International Journal of Hybrid Information Technology
Vol.7, No.4 (2014), pp.331-344
http://dx.doi.org/10.14257/ijhit.2014.7.4.28

An Improved Evolutionary Strategy of Genetic Algorithm and a New Method on Generation of Initial Population When Using Genetic Algorithms for Solving Constrained Optimization Problems

Xu Sun[1], Fulin Wang[2] and Shifa Wen[3]

[1]Heilongjiang Institute of Technology
[1, 2, 3]Northeast Agricultural University
Harbin, China
3912400@qq.com, fulinwang@yahoo.com.cn, wenshifa1@126.com

Abstract

The paper provides an improved evolutionary strategy (ES) of genetic algorithm (GA) on the basis of the existing literature. The ES overcomes the shortage of traditional GA whose excellent child individuals obtained in the crossover process may not survive in the process of mutation. In addition, the crossover probability and mutation probability which is hard to determine in traditional GA is removed for this proposed strategy. At the same time, it increases the number of individuals produced in process of crossover. This may increase the possibility of producing excellent individuals, thus lead to better improvement of the traditional GA. The test result of finding the optimal values of four functions using transitional GA and the proposed GA is presented in this paper. The result shows that the improved ES presented in this paper has faster calculation speed and significantly smaller number of iterations than the traditional GA. Thus, the improvement of improved ES is powerfully illustrated.

Based on articles in the existing research literature, the initial population generation methods were further explored when using the genetic algorithm(GA) for solving constrained optimization problem. Through the research we present a new method about initial interior point's generation. Firstly, construct a constraint posed by the objective function, which is based on the characteristics of constrained optimization problems. Then translate the problem of evaluating the initial interior point into a problem of solving a series of unconstrained optimization. By solving the unconstrained optimization problem, we achieve the solution of the initial interior point. Based on this idea, the research has given a method on the generation of the rest initial population individuals. In addition, through the research we concluded that the key to generate the initial population is to obtain an initial point. The production of other individuals will take less time after the initial internal point is obtained. Finally, we verified by examples that the initial population generation method given by this paper is a fast and reliable method. Thus the shortage of the GA of which the initial population is difficult to be produced in some constrained optimization problem is overcome.

Keywords: *We would like to encourage you to list your keywords in this section. Genetic algorithm; Evolutionary strategies; Crossover probability; mutation probability; Initial interior point; Initial population; Constrained optimization problem*

ISSN: 1738-9968 IJHIT

1. Introduction

Genetic Algorithm(GA) is important and widely used searching method established by Professor John Holland in University of Michigan. The idea of GA origins from the population genetics and Evolution Theory. It is a group optimization method based on genetics and natural selection theory. It is especially useful in solving complex nonlinear optimization problems with constrains which are difficult to solve for the conventional searching methods. It is widely used in areas like machine learning, combinatorial optimization, systems engineering, adaptive control, artificial intelligence, planning and design, intelligent manufacturing systems, intelligent machine systems, artificial life, *etc.*, [2, 3] It is one of the most important intelligent computation technologies in 21th Century.

Compared to the other optimization algorithms, the GA has the following characteristics:

GA is a group optimization algorithm. The searching of GA starts from various initial points, which will avoid the situation where the searching process is trapped in local minimum. As a result, the chance to find the global minimum solution is increased.

GA selects the excellent individuals based on the fitness function, which avoids other derivations and additional information. The robustness of the solution is much better.

GA has no restrictions on the objective functions. GA does not require objective function to be continuous or differentiable. The objective function could be explicit function or implicit function (mapping matrix, neural networks) which leads to wide applications of GA.

Genetic algorithm is a heuristic search method, it is neither exhaustive search nor compete random search. As long as the positions of the genes are selected properly and the genetic manipulation process is reasonable, the searching process will be efficient. Besides, the optimal solution could be achieved in finite searching time.

The computation of GA is parallel, and the computation speed could be greatly improved by using massive parallel computing.

The GA is especially useful for solving complex optimization of large-scale systems.

Although the GA has many advantages, when it is applied to solve the constrained optimization problem, it is sometimes difficult to obtain enough number of initial populations using random number generation method. Sometimes it will take tens of hours or even longer to generate the initial populations. Based on the studies of the algorithm, the authors propose a new method to generate initial populations presented in this paper.

The algorithm proposed by Holland is commonly recognized as traditional GA (or standard GA). De Jong KA first proposed the elitist ES in his doctoral thesis. Later researchers proposed various elitist strategies and ESs based on selection instead of copying [5-8]. The existing GAs are almost following the similar idea of these ESs. In this paper, they are called traditional GAs for short.

This paper presents the work of proposing an improved ES on the basis of existing literature. The test result of finding the optimal values of four functions using transitional GA and the proposed GA is presented in this paper. The comparison and analysis is presented as well.

2. Analysis of Traditional GA and ES

The ES of traditional GA may be illustrated as following four steps. Firstly, the initial population is generated and the crossover rate and mutation rate is initiated. Secondly, the initial population is taken as parent population, and each individual's fitness is calculated. Then the selection, crossover and mutation process is done to generate a child population. Thirdly, a few elite individuals in the parent generation are selected to replace the worst individuals in the child population. Lastly, if the design requirement is satisfied, the process ends; otherwise, treat child population as parent generation and repeat all the above process. Let the population size be n, size of elite individuals to keep be m, crossover rate be Pc, and mutation rate be Pm, then the flowchart of the ES is illustrated in Figure 1.

Through the study of the ES of the traditional GA, the authors found that in the traditional GA evaluation process, the excellent individuals generated in the crossover process may be destroyed in the mutation process. Besides, the crossover rate must be specified in the traditional GA which follows the concept of the basic GA. If the crossover rate is 1, the excellent individuals may not be able to survive. After researching on this algorithm, the authors found that if the elite individuals in the parent population and the crossover process are kept, even if the excellent individuals are destroyed in the crossover process, the excellent individuals will still survive in the child population. Based on this analysis, the crossover rate may be removed and the child population size obtained in the crossover process will be n. In this way, the chance of obtaining excellent individuals will be increased, which will lead to a better performance of the proposed GA compared to the traditional GA.

3. Main Title

Based on the above analysis, this paper proposed a new ES. In the evolution of the GA search, there are two different cases of the individuals of the parent generation chosen to crossover: two parent individuals have the same local solutions or they have different local solutions. For the first case, the crossover process is not necessary, and they will directly proceed to the mutation process. For the second case, the crossover process is carried out as usual. The flowchart of the ES is shown in Figure 2.

Comparing the ES of the traditional GA and the proposed ES, the proposed ES removes the crossover rate which leads to more child individuals generated in the crossover process. As a result, the chance of having excellent individuals is increased. Besides, the excellent individuals in the crossover process is kept, thus the computation speed is greatly increased.

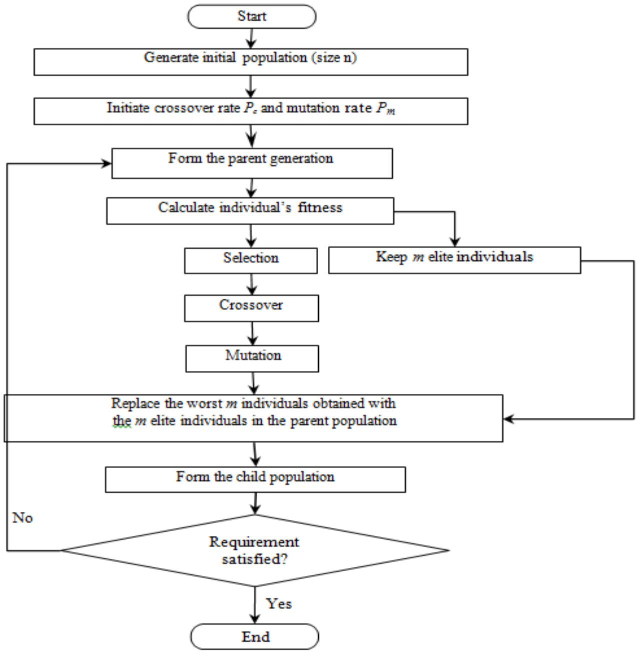
Start
Generate initial population (size n)
Initiate crossover rate P_c and mutation rate P_m
Form the parent generation
Calculate individual's fitness
Selection
Keep m elite individuals
Crossover
Mutation
Replace the worst m individuals obtained with the m elite individuals in the parent population
Form the child population
No
Requirement satisfied?
Yes
End

Figure 1. The Flowchart of the ES of Traditional GA

4. Testing and Analysis of the ES

4.1. The Selection of the Testing Functions and Parameters

Four commonly used complicated testing functions are selected to compare the performances of the proposed ES and the traditional ES, of which three has multiple local extreme points.

Testing function f1:

$$\min f_1(x,y) = 100*(y-x^2)^2 + (x-1)^2$$
$$-10 \le x, y \le 10 \qquad (1)$$

The 3-D plot of the function is shown in Figure 3.

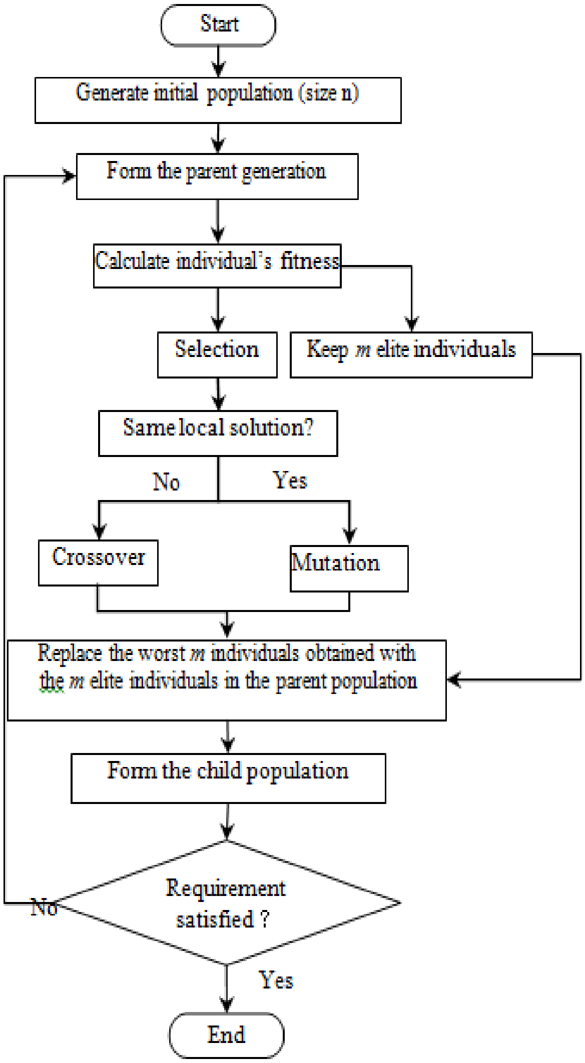
Start
Generate initial population (size n)
Form the parent generation
Calculate individual's fitness
Selection
Keep m elite individuals
Same local solution?
No
Yes
Crossover
Mutation
Replace the worst m individuals obtained with the m elite individuals in the parent population
Form the child population
Requirement satisfied ?
No
Yes
End

Figure 2. Flowchart of the Proposed GA

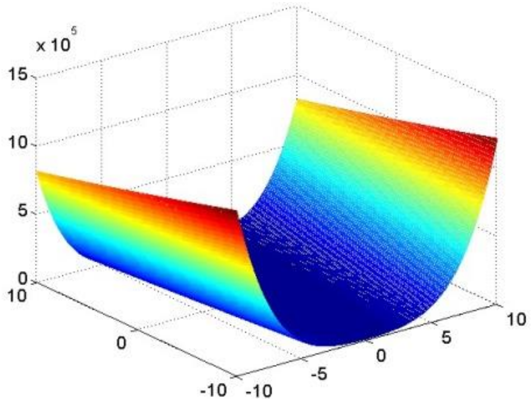

Figure 3. 3-D Plot of f1

Function f1 is called Rosenbrock function, or banana function. It is a typical pathological function that is very difficult to find the minimum point. Although the function only has one global minimum point in the range of -10≤x,y≤10, the region near the global minimum if almost flat which greatly increases the searching difficulty. As a result, it is a commonly used function to evaluate the searching algorithms. Function f1 has a global minimum value 0 at the point (1,1).

Testing function f2:

$$\max f_2(x,y)=\left[\frac{3}{0.05+(x^2+y^2)}\right]^2+(x^2+y^2)^2$$
$$-5.12\le x,y\le 5.12 \qquad (2)$$

The 3-D plot is shown in Figure 4.

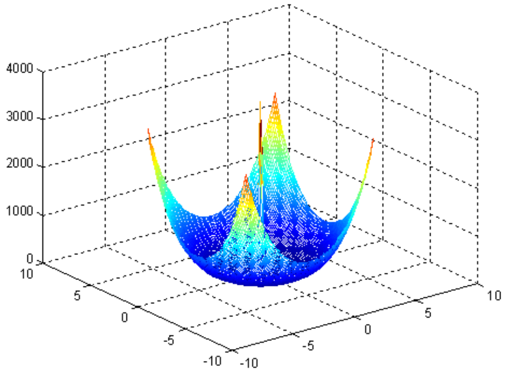

Figure 4. The 3-D Plot of f2

The global maximum point of the function f2 is surrounded by the global minimum points. Besides, the function has four local extreme points which makes it very difficult to find the global maximum point of the function. The function has a global maximum value of 3600 at the point (0,0).

Testing function f3:

$$\max f_3(x,y) = -\frac{3600(\sin^2\sqrt{x^2+y^2}-0.5)}{\left[1+0.001(x^2+y^2)\right]^2} + \left[\frac{3}{0.1+(x^2+y^2)}\right]^2 + (x^2+y^2)^2 \qquad -4 \le x, y \le 4 \tag{3}$$

The 3-D plot of function f3 is shown in Figure 5.

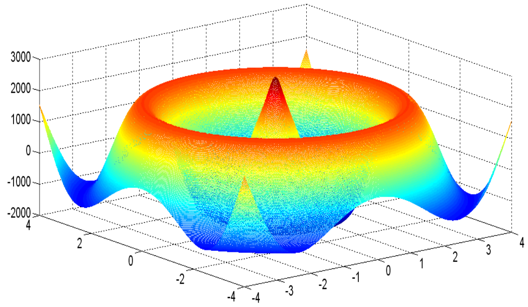

Figure 5. 3-D Plot of f3

Function f3 has infinite number of local maximum points, of which only one is the global maximum point with a value of 2700. However, the global maximum point is surrounded by a circular range of local maximums whose values are all 2698.6. Besides, the function also has four local extreme points. As a result, the searching of maximum value of the function will be easily trapped in the local maximum points.

Testing function f4:

$$\min f_4(x,y) = \left\{\sum_{i=1}^{5} i\cos\left[(i+1)x+i\right]\right\} \left\{\sum_{i=1}^{5} i\cos\left[(i+1)y+i\right]\right\} \qquad -10 \le x, y \le 10$$
$$+0.5\left[(x+1.42513)^2+(y+0.80032)^2\right] \tag{4}$$

The 3-D plot is shown in Figure 6.

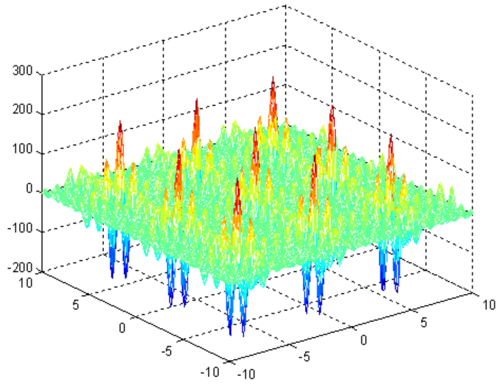

Figure 6. 3-D Plot of f4

Function f4 has 760 local minimum points, of which only point (-1.42513, -0.80032) is the global minimum point. The minimum value of the function is -186.7309.

In the testing procedure, both GAs adopt the binary coding and single point crossover strategy. The initial population is initiated randomly with size n=100. The mutation rate Pm is set to be 0.1 and the crossover rate Pm is 0.9 for traditional GA. The m=10 elite individuals is selected to kept. The calculation result is recorded with an accuracy of ten decimal points.

The ending condition of the two iterative methods is

$$\left| f_i - f_i^* \right| \le \varepsilon_i \qquad i = 1, 2, 3, 4 \tag{5}$$

where f_i^* is the global maximum or minimum value of ith testing function, fi is the minimum or maximum value obtained by the searching algorithms of function fi and ε_i is the given searching accuracy.

The four testing functions are given the same searching accuracy with $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \varepsilon_4 = 10^{-6}$. The parameters of the testing functions are given in the equation (1) ~ (4). The testing of searching algorithm for each function with the two GAs are carried out in the same computer for 100 times.

4.2. Testing Result and Analysis

The testing result of the four testing functions is shown in Table 1.

From Table 1, it can be seen that compared to the traditional ES, the average computation time, longest computation time, shortest computation time, maximum number of iterations, minimum number of iterations and average number of iterations of the improved ES are all much less. The average computation time for finding the minimum point of f1 is 250.36s for improved ES which is 4.48 times faster than 1121. 68s for the traditional ES. For the improved ES, the average number of iterations for searching convergence of f1 is 203.93.

Table 1. The Testing Result of the Two GAs

Function	Searching algorithm	Average computation time/s	Longest computation time/s	Shortest computation time/s	Average number of iterations	Maximum number of iterations	Minimum number of iterations
f_1	Traditional GA	1121.68	5746.06	31.92	1226.28	6209	33
	Improved GA	250.36	1591.46	14.34	203.93	1335	11
f_2	Traditional GA	422.68	874.65	158.32	453.73	943	168
	Improved GA	44.80	85.40	25.12	37.79	72	21
f_3	Traditional GA	270.87	522.30	88.61	276.78	504	92
	Improved GA	41.51	151.79	18.99	34.93	129	16
f_4	Traditional GA	1493.86	7321.59	121.29	1567.98	7660	135
	Improved GA	401.34	3248.76	20.99	314.10	2556	17

From Table 1, it can be seen that compared to the traditional ES, the average computation time, longest computation time, shortest computation time, maximum number of iterations, minimum number of iterations and average number of iterations of the improved ES are all much less. The average computation time for finding the minimum point of f1 is 250.36s for improved ES which is 4.48 times faster than 1121.68s for the traditional ES. For the improved ES, the average number of iterations for searching convergence of f1 is 203.93, which is

16.63% of the number 1226.28 for the traditional ES. For function f2, the average computation time for improved ES is 44.80s which is 9.04 times faster than the 402.68s for traditional ES. For the improved ES, the average number of iterations for searching convergence of f2 is 38.00 which is 8.33% of the number 453.73 for the traditional ES. The average searching time for function f3 is 41.51s and 270.87s; the improved ES has a 6.53 times faster computation speed. The average number of iterations are 34.93 and 276.78 for the two GAs; the improved ES has a 7.92 times less iterations. The average searching time for function f4 is 401.34s and 1493.86s; the improved ES has a 3.72 times faster computation speed. The average number of iterations are 314.10 and 1567.98 for the two GAs; the improved ES has a 5.02 times less iterations. As a result, it may be conclude that the improved GA has a much faster computation speed and less iterations than the tradition GA. As a result, it is shown that the proposed ES is better than the tradition ES.

5. Overview of Methods for Solving Constrained Optimization Problems Using Genetic Algorithm

Constrained optimization problem is frequently encountered in scientific research and engineering applications, the general constrained optimization problem can be described as:

$$\begin{aligned} &\min f(X) = f(x_1, x_2, \dots, x_n) \\ &s.t. \begin{cases} g_j(X) = g_j(x_1, x_2, \dots, x_n) \geq 0 & j = 1, 2, \dots, p \\ h_k(X) = h_k(x_1, x_2, \dots, x_n) = 0 & k = 1, 2, \dots, q \end{cases} \end{aligned} \tag{6}$$

where $X = (x_1, x_2, \dots, x_n)^T$ is a point vector in n dimensional Euler space En. The objective function $f(X)$ and constrains $g_j(X)$, $h_k(X)$ are all explicit functions of X [4].

The main challenge for solving constrained optimization problem is how to effectively balance the searching between feasible region and infeasible regions, which is to design an effective constraint handling method to locate the best possible global regions. To handle the issue of infeasible solutions, in the study of evolutionary algorithms in recent decades, many ideas have been proposed, such as the restoration method, punishment function and so on. Particularly, the restoration method is to restore the infeasible solutions in some way so that it becomes a feasible solution.

When applying the GA to constrained optimization problem, the equality constrains have to be converted into inequality constrains using methods such as variable substitution. As a result, the mathematical model for the GA to solve the constrained optimization problem may be denoted as:

$$\begin{aligned} &\min f(X) = f(x_1, x_2, \dots, x_n) \\ &s.t.\ g_j(X) = g_j(x_1, x_2, \dots, x_n) \geq 0 \quad j = 1, 2, \dots, p \end{aligned} \tag{7}$$

The commonly used ways for the GA to deal with constrained optimization problems include penalty function method, the decoder method, correction operator method and feasible solution search method [5]. The penalty function method is a commonly method for solving constrained optimization problems. To add a suitable penalty term on the fitness function, the constrained optimization problem is converted to unconstrained optimization problem. Penalty function method is simple, but in practice, the determination of the penalty factor is usually difficult. Different from the penalty function method, decoder method adopts special coding method to avoid the generation of infeasible individuals. This method is quite efficient, but it could not be applied to the entire constrained optimization problem. The correction operator method is to design special operator to ensure the child individual generated by the feasible individual is still a feasible solution. In another word, these operators are closed in the feasible solution space. The correction operator method is only applicable to linear constrained problems, but not nonlinear constrained problems. Feasible solution search method is an improved method for the correction operator method; it is applicable for both linear and nonlinear constrained optimization problems.

When solving the constrained optimization problem, the first step of GA is to randomly initialize initial population. Then selection, crossover and mutation process is done to the initial population under the scheme of GA. The algorithm mainly relies on the objective function (fitness function), the excellent individuals will have larger fitness values while the worse individuals will have smaller fitness values. In the searching process, only the fitness function and the constrains are required. The other complicated information is however not required. Meanwhile, GA has the characteristics of randomness, which leads to the advantage of avoiding falling into local optimum points. The first step in the GA is to generate a set of feasible solutions, which was used as the initial population. However, the convergence speed, performance of the algorithm is greatly affected by the initial population, and therefore determining the initial population is particularly important for GA.

6. A New Method of Finding the Initial Population

Definition 1: For any individual in the population, if it satisfies all the constrains in equation (2), then the individual is called the feasible individual. If the individual does not satisfy all the constraints, then the individual is called non-feasible individual;

Definition 2: Define the entire initial feasible individuals as the initial population, the number of individuals in the initial population as population size;

Definition 3: For any one individual in the population, if the individual satisfies all the constrains in equation (3) then the individual is known as the initial point.

$$s.t.\ g_j(X) = g_j(x_1, x_2, \ldots, x_n) > 0 \quad j = 1, 2, \ldots, p \tag{8}$$

The method presented in this paper is adopting internal correction method based on the initial points. The generation of the initial population may be divided into two cases. The first case is that the initial point could be manually given directly. The other case is that the initial point could not be given manually. For the second case, as long as suitable searching method is applied to generate one initial point, the problem is converted same as the first case. As a result, the most important step is to generate the initial point, the details steps are as follows:

Firstly random initialize one point $X_1^{(1)}$, if it satisfies all the constrains, then $X_1^{(1)}$ is the required initial point. If the point satisfies some of the constrains but not all, then the unsatisfied constrains is treated as the virtual objective function, the satisfied constrains are treated as barrier term to form an unconstrained optimization function. Then calculate the negative gradient vector of the function on the point $X_1^{(1)}$. Continue searching on the direction and obtain a new point $X_1^{(1)}$ which has a smaller function value than previous point. If the new $X_1^{(1)}$ is still not the initial point, then follow the same method to reduce the barrier factor and form the new objective function and continue searching until the initial point is obtained.

The recursive procedure of finding the initial point is given as below:

(1)Randomly initialize a point $X_1^{(1)} \in E_n, r_1 > 0$ (barrier factor, for example $r_1 = 1$, let the index $k = 1$.

(2)Obtain the constrain set T_k and $\overline{T_k}$

$$\begin{aligned} T_k &= \{ j \mid g_j(X_1^{(k)}) > 0,\ j = 1,2,\cdots,p \} \\ \overline{T_k} &= \{ j \mid g_j(X_1^{(k)}) \le 0,\ j = 1,2,\cdots,p \} \end{aligned} \tag{9}$$

(3)Check if the set $\overline{T_k}$ is an empty set or not, if it is an empty set, then $X^{(k)}$ is the initial point and the recursive process stops, otherwise continue to next step.

(4) Form the following objective function

$$P(X, r_k) = -\sum_{j \in \overline{T_k}} g_j(X) + r_k \sum_{j \in T_k} \frac{1}{g_j(X)} \ (r_k > 0) \tag{10}$$

(5)Find the negative gradient vector of $P(X, r_k)$ at point $X_1^{(k)}$,

$$p^{(k)} = -\frac{\nabla P(X_1^{(k)}, r_k)}{\| \nabla P(X_1^{(k)}, r_k) \|} \tag{11}$$

(6)Let

$$X_1^{(k+1)} = X_1^{(k)} + \lambda_k p^{(k)} \tag{12}$$

where λ_k is the searching gain.

If the obtained $X_1^{(k+1)}$ does not satisfies equation (8), then half the λ_k and follow equation (7) to recalculate $X_1^{(k+1)}$, until equation (8) is satisfied.

$$P(X_1^{(k+1)}, r_k) < P(X_1^{(k)}, r_k) \tag{13}$$

(7)Let $r_{k+1} = \dfrac{1}{k+1}, k \Leftarrow k+1$, and go back to (2).

The point obtained following the above steps is the required initial point.

7. The Method to Generate the Other Individuals in the Initial Population

Suppose the initial point obtained following the above mentioned method as $X_1^{(k+1)}$, denoted as the first individual in the initial population. Denote $X_1^{(k+1)}$ as X_1 for simplicity, then randomly initial an individual $X_2 = (x_1, x_2, \ldots, x_n)^T$ following the equation (9)

$$x_i = a_i + r_i(b_i - a_i) \quad i = 1, 2, \ldots, n \tag{14}$$

where a_i is the lower bound of the variable x_i, and b_i is the upper bound, r_i is a random number between 0~1.

If X_2 satisfies equation (10)

$$g_j(X) = g_j(x_1, x_2, \ldots, x_n) \ge 0 \quad j = 1, 2, \ldots, p \tag{15}$$

then generate the next individual X_3 , otherwise

$$X_2 \Leftarrow X_1 + \alpha(X_2 - X_1) \tag{16}$$

where α is the convergence factor ($0 \le \alpha < 1$).

When α is 0.5, it is in fact the midpoint of X_1 and X_2 , as shown in Figure 1.

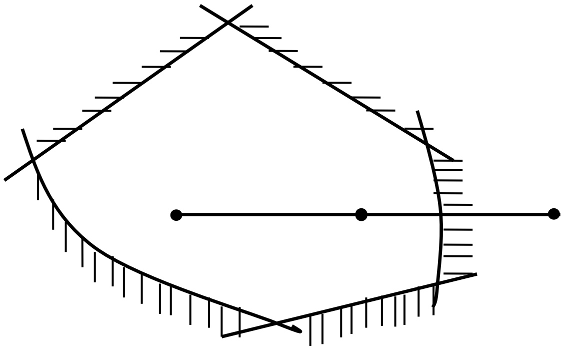

Figure 7. The Relation of X$_1$,Primary X$_2$ and New X$_2$

If X_2 is still not a feasible individual after convergence process following equation (11), then reduce α by half following equation (12)

$$\alpha \Leftarrow 0.5\alpha \tag{17}$$

Then follow equation (11) to generate a new X_2 approaching X_1 until X_2 becomes a feasible individual.

After generating X_2 , the new point X_3 is generated following the same procedure as X_2 to be a new feasible individual. Continuing the same procedure until all the feasible individuals are generated.

8. A Calculation Example

Example 1[6]:

$$\min\ f(X) = x_1 + x_2 + x_3$$

$$s.t.\begin{cases} 0.0025(x_4 + x_6) \le 1 \\ 0.0025(x_5 + x_7 - x_4) \le 1 \\ 0.01(x_8 - x_5) \le 1 \\ -x_1x_6 + 833.33252x_4 + 100x_1 \le 83333.333 \\ -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \le 0 \\ -x_3x_8 + 1250x_5 + x_3x_5 - 2500x_5 \le 0 \\ 100 \le x_1 \le 20000 \\ 1000 \le x_{2,3} \le 20000 \\ 10 \le x_i \le 20000\text{ , } i = 4,5,6,7,8; \end{cases} \tag{18}$$

In example 1, the upper bounds and the lower bounds of variable $x_1 \sim x_8$ are all given in the model. The tests have been done for the initial population generation using the random generation method and the method proposed in this paper. The results of generation time of 100 tests under different population size are given as the following table.

Table 2. Comparison of Creation Time in Different Size of Population of Example 1

Generation Method	Population Size			
	50	100	200	500
Random Generation Method(s)	36.0120	67.0888	131.8163	329.7909
New Method Proposed(s)	0.0339	0.0413	0.0624	0.0903

Example 2[7]:

$$\begin{aligned} \min f(X) = &-x_{10}(x_1 x_4 + x_2 x_5 + x_3 x_6) \\ &-x_{11}(x_4 x_7 + x_5 x_8 + x_5 x_9) \\ &-x_{12}(x_1 x_7 + x_2 x_8 + x_3 x_9) \end{aligned}$$

$$s.t. \begin{cases} x_{10} + x_{11} + x_{12} \le 300000 \\ x_{10}x_1 + x_{11}x_4 + x_{12}x_7 \le 1600000 \\ x_{10}x_2 + x_{11}x_5 + x_{12}x_8 \le 1200000 \\ x_{10}x_3 + x_{11}x_6 + x_{12}x_9 \le 900000 \\ x_{10} \le 300000 \\ x_{11} \le 300000 \\ x_{12} \le 300000 \\ x_i \ge 0, i = 1, 2 \ldots 12; \end{cases} \tag{19}$$

In example 2, the lower bounds of variable $x_1 \sim x_{12}$ are all 0, and the upper bounds of x_{10}, x_{11}, x_{12} are fixed. The upper bounds of $x_1 \sim x_9$ are all 800. The tests have been done for the initial population generation using the random generation method and the method proposed in this paper. The results of generation time of 100 tests under different population size are given as the following table (Because the random generation method is too slow, only 5 times of test are done).

Table 3. Comparison of Creation Time in Different Size of Population of Example 2

Generation Method	Population Size			
	50	100	200	500
Random Generation Method(s)	4248.6632	8443.7233	16832.7195	42409.2278
New Method Proposed(s)	0.1422	0.1615	0.1725	0.2203

From Table 2 and Table 3, it can be seen that the new method of generating initial population can greatly increase the generation speed. As the population size increased, the advantage of the new method becomes more obvious.

9. Conclusion

(1) This ES proposed in this paper overcomes the shortage of tradition ES through which the excellent individuals in the crossover process may be destroyed. The excellent individuals have a larger chance to survive in the new ES thus the performance of the GA is improved.

(2) The improved ES removes the crossover rate and the mutation rate, which increases the number of individuals generated in the crossover process. As a result, the chance of having excellent individuals is increased as well. Besides, it also avoids the difficulty of traditional GA for setting the proper crossover rate and mutation rate.

(3) The testing result of searching maximum/minimum points of the four testing function shows that the proposed ES has a much faster computation speed and less iterations than the tradition ES. Thus the effectiveness of the improved ES is verified.

(4) This paper presents a new method for generating the initial population of GA for solving the constrained optimization problem.

(5)The examples show that the new method proposed in this paper is a fast and reliable method, which overcomes the difficulty for generating the initial population for some constrained optimization problem using GA.

References

[1] Y. M. Chiang, L. C. Chang and F. J. Chang, "Comparison of Static-feed Forward and Dynamic-feedback Neural Networks for Rainfall-runoff Modeling", Journal of Hydrology, vol. 290, no. 3-4, **(2004)**, pp. 297-311.

[2] F. L. Chung and L. Lee. "Network-growth Approach to Design of Feed-forward Neural Networks", IEEE Proc Control Theory Appl, vol. 142, no. 5, **(1995)**, pp. 486-492.

[3] A. Danok, B. McCarl and T. K. White, "Machinery Selection and Crop Planning on a State Farm in Iraq", American Journal of Agricultural Economics, vol. 60, no. 3, **(1978)**, pp. 544-549.

[4] C. Darken and J. Moody, "Towards Faster Stochastic Gradient Search", Advances in Neural Information Processing Systems, San Mateo, CA: Morgan Kaufmann, vol. 4, **(1991)**, pp. 1009-1016.

[5] K. Funahashi, "On the Approximate Realization of Continuous Mappings by Neural Networks", Neural Networks, vol. 2, no.7, **(1989)**, pp. 183-192.

[6] Z. Griliches, "Measuring Inputs in Agriculture: A Critical Survey", Farm Economics, vol. 42, no. 5, **(1960)**, pp. 1411-1427.

[7] T. Gulati, M. Chakrabarti and A. Singh, "Comparative Study of Response Surface Methodology, Artificial Neural Network and Genetic Algorithms for Optimization of Soybean Hydration", Food Technol Biotechnol, vol. 1, no. 48, **(2010)**, pp. 11-18.

[8] J. Leonard and M. A. Kramer, "Improvement of the Back-propagation Algorithm for Training Neural Networks", Computers & Chemical Engineering, vol. 14, no. 1, **(1990)**, pp. 337-341.

[9] K. Levenberg, "A Method for the Solution of Certain Problem in Least Squares", Quart. Appl. Math., vol. 2, pp. 164-168.

[10] R. P. Lippmann, "An Introduction to Computing with Neural Nets", Washington D C: IEEE ASSPM Magazine, **(1999)**.

[11] D. W. Marquardt, "An Algorithm for Least-squares Estimation of Nonlinear Parameters", SIAM J. Appl. Math., vol. 11, **(1963)**, pp. 431-441

[12] R. Martin and B. Heinrich. "A Direct Adaptive Method for Faster Back-propagation Learning: The RPROP Algorithrm", Ruspini H. Proceedings of the IEEE international Conference on Neural Networks (ICNN), IEEE Press, New York, **(1993)**.

Author

Xu Sun (1978-), female, Heilongjiang Harbin people, doctor graduate student,(E-mail) *3912400@qq.com.*