a complicated system, and its particular strengths are ability to study, memory and self-adaptive adjustment. Artificial immune algorithm is a calculating model which combines the main features of biological immune system with engineering application. The diagram of artificial Immune algorithm can be seen below in figure 1 [12].

As shown in figure 1, antigen is corresponding to the object function of optimization problems; antibody is corresponding to the candidate solution of optimization problems. The antibodies affinities can describe the approximation degree between feasible solution and optimal solution in the optimization problems. At the same time, there are some mechanisms such as clone selection, genetic cross-over, and genetic mutation which can be used to solve the optimization problems effectively.

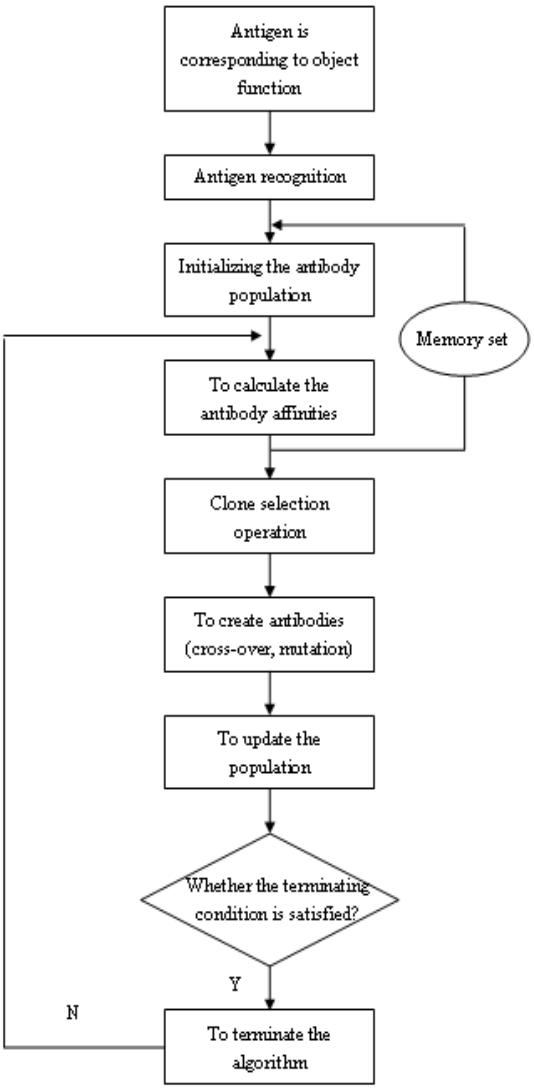

Fig. 1 the process of artificial immune algorithm

In this paper, we will discuss the following mechanisms in order to construct a new improved clone immune algorithm based on the intrinsic characteristics of immune system.

(1) Clonal selection theory

Immunologist Burnet proposed a new clonal selection theory in the late 1950s [13]. His central idea is that the antibodies around the cell surface will react selectively with antigens. This reaction will lead the cells to be clonal expansion. The cells after clonal expansion have the similar character with their father cells. At the same time, some cells will differentiate into antibody-producing cells, and the others become immune-memory cells, which participate in the second immune response. This process is a clonal selection process. At a later time, Castro proposed a complete and systematic immune clone selection algorithm [14]. Castro pointed out that the cells which can recognize the antigen will have

fission process. The fission process is not a simple copy and replication. These cells after fission will be through gene mutation in order to maintain the diversity of the whole immune cells.

(2) Genetic mutation theory

Traditional artificial immune theory considers that high-frequency mutation by antibodies is the major cause to make sure the diversity of antibodies and the attraction between antigen and antibody. However, other evolutionary algorithms such as genetic algorithm will use cross-over operator instead of mutation operator as the main operation. That is to say, mutation is a very important operation in immune algorithm [15].

Generally speaking, if we use these theories in the actual optimization problems, it can avoid the pre-maturity and maintain the diversity of solution space. However, pure clonal selection and genetic mutation theory can not ensure the immune algorithm to solve the COPs effectively. The reasons can be summarized as follows.

(1) The progeny antibodies after clone only inherit the genetic features from their corresponding father antibodies. In other words, the feature of father antibodies has a direct bearing on the corresponding progeny antibodies regardless of other antibodies. The result is progeny antibodies can not learn enough excellent information from current good antibody in the whole population, which is lack of heuristic guidance for the progeny antibodies. And lastly, this is not conductive to the global convergence.

(2) The antibody population can maintain its diversity by mutation based on the genetic mutation theory. However, if the immune algorithm uses mutation operation alone to implement evolution, which can guarantee the algorithm does not fall into local optimum with great probability. This does not ensure the algorithm be capable of convergence to global optima. Therefore, pure mutation operation has limited significance for the immune algorithm to find the global optimum.

Based on the reason above, we propose an improved immune clone algorithm in order to solve the COPs effectively in the following chapter.

3 An improved immune clone algorithm

As for the traditional AIS, the immune mechanism in the organisms is embodied in the recognition of self or non-self by organisms themselves, and then these organisms can exclude the non-self. It also means that the organisms can recognize and obviate the foreign object of antigen in order to maintain its balance of physiology. That is to say, antigen can induce the organisms' immune response and has a specific reaction with corresponding antibody.

In this paper, new clone and recombination mechanism are imported into the immune algorithm, at the same time, the defining of concepts which are called preference difference and antibody affinities are used in order to build the improved immune clone algorithm (IICA).

Firstly, the transformed multi-object optimization problem in this paper can be seen as antigen: $y = (f(x), F(x))$. The candidate solution of the MOPs can be seen as antibody as follows: $a = (a_1, a_2, .., a_n)$. Where $a_1, a_2, .., a_n$ are in compliance with the restriction conditions in model (3). The antibody population is defined as $A_0(it)$. The detailed steps for the IICA appear below.

3.1 Immune clone operation

The essence of clone operation is to copy the optimization individual in father population into the subsequent population. As for the population $A_0(it)$, we can not let every individual have the same scale to achieve the clone operation. We have a general principle before achieving the clone operation.

As for the father individual, more outstanding it is, more bigger its clone scale. Then the excellent individuals will have the larger probability to transfer its excellent information into next generation. At the same time, by using definition (2) and (3), we can divide the population $A_0(it)$ into two sub-populations: $A_{0-1}(it)$ and $A_{0-2}(it)$. $A_{0-2}(it)$ is corresponding to the Pareto optimization solution set, and $A_{0-1}(it)$ is corresponding to the non- Pareto optimization solution set. $A_0(it) = A_{0-1}(it) \bigcup A_{0-2}(it)$.

Therefore, we introduce some new definitions in order to make sure different individual will have different clone opportunity in the solution space.

Definition 5(preference difference ε): as for the model (3), its candidate solution is a and $a \in A_0(it)$, the definition of preference difference ε is as follows:

$$\varepsilon = \sqrt{\sum_{i=1}^{l} (0 - g_i(a))^2 + (\sum_{j=l+1}^{m} h_j(a))^2} \qquad (4)$$

Where $g_i(a)$ and $h_j(a)$ are corresponding to the restriction conditions in model (1), respectively. Preference difference ε can be used to measure the difference between the current candidate solution a and actual feasible region. Generally, more lager the ε is, the greater the difference between a and actual feasible region. This means the possibility of a to become a feasible solution become smaller, too.

Definition 6(antibody affinities): as for the formula (3), we have the following:

$$\alpha_1 = \min(f(x)),$$
$$\alpha_2 = \min(F(x))$$

Then for the random individual a in the population, we have an antibody affinity.

$$aff = \sqrt{(\alpha_1 - f(a(it)))^2 + (\alpha_2 - F(a(it)))^2} \quad (5)$$

From the formula (5), we can see that if *aff* becomes larger, it means the matching degree between father antibody individual and antigen is to be lower.

According to the Definition 5 and Definition 6, we will apply the theory of nature objects existed being excellent and washed out being of feebleness and build a principle to achieve the clone operation. Be especially careful that only the individuals in $A_{0-2}(it)$ have the opportunity to involve this operation. So the self-adaptive clone scale for the different individual in $A_{0-2}(it)$ can be described as follows.

$$d = \text{int}(\alpha \times \frac{1}{aff} \times \frac{1}{\varepsilon})$$

Where $\text{int}(\bullet)$ is a function rounds a number to the nearest integer. α is a clone coefficient, it can be used to set the upper limit for the clone scale, and it will be set to a positive integer which is between [3,5] according to the actual needs.

Lastly, we construct a memory set, which combines two individual. They are as follows:

$$memory_1 = \min(aff_1, aff_2, \ldots\ldots, aff_m)$$
$$memory_2 = \min(\varepsilon_1, \varepsilon_2, \ldots\ldots, \varepsilon_m)$$

So there are two individuals in the memory set, which can be used to guide the population evolution in the following steps. $memory_1$ means the best solution in current iteration, and $memory_2$ means the smallest difference between the current candidate solution and actual feasible region.

Then this process is:

$A_0(it) \to A_0^{'}(it)$.

It can be described as following procedure.

Procedure 1

Begin

(1)To initialize the antibody population $A_0(it)$;

(2)As for the $A_0(it)$, it will be divided into two sub-population $A_{0-1}(it)$ and $A_{0-2}(it)$. $A_{0-1}(it)$ is corresponding to the non-Pareto optimization solution set; $A_{0-2}(it)$ is corresponding to the Pareto optimization solution set.

(3) To clone the antibody in $A_{0-2}(it)$ with self-adaptive method. Its clone scale can described as follows.

$$d = \text{int}(\alpha \times \frac{1}{aff} \times \frac{1}{\varepsilon})$$

(4) To construct a memory set.

(5) To obtain a new population, $A_0^{'}(it)$.

End

3.2 Immune recombination operation

It might also be noted that as for the $A_0^{'}(it)$, it had been divided into $A_{0-1}^{'}(it)$ and $A_{0-2}^{'}(it)$. In this paper, according to the above definition, we hatched an arrangement as follows. Because $A_{0-1}^{'}(it)$ is the set with non- Pareto optimization solution set, so there will be a heuristic evolution in order to ensure the individuals in $A_{0-1}^{'}(it)$ become better; and $A_{0-2}^{'}(it)$ is the set with Pareto optimization solution set, so there will have a new operation to make sure the individuals in $A_{0-2}^{'}(it)$ can be equally distributed in the solution space. Based on this thought, we define a linear non-equilibrium recombination operator to complete the immune recombination operation. The details are as follows.

(1) To select the individual $a_i(it)$ from the $A_{0-1}^{'}(it)$ and the individual a_{best} in $memory_1$, which can build an independent father vector. And then a non-equilibrium coefficient is defined in formula (6).

$$p_r(a_i) = \frac{aff(a_i)}{aff(a_{best}) + aff(a_i)} \quad (6)$$

(2) According to the non-equilibrium coefficient $p_r(a_i)$, a linear non-equilibrium recombination operator is designed as following.

$dis\tan ce = \|a_{best} - a_i\|$, and $\|\bullet\|$ means the Euclidean distance.

if $a_{best} < a_i$

then $a^{'} = p_r \cdot dis\tan ce + a_{best}$ *or*

$a^{'} = a_i - (1 - p_r) \cdot dis\tan ce$

else $a^{'} = p_r \cdot dis\tan ce + a_i$ *or*

$a^{'} = a_{best} - (1 - p_r) \cdot dis\tan ce$

end

(3) The new individual $a^{'}$ is corresponding to the father individual $a_i(it)$.

So the linear non-equilibrium recombination operator should look like the following illustration.

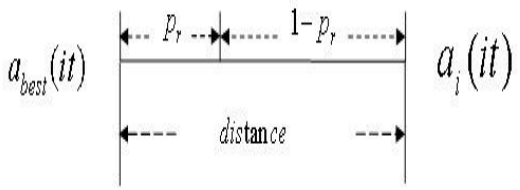

Fig. 2 the illustration of linear non-equilibrium recombination operator

We can see that the individuals in $A_{0-1}(it)$ can maintain its own information after recombination operation, and it can get the guidance of evolution at the same time.

As for the individuals in $A_{0-2}(it)$, its immune recombination operation can help them equally distributed in the solution space. Because the individuals in $A_{0-2}(it)$ have a lot of clone progeny individuals, so recombination possibility p_{re} is defined in order to avoid repeat computation. As an individual $a_i(it)$ in $A_{0-2}(it)$ and the individual a_{best2} in $memory_2$. The non-equilibrium coefficient is defined in formula (7).

$$p_r^{'}(a_i) = \frac{\varepsilon(a_i)}{\varepsilon(a_{best2}) + \varepsilon(a_i)} \quad (7)$$

The details are as follows.

(1) If $p_{re} > rand(1)$, and $rand(1)$ is a function which can create a real number between [0,1]. To select the two individuals, $a_i(it)$ from the $A_{0-2}^{'}(it)$ and the individual a_{best2} in $memory_2$, which can build an independent father vector. And then a non-equilibrium coefficient is defined in formula (7).

(2) According to the non-equilibrium coefficient $p_r^{'}(a_i)$, a linear non-equilibrium recombination operator is designed as following.

$dis\tan ce = \|a_{best2} - a_i\|$, and $\|\bullet\|$ means the Euclidean distance.

if $a_{best2} < a_i$

then $a^{'} = p_r^{'} \cdot dis\tan ce + a_{best2}$ *or*

$a^{'} = a_i - (1 - p_r^{'}) \cdot dis\tan ce$

else $a^{'} = p_r^{'} \cdot dis\tan ce + a_i$ *or*

$a^{'} = a_{best} - (1 - p_r^{'}) \cdot dis\tan ce$

end

(3) The new individual $a^{'}$ is corresponding to the father individual $a_i(it)$.

From the view of Immune recombination operation, we can see that this step focus on guiding the individuals to achieve evolution by using the individuals in memory set. At the same time, the information that the individuals in $A_0^{'}(it)$ contain will transmit into offspring because of the existing of linear non-equilibrium recombination operator. Generally speaking, the results of Immune recombination operation will change the structure of antibody population. Firstly, as for $A_{0-1}^{'}(it)$, the sub-population will trend to object value by using a_{best} in $memory_1$; secondly, as for $A_{0-2}^{'}(it)$, the sub-population will get a equal distribution in the feasible solution space.

Then this process is:

$A_0^{'}(it) \rightarrow A_0^{''}(it)$.

It can be described as following procedure.

Procedure 2

Begin

(1) As for the individuals a_{best} in $memory_1$, a_{best2} in $memory_2$, they will perfume the Immune recombination operation respectively.

(2) As for sub-population $A_{0-1}^{'}(it)$, its scale is N, and every individual $a_i^{'}(it)$ in $A_{0-1}^{'}(it)$ will have the following operation.

For $i = 1:1:N$

Begin

if $a_{best} < a_i$

then $a^{'} = p_r \cdot dis\tan ce + a_{best}$

or $a^{'} = a_i - (1 - p_r) \cdot dis\tan ce$

else $a^{'} = p_r \cdot dis\tan ce + a_i$

or $a^{'} = a_{best} - (1 - p_r) \cdot dis\tan ce$

End

(3) As for sub-population $A'_{0-2}(it)$, to define recombination possibility p_{re}, then they will use the linear recombination operator to execute the same immune recombination operation like step (3).

(4) to obtain a new population, $A''_0(it)$.

End

3.3 Immune mutation operation

From the mechanism of algorithm view, the artificial immune algorithm is very different from traditional evolutionary algorithms[16]. Traditional evolutionary algorithms use a large and density cross-over operator and mutation operator to maintain the diversity of the whole population; this can help the traditional evolutionary algorithms avoid the prematurity. However, the IICA in this paper do not need these operators. Firstly, it uses immune clone operation in order to improve its potential global search ability, and then it can equally dispose its candidate solutions in the feasible solution space by using the immune recombination operation, which can avoid the prematurity and satisfy the need to solve multi-object functions. At the same time, immune mutation operation is used here to avoid the prematurity and maintain the diversity of the population..

The concrete details are as follows.

If $\forall a''_i(it) \in A''_{0-1}(it)$,

then

$$a'''_i(it) = \begin{cases} a_{new}, & if\ rand(1) \geq 0.5 \\ a''_i(it) + \sigma \cdot rand(-1,1), & if\ rand(1) < 0.5 \end{cases}$$

Where σ is a mutation coefficient, it can have different value according to the different object functions. $rand(-1,1)$ is a random function which can create a random number between [-1,1]. $rand(1)$ is a random function which can create a random number between (0,1).

On the face of it, the mutation process is a Gaussian mutation. The process can be divided into two parts. a_{new} means there will be create a random individuals in the feasible solution space, which can be used to extend the feasible solution space and maintain the diversity of population.

If $\forall a''_i(it) \in A''_{0-2}(it)$,

then

$$a'''_i(it) = \begin{cases} a''_i(it), & if\ rand(1) \geq 0.5 \\ a''_i(it) + \sigma \cdot rand(-1,1), & if\ rand(1) < 0.5 \end{cases}$$

Where σ is a mutation coefficient, its values can change according to the different object functions. The mutation process can be divided into two parts, too. But the first part is different with the mutation in $A''_{0-1}(it)$. When $rand(1) \geq 0.5$, the antibody will not change, and keep it in the next generation.

Then this process is:

$A''_0(it) \rightarrow A'''_0(it)$.

It can be described as following procedure.

Procedure 2

Begin

(1)As for $\forall a''_i(it) \in A_{0-1}(it)$

For every individual in $A_{0-1}(it)$

$$a'''_i(it) = \begin{cases} a_{new}, & if\ rand(1) \geq 0.5 \\ a''_i(it) + \sigma \cdot rand(-1,1), & if\ rand(1) < 0.5 \end{cases}$$

End

(2)As for $\forall a''_i(it) \in A_{0-2}(it)$

For every individual in $A_{0-2}(it)$

$$a'''_i(it) = \begin{cases} a''_i(it), & if\ rand(1) \geq 0.5 \\ a''_i(it) + \sigma \cdot rand(-1,1), & if\ rand(1) < 0.5 \end{cases}$$

End

(3) To obtain a new population, $A'''_0(it)$

End

3.4 Immune selection operation

In this paper, the scale of $A'''_0(it)$ has been changed through above three stages of operation. We should select the excellent individuals in order to keep the scale of population be invariant. The details are as follows.

Firstly, the antibodies in $A'''_{0-2}(it)$ should be selected. Then we get $\{a'''_1(it), a'''_2(it), \ldots\ldots, a'''_s(it)\}$.

In general conditions, if $s < n$, then the $A'''_{0-1}(it)$ will provide the remaining $n - s$ individuals randomly. If $s > n$, then the second selection is implementing according to the aff.

Then this process is:

$A''_0(it) \rightarrow A'''_0(it)$.

4 Experiments and simulations

Some famous different benchmark functions are tested in this paper in order to analyze the performance of this new algorithm [17]. These functions are as follows.

Function 1：

$$\min \quad f_1 = (x_1 - 10)^3 + (x_2 - 20)^3$$

subject to

$$g_1 = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \le 0$$

$$g_2 = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \le 0$$

Where $x_1 \in [13,100]$, $x_2 \in [0,100]$. Its global optimum is $f(x^*) = -6961.81388$, $x^* = (14.095, 0.84296)$

Function 2：

$$\min \quad f_1 = x_1^{\ 2} + (x_2 - 1)^2$$

subject to

$$h_1 = x_2 - x_1^2 = 0$$

Where $x_1 \in [-1,1]$, $x_2 \in [-1,1]$. And its global optimum is $f(x^*) = 0.75$, $x^* = (\pm\frac{1}{\sqrt{2}}, \frac{1}{2})$

Function 3：

$$\max \quad f_1 = (\sqrt{10})^{10} \prod_{i=1}^{10} x_i$$

subject to

$$h_1 = \sum_{i=1}^{10} x_i^2 - 1 = 0$$

Where $0 \le x_i \le 1$, and the global optimum is $f(x^*) = 1$, $x_i^* = \frac{1}{\sqrt{10}}$, $i = 1,2,\ldots\ldots,10$.

Function 4:

$$\min$$

$$f(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6 x_7 - 10x_6 - 8x_7$$

subject to

$$g_1(x) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \le 0$$

$$g_2(x) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \le 0$$

$$g_3(x) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \le 0$$

$$g_4(x) = 4x_1^2 + x_2^2 - 3x_1 x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0$$

Where $-10 \le x_i \le 10$, $i = 1,2,\ldots 7$. The optimum solution is $x^* = (2.330499, 1.951372, -0.4775414, 4.365726, -0.6244870, 1.038131, 1.594227)$ and $f(x^*) = 680.6300573$

The performances of this new algorithm IICA will be simulated and tested, and then the simulation results are compared with the existing algorithms, such as IFDNAGA[18] , RY[19], and KM[20]. In this paper, the scale of population is 100. $p_{re} = 0.45$, Function 3 is a function to get the maximization value, so we will transformer it into $-f_1$, and then its optimal value is -1. As for every test functions, they will be computation for 50 times independently. And its statistical results are as table 1.

Table 1 the simulation results

Function	Optimal	Algorithm	Best solution	Mean solution	Worst solution
Function 1	-6961.814	IFDNAGA	-6960.793	-6598.400	-6271.203
		RY	-6961.814	-6875.940	-6350.262
		KM	-6952.141	-6342.667	-5473.982
		IICA	-6961.814	-6942.441	-6903.133
Function 2	0.750	IFDNAGA	0.750	0.742	0.726
		RY	0.750	0.750	0.750
		KM	0.750	0.750	0.750
		IICA	0.750	0.750	0.750
Function 3	-1.000	IFDNAGA	-0.9999	-0.9997	-0.9921
		RY	-1.000	-1.000	-1.000

		KM	-0.9997	-0.9989	-0.9978
		IICA	-1.000	-1.000	-1.000
Function 4	680.630	IFDNAGA	680.641	680.632	680.741
		RY	680.630	680.656	680.763
		KM	680.91	681.16	683.18
		IICA	680.630	680.625	680.690

As shown in table 1, the simulation results of RY and IICA is better than the IFDNAGA and KM. The IFDNAGA algorithm uses the exclusion method, and special saving operation to construct a recognition evolution algorithm, but its judge process is very complex and it has the weak adaptability, so the performance of simulation results is not ideal. At the same time, the KM algorithm has its inherent drawback according to the reference [21]. By using the self-adaptive clone mechanism, The IICA can effectively extend its solution space in order to improve its performance, and then the recombination operation uses the linear non-equilibrium recombination operator to make sure the excellent information in the memory set can be inherited to progeny individuals. Subsequently, the immune mutation operation can be used to avoid the prematurity and maintain the diversity of the population in the solution space. Generally speaking, the overall concept of IICA can help this algorithm find the global optimal feasible solution like RY algorithm.

5 Conclusions

In this paper, an IICA is proposed by introducing the clone mechanism, recombination operation, mutation operation and selection operation, which are used to solve the COPs. The introduction of new concepts into IICA can improve the quality of Pareto optimal solution set, and they can expand the distribution range of Pareto feasible solution, maintain the diversity of the population, and improve the efficiency of algorithm. Based on the fact that IICA successfully solved the complex COPs, and the performance of IICA has satisfied effect.

References:

[1] Guo Meng, Chen Qiuhong, A New Immune Algorithm and Its Application[J], *WSEAS Transactions on Computers*, 9(1), 2010, pp.72-82.

[2] Vincent C. S. Lee, Xingjian Yang, An Artificial Immune System Based Learning Algorithm for Abnormal or Fraudulence Detection in Data Stream, *Proc. Of the 5th WSEAS International Conference on Applied Computer Science*, 4, 2006, pp.822-825.

[3] Ronghua Shang, Lichen Jiao and Wenping Ma, Immune Clonal Multi-Objective Optimization Algorithm for Constrained Optimization[J], *Journal of software,* 19, 2008, pp. 2943-2956.

[4] Yuexuan Wang, Lianchen Liu, Shengjing Mu, Constrained multi-objective optimization evolutionary algorithm[J], *Journal of tsinghua university (science and technology)*, 45, 2005, pp. 1-5.

[5] Dongdong Yang, Lichen Jiao, Maoguo Gong, Clone Selection Algorithm to Solve Preference Multi-Objective Optimization[J], *Journal of Software*, 21, 2010, pp. 14-33.

[6] Ronghua Shang, Licheng Jiao, Maoguo Gong, An Immune Clonal Algorithm for Dynamic Multi-Objective Optimization[J], *Journal of Software*, 18, 2007, pp. 2700-2711.

[7] Zhou Y. R., Li Y. X., Wang Y., Kang L. S., A pareto strength evolutionary algorithm for constrained optimization[J], *Journal of Software*, 14(7), 2003, pp. 1243-1249.

[8] Michalewicz Z., Deb K., Schmidt M., Stidsen T., Test-Case generator for nonlinear contimuous parameter optimization techniques[J], *IEEE Trans. On Evolutionary Computation*, 4(3), 2004, pp. 197-215.

[9] Farmani R., Wright J.A., Self-Adaptive fitness formulation for constrained optimization[J], *IEEE Trans. On Evolutionary Computation*, 7(5), 2003, pp. 445-455.

[10] Shang R.H., Ma W. P., Immune clonal MO algorithm for ZDT Problems. In: Jiao L.C., Wang L. P., Gao X. B., Liu J., Wu F., eds. *Lecture Notes in Computer Science*, 4222. Berlin, Heidelberg, New York: Springer-Verlag, 2006, pp. 100-109.

[11] Maoguo Gong, Licheng Jiao, Dongdong Yang et al: Research on Evolutionary Multi-Objective Optimization Algorithms[J], *Journal of Software*, 20, 2009, pp. 271-289.

[12] Zhou Honggang, Yang Chunde, anomaly detection approach based on immune algorithm and support vector machine[J], *Computer Applications*, 26(9), 2006, pp. 2145-2147.
[13] Macfarlane Burnet, Auto-immune Disease II. Pathology of the Immune Response[J], *British Medical Journal*, 10(2), 1959, pp. 720-725.
[14] Leandro Nunes de Castro, Fernando Jose Von Zuben, The Clonal Selection Algorithm with Engineering Applications. Workshop on Artificial Immune Systems and Their Applications, Las Vegas, USA, *In Workshop Proceedings of GECC'00*, 7, 2000, pp. 36-37.
[15] Cai Zixing, Gong Tao, Advance in research on immune algorithms[J], *Control and Decision,* 19(8), 2004, pp. 841-846
[16] Jiao Licheng, Du Haifeng, Development and Prospect of the Artificial Immune System[J], *ACTA Electronica Sinica*, 31(10), 2003, pp. 1540-1548.
[17] Thomas P. Runarsson, Xin Y., Stochastic Ranking for Constrained Evolutionary Optimization [J], *IEEE Transactions on Evolutiaonry Computation*, 4(3), 2000, pp. 284-294.
[18] Wang Y., Liu L., Mu S., Wu C., Constrained multi-objective optimization evolutianry algorithm[J], *Journal of Tsinghua University*, 45(1), 2005, pp. 103-106.
[19] Thomas P. Runarsson, Xin Y., Stochastic Ranking for Constrained Evolutionary Optimization [J], *IEEE Transactions on Evolutiaonry Computation*, 4(3), 2000, pp. 284-294.
[20] Michalewicz Z., Deb K., Schmidt M., Stidsen T., Test Case generator for nonlinear continuous parameter optimization techniques[J], *IEEE Transactions on Evolutionary Computation*, 4(3), 2000, pp. 197-215.
[21] Thomas P. Runarsson, Xin Y., Stochastic Ranking for Constrained Evolutionary Optimization [J], *IEEE Transactions on Evolutiaonry Computation*, 4(3), 2000, pp.284-294.