International Journal of Hybrid Information Technology
Vol. 10, No. 6 (2017), pp.47-56
http//dx.doi.org/10.14257/ijhit.2017.10.6.05

A Method of Initial Population Generation of Intelligent Optimization Algorithms for Constrained Global Optimization

Jiquan Wang[1], Okan K. Ersoy[2], Xinxin Chen[1] and Fulin Wang[1]

[1]*College of Engineering, Northeast Agricultural University Harbin 150030, China*
[2]*School of Electrical and Computer Engineering, Purdue University,West Lafayette, Indiana 47907-1285*
wang-jiquan@163.com, ersoy@purdue.edu, 1209098482@qq.com, fulinwang1462@126.com

Abstract

When the constraint conditions and variables are very many in a global optimization application, it is a challenging task to generate initial population to be used in an evolutionary optimization algorithm to solve the constrained global optimization problem. In this paper, a method of rapidly generating an initial population is proposed. The key to this method is to use the genetic algorithm to generate a first initial interior point. First, by using the interior point method, the problem of the first initial interior point of generating the initial population is converted in to solving an unconstrained optimization problem, which is next solved by using the genetic algorithm to generate the first initial interior point. Secondly, the remaining individuals of the initial population are randomly generated. In this process, the feasibility of each randomly generated member is first checked. If it is feasible, then the next member is checked. If this member is infeasible, then it is moved closer to the first interior point until it becomes feasible. When all the members of the population are feasible, the initial population is ready to be used with the intelligent optimization algorithm. The experimental results with three test functions show that the proposed method can quickly generate the initial population.

Keywords: *initial population, intelligent optimization algorithm, genetic algorithm, constrained optimization, interior point*

1. Introduction

Optimization problems with many constraints are quite common in engineering and scientific applications. Some traditional optimization methods have limitations in solving constrained optimization problems, including (1) single-point operation mode greatly limiting the computational efficiency, (2) weak global search capability resulting in local optimum, (3) usually requiring object function and constraint function in the form of analytic functions and so on [1]. The biggest advantages of intelligent optimization algorithms are algorithmic simplicity, multi-point parallel computing capability, strong global search capability, and objective function and constraint conditions with possibly no derivatives [2,3]. In recent years, intelligent optimization algorithms have been rapidly expanding and successfully applied in the fields of system control, production scheduling, artificial intelligence, pattern recognition, path planning and so on [4,5]. Most commonly used intelligent optimization algorithms are genetic algorithm, particle swarm optimization, ant colony algorithm, simulated annealing, tabu search algorithm, particle swarm optimization, and predatory search algorithm [6,7].

A common feature of intelligent optimization algorithms is parallelism, that is to say simultaneously carrying out independent search from multiple points in the solution space. This is why an initial population is necessary. In the case of unconstrained optimization problems or relatively few constrained optimization problems, the

ISSN: 1738-9968 IJHIT

generation of initial population is easy, for example, randomly generating an initial population, some other methods exist to generate the initial population, thereby significantly improving the performance of intelligent optimization algorithms [8,9]. When there are many constraints, these methods do not work well.

Reference [10] presents a method of generating the initial population suitable for constrained optimization problems, and achieves good results. However, the method requires derivatives.

The proposed method aims at constrained optimization problems with many constraints, does not require derivatives, and is fast to compute.

2. Mathematical Model of a Constrained Optimization Problem

In most real world optimization problems, variables occur in a finite range, and such restrictions are embodied by constraints. Optimization problems with constraint conditions are called constrained optimization problems. The mathematical model of constrained optimization is as follows [11]:

$$\min f(X) \quad s.t. \begin{cases} g_j(X) \geq 0, j = 1,2,\cdots,l \\ h_i(X) = 0, i = 1,2,\cdots,m \end{cases} \tag{1}$$

Where $X = (x_1, x_2, \cdots, x_n)^T$ is a *n*-dimensional Euclidean space point (vector). It will be referred to as individual. The objective function and constraint condition are real functions of *X*.

If the i^{th} constraint condition is equality constraint in equation (1), then the i^{th} constraint condition can be replaced by the following two inequality constraints

$$\begin{cases} h_i(X) \geq 0 \\ -h_i(X) \geq 0 \end{cases} \tag{2}$$

Therefore, formula (1) can also be expressed as

$$\min f(X) \quad s.t.\ g_j(X) \geq 0, j = 1,2,\cdots,L \tag{3}$$

3. Method of Initial Population Generation

Definition 1: An individual which satisfy all constraints in formula (3) is called feasible individual, otherwise it is called infeasible individual.

Definition 2: All feasible individuals generated before the first iteration with the intelligent optimization algorithm are called the initial population. The number of individuals is called the population size of the initial population.

Definition 3: An individual of the population within the feasible region (not outside or on the boundary of the feasible region), is called interior point or strict interior point.

3.1. Basic Genetic Algorithm

The evolutionary strategy of basic genetic algorithm is as follows: first, the initial population is generated. Using the initial population as the parent generation, all individuals in the population are sorted in ascending order according to their objective function values; the fitness value of each individual is calculated according to its objective function value. This is followed by making selections, crossovers, retaining *s* elite individuals from *n* parent individuals and cross-generated *n* individuals. Cross-generated *n* individuals are mutated according to mutation probability p_m, and the worst *s* individuals including np_m mutated individuals and $(1-p_m)n$ nonmutated individuals are

replaced by *s* elite individuals. In this way, the offspring population is generated. If the computing requirements are met, then the iterations are stopped. If the computing requirements are not met, the above steps are repeated until the computing requirements are met. The evolutionary strategy of the basic genetic algorithm is shown in Figure 1.

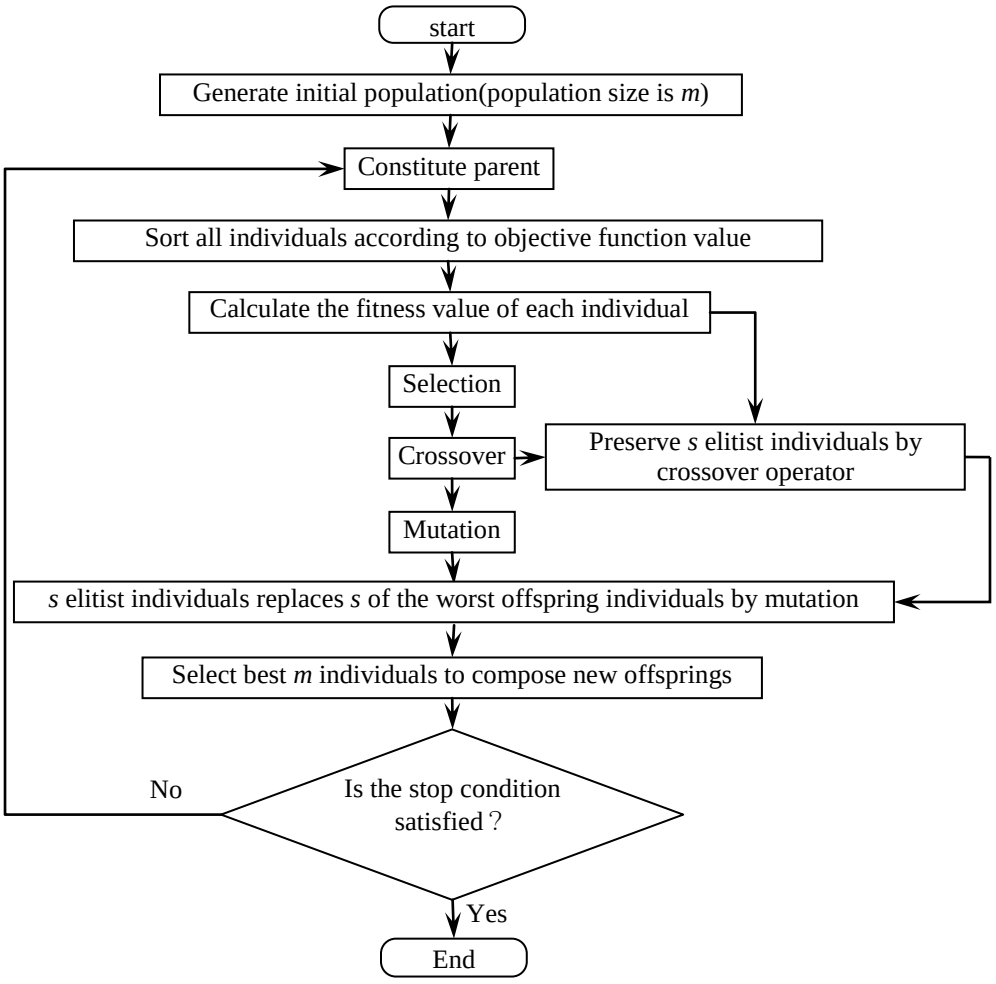

Figure 1. The Evolution Strategy Blocks Diagram Of Basic Genetic Algorithm

Figure 1 is based on the fact that preserving s elite individuals from all parents and offsprings by crossover operator and retaining elitist individuals in the new offspring population is better than using the elitist individuals of the parent population even if the elitist individuals in the parent population are destroyed in the process of crossovers.

3.1.1. Selection: The individuals in the population are expressed as $X_1^{(0)}, X_2^{(0)}, \cdots, X_i^{(0)}, \cdots, X_n^{(0)}$, $X_i^{(0)} = (x_{i1}^{(0)}, x_{i2}^{(0)}, \cdots, x_{id}^{(0)})$. In order to calculate the fitness values, the individuals are sorted in descending order according to objective function values. After sorting the individuals as $\bar{X}_1^{(0)}, \bar{X}_2^{(0)}, \cdots, \bar{X}_i^{(0)}, \cdots, \bar{X}_n^{(0)}$, and letting $\beta \in (0,1)$, the fitness value of $\bar{X}_i^{(0)}$ is computed as follows:

$$eval(\bar{X}_i^{(0)}) = \beta(1-\beta)^{i-1} \quad i = 1, 2, ..., n \tag{4}$$

Where, $eval(\bar{X}_i^{(0)})$ is the fitness value of i^{th} member in population, and $\beta \in (0,1)$ is a parameter usually chosen between 0.01 and 0.3 [12].

The fitness value of each individual is calculated according to equation (4), and then the roulette wheel method is used to pair members [1]. The roulette wheel method is as follows:

The selection probability of the i^{th} member is given by

$$P_i = \frac{eval(\bar{X}_i^{(0)})}{\sum_{i=1}^{n} eval(\bar{X}_i^{(0)})} \tag{5}$$

Letting

$$PP_0 = 0 \tag{6}$$

$$PP_i = \sum_{j=1}^{i} P_i, \quad i = 1, 2, \cdots, n \tag{7}$$

The roulette wheel is rotated up to n times, and a random number $\eta_k \in (0,1)$ is generated at each rotation. When this random number satisfies

$$PP_{i-1} \le \eta_k < PP_i \tag{8}$$

The i^{th} member is selected to take part in crossover.

3.1.2. Crossovers: Suppose the i^{th} and j^{th} individuals $\bar{X}_i^{(0)}$ and $\bar{X}_j^{(0)}$ are matched for crossover as discussed above. If their fitness values satisfy

$$eval(\bar{X}_i^{(0)}) > eval(\bar{X}_j^{(0)}) \tag{9}$$

Then, an offspring $X_j^{(1)}$ is generated by

$$X_j^{(1)} = \gamma \bar{X}_i^{(0)} + (1-\gamma)\bar{X}_j^{(0)} \tag{10}$$

Where

$$\gamma = \frac{eval(\bar{X}_i^{(0)})}{eval(\bar{X}_i^{(0)}) + eval(\bar{X}_j^{(0)})} \tag{11}$$

The second offspring $X_i^{(1)}$ is generated by

$$X_i^{(1)} = \bar{X}_i^{(0)} + \lambda(\bar{X}_i^{(0)} - X_j^{(1)}) \tag{12}$$

Where λ is a mapping coefficients, and is a constant greater than 0.

The relative positions of the offsprings individuals are shown in Figure 2.

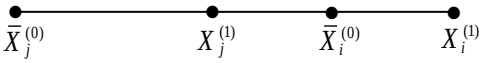

Figure 2. Relative positions of $\bar{X}_i^{(0)}$, $\bar{X}_j^{(0)}$ and $X_i^{(1)}$, $X_j^{(1)}$

The $\bar{X}_i^{(0)}$ and $\bar{X}_j^{(0)}$ cross to generate the two offspring individuals $X_i^{(1)}$ and $X_j^{(1)}$. The above crossover methods are repeated until all offspring individuals are gained.

3.1.3. Mutations: Let $a_i = (a_{i1}, a_{i2}, \cdots, a_{ij}, \cdots, a_{id})$ and $b_i = (b_{i1}, b_{i2}, \cdots, b_{ij}, \cdots, b_{id})$ be the upper and lower limits of $X_i (i = 1, 2, \cdots, n)$. Cross-generated the i^{th} individual $X_i^{(1)}$ is expressed as

$$X_i^{(1)} = (x_{i1}^{(1)}, x_{i2}^{(1)}, \cdots, x_{id}^{(1)}), i = 1, 2, \cdots, n \quad (13)$$

Cross-generated the i^{th} individual $X_i^{(1)}$ is mutated according to the following formula

$$\bar{x}_{ij}^{(1)} = \begin{cases} x_{ij}^{(1)} + r_{ij}(b_{ij} - x_{ij}^{(1)}) & r_{ij} > 0.5 \\ x_{ij}^{(1)} - r_{ij}(x_{ij}^{(1)} - a_{ij}) & r_{ij} < 0.5 \\ 0.5x_{ij}^{(1)} + 0.25(b_{ij} + a_{ij}) & r_{ij} = 0.5 \end{cases} \quad i = 1, 2, \cdots, n;\ j = 1, 2, \cdots, d \quad (14)$$

Where, r_{ij} is an obeying uniform distribution random number of [0,1] interval corresponding to the j^{th} component of the i^{th} individual in population.

3.2. Method of Generating the First Interior Point Based on Genetic Algorithm

When using intelligent optimization algorithms to solve constrained optimization problem, the problem of generating initial population is divided into two steps. The first step is to use the genetic algorithm to obtain the first interior point of the initial population; the second step is to generate the remaining feasible individuals of the initial population by iterations. The proposed method divides the problem of initial population generation into two steps:

The problem of generating the first interior point is converted into solving an unconstrained optimization problem according interior point method, and then the genetic algorithm is used to solve this unconstrained optimization problem, gains the first initial interior point. This is the key to subsequently generate the initial population. The generation method of the first interior point based on the genetic algorithm is as follows:

(1) Let the size of initial population be *m*. Then, *m* individuals are randomly generated with the i^{th} individual being $X_i^{(k)} = (x_{i1}^{(k)}, x_{i2}^{(k)}, \cdots, x_{ij}^{(k)}, \cdots x_{id}^{(k)})$. $a_i = (a_{i1}, a_{i2}, \cdots, a_{ij}, \cdots, a_{id})$ and $b_i = (b_{i1}, b_{i2}, \cdots, b_{ij}, \cdots, b_{id})$ are the upper and lower limits of the i^{th} individual, Let $k := 0$.

(2) For each individual, the indicator vectors S_k and T_k are generated by

$$T_k = \left\{ j \middle| g_j(X_i^{(k)}) > 0, \quad 1 \le j \le l \right\} \quad (15)$$

$$S_k = \left\{ j \middle| g_j(X_i^{(k)}) \le 0, \quad 1 \le j \le l \right\} \quad (16)$$

Where T_k is the vector of indices satisfying the constraint condition in formula (3); S_k is the vector of indices which does not meet the constraint condition in formula (3).

(3) Check whether S_k is the empty set, if so, stop the iterations. Else,

(4) Construct the objective function

$$f(X_i^{(k)}, r_t) = -\sum_{j \in S_k} g_j(X_i^k) + r_k \sum_{j \in T_k} \frac{1}{g_j(X_i^k)} \quad (17)$$

To solve the minimization problem

$$\min_{X \in R_k} f(X_i^{(k)}, r_k) \quad (18)$$

Where $R_k = \{X_i^{(k)} \mid g_j(X_i^{(k)}) > 0, j \in T_k\}$, $r_k = 1/k$.

The constructed objective function represents an unconstrained optimization problem. Then using genetic algorithm solves the unconstrained optimization problem, gains the first initial interior point.

(5) Use the roulette method to select individuals to participate in crossover.

(6) Mutate crossovered individuals according to mutation probability p_m.

(7) Let $k := k+1$, go to step (2).

3.3. Generation of Remaining Feasible Individuals in Initial Population

After the first interior point $\bar{X}_1^{(0)}$ is generated, the second initial individual $X_2^{(0)}$ is randomly generated according to

$$X_2^{(0)} = a_2 + c_2 .* (b_2 - a_2) \tag{19}$$

Where, $a_2 = \left(a_{21}, a_{22}, \cdots, a_{2j}, \cdots, a_{2d}\right)^T$, $b_2 = \left(b_{21}, b_{22}, \cdots, b_{2j}, \cdots, b_{2d}\right)^T$, $c_2 = \left(c_{21}, c_{22}, \cdots, c_{2j}, \cdots, c_{2d}\right)^T$; c_{2j} is a random number chosen from the uniform distribution in the [0,1] interval. The last term in Equation (19) represents pointwise multiplication.

$X_2^{(0)}$ is tested to find out whether it meets the constraint condition. If so, the next initial individual $X_3^{(0)}$ is generated. If not, $X_2^{(0)}$ is moved closer to $\bar{X}_1^{(0)}$ by

$$X_2^{(0)} = \bar{X}_1^{(0)} + \lambda(X_2^{(0)} - \bar{X}_1^{(0)}) \tag{20}$$

Where, $0 < \lambda < 1$, usually let $\lambda = 0.5$.

If $X_2^{(0)}$ is not sufficient to satisfy the constraint condition, the λ value is halved and the process is repeated until the constraint condition is satisfied, and let $\bar{X}_2^{(0)} = X_2^{(0)}$. The iterative procedure is shown in Figure 3.

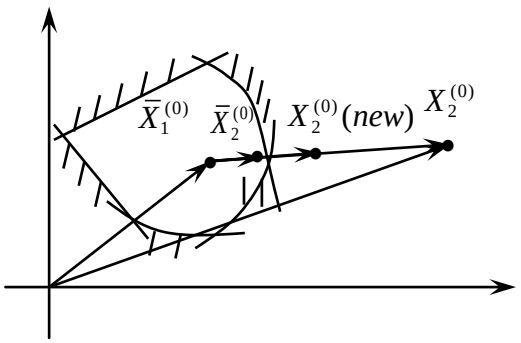

Figure 3. The Iterative Procedure to Make $X_2^{(0)}$ Feasible

The above procedure is continued until the whole population is created.

4. Simulations

4.1. Selected Test Functions

In order to validate the effectiveness of the method of generating initial population, the following three functions with complex constraints and multiple variables are chosen:

(1) Test function 1:

$$\begin{aligned}\min f_1(X) &= (x_1-10)^2+5(x_2-12)^2+x_3^4+3(x_4-11)^2+10x_5^6\\ &\quad +7x_6^2+x_7^4-4x_6x_7-10x_5-8x_7\\ s.t.&\begin{cases}127-2x_1^2-3x_2^4-x_3-4x_4^2-5x_5\ge 0\\ 282-7x_1-3x_2-10x_3^2-x_4+x_5\ge 0\\ 196-23x_1-x_2^2-6x_6^2+8x_7\ge 0\\ -4x_1^2-x_2^2+3x_1x_2-2x_3^2-5x_6+11x_7\ge 0\\ -10\le x_i\le 10, i=1,2,\cdots,7\end{cases}\end{aligned} \tag{21}$$

(2) Test function 2:

$$\begin{aligned}\max f_2(X) &= x_1^2+x_2^2+x_1x_2-14x_1+16x_2+(x_3-10)^2+4(x_4-5)^2+(x_5-3)^2\\ &\quad +2(x_6-1)+5x_7^2+7(x_8-11)^2+2(x_9-10)^2+(x_{10}-7)^2+45\\ s.t.&\begin{cases}105-4x_1-5x_2+3x_7-9x_8\ge 0\\ -3(x_1-2)^2-4(x_2-3)^2-2x_3^2+7x_4\ge -120\\ -10x_1+8x_2+17x_7-2x_8\ge 0\\ -x_1^2-2(x_2-2)^2+2x_1x_2-14x_5+6x_6\ge 0\\ 8x_1-2x_2-5x_9+2x_{10}+12\ge 0\\ -5x_1^2-8x_2-(x_3-6)^2+2x_4+40\ge 0\\ 3x_1-6x_2-12(x_9-8)^2+7x_{10}\ge 0\\ -0.5(x_1-8)^2-2(x_2-4)^2-3x_5^2+x_6\ge -30\\ -10\le x_i\le 10, i=1,2,\cdots,10\end{cases}\end{aligned} \tag{22}$$

(3) Test function 3:

$$\begin{aligned}\max f_3(X) &= x_1(x_1x_4+x_2x_5+x_3x_6)+x_2(x_1x_7+x_2x_8+x_3x_9)+x_3(x_1x_{10}+x_2x_{11}+x_3x_{12})\\ s.t.&\begin{cases}x_1+x_2+x_3\le 250\ 000\\ x_1x_4+x_2x_7+x_3x_{10}\le 1\ 500\ 000\\ x_1x_5+x_2x_8+x_3x_{11}\le 1\ 100\ 000\\ x_1x_6+x_2x_9+x_3x_{12}\le 950\ 000\\ x_1\le 260\ 000\\ x_2\le 260\ 000\\ x_3\le 260\ 000\\ x_j\ge 0\quad j=1,2,\cdots,12\end{cases}\end{aligned} \tag{23}$$

In simulations, the proposed method was investigated in comparison to the method of randomly generated initial population and the method of initial population generation discussed in [9]. The ranges of the variables in functions f_1 and f_2 are given in formulae

(17) and (18), and the ranges of the variables in function f_3 are shown in Table 1. With the population size chosen as 100, the average running times of the methods are shown in Table 2.

Table 1. Variable Value Range of Function f_3

Variables	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
Lower limit	0	0	0	0	0	0	0	0	0	0	0	0
Upper limit	260000	260000	260000	200	200	200	200	200	200	200	200	200

Table 2. The Average Running Time in Seconds Of Different Methods

Method	Randomly generated initial population method	Initial population method in literature [9]	Proposed method
f_1	4.0283	1.8239	0.0448
f_2	950.5148	24.3619	0.1336
f_3	2095.4687	45.8908	0.4376

Table 2 shows that the proposed method reduces the average running time by 98.8879% for f_1, 99.9859% for f_2, 99.9791% for f_3 in comparison to randomly generated initial population, and reduces the average running time by 97.5437% for f_1, 99.4516% for f_2, 99.0464% for f_3 in comparison to Initial population method in literature [9]. It can be seen that the proposed method is much faster to generate the initial population as compared to the other two methods. The difference in speed becomes more serious as the problem size and complexity increases. When the constraint conditions and variables are very many in a global optimization application, the proposed method

5. Conclusions

When using an intelligent optimization algorithm to solve a constrained optimization problem, if the constraint conditions and the number of variables are very many, the random method and the method of reference [9] take a long time, or may even be unable to generate the initial population. Although the method of reference [10] can quickly generate the initial population, constraint conditions are required to be derivable, thus having limitations. The proposed method in this paper is capable of quickly generating the initial population without having limitations. The proposed method divides the problem of initial population generation into two steps: The first step is to use the genetic algorithm to obtain the first interior point of the initial population; the second step is to generate the remaining feasible individuals of the initial population by iterations.

The paper gives a generation method of the first initial interior point. Namely, the problem of generating the first initial interior point is converted into solving an unconstrained optimization problem. The genetic algorithm is used to solve this unconstrained optimization, resulting in the first initial interior point. The generation method of the first initial interior point is key to this method.

The paper gives a generation method of the remaining individuals of the initial population. Namely, the remaining individuals are first randomly generated; then the feasibility of each individual is checked. If an individual is feasible, it is accepted. If not, it is gradually moved closer to the first initial interior point, until it becomes feasible. This procedure is repeated until the initial population is completed. This method can quickly generate the remaining individuals of the initial population.

The experimental results with the three test functions chosen show that the proposed method is much faster to generate the initial population as compared to the method of randomly generating initial population and the method given in reference [9].

Acknowledgments

The paper is supported by sub project of national science and technology support program under Grant 2014BAD06B01-23, sub project of public welfare industry (Agriculture) research project (Grant No. 201503116-04-01).

References

[1] D. Wang, J. Wang and Hongfeng Wang, "Intelligent optimization method", Higher education press, Beijing, **(2007)**.
[2] K. Mario, s. Gerald and A. Ajith, "Intelligent computational optimization in engineering: techniques & applications", Springer Berlin Heidelberg, Berlin, **(2011)**.
[3] Y. Mo, H. Liu and Q. Wang, "Intelligent optimization algorithm in teaching and research", Science and Technology Innovation Herald, no. 13, **(2008)**, pp. 2-3.
[4] X. Liang and M. Huang. "Modern intelligent optimization hybrid algorithm and its application", Publishing house of electronics industry, Beijing, **(2012)**.
[5] D. Sharma, K. Deb and N. N. Kishore, "Domain-specific initial population strategy for compliant mechanisms using customized genetic algorithm", Structural and Multidisciplinary Optimization, vol. 43, no. 4, **(2011)**, pp. 541-554.
[6] Y. Zhou, "Hybrid strategy of intelligent optimization algorithm: analysis, design and model", Application Research of computer, vol. 27, no. 12, **(2010)**, pp. 4423-4426.
[7] C. Blum and A. Roli, "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison", ACM computing survey, vol. 35, no. 3, **(2003)**, pp. 268-303.
[8] V. Toğan and A. T. Daloğlu, "An improved genetic algorithm with initial population strategy and self-adaptive member grouping", Computers and Structures, vol. 86, no. 11, **(2008)**, pp. 1204-1218.
[9] F. Wang, C. Wu and H. Yang, "Study on the productive method on the initial population by using genetic algorithm to solve the constrained optimization problems", Journal of Northeast Agricultural University, vol. 35, no. 5, **(2004)**, pp. 608-611.
[10] Me. Xu, S. Wen and F. Wang, "Improved method on generation of initial population by using GA for solving constrained optimization problems", Journal of Northeast Agricultural University, vol. 45, no. 7, **(2014)**, pp. 104~107.
[11] Operations Research Teaching Materials Writing Group. "Operations Research", Tsinghua university press, Beijing, **(2012)**.
[12] F. Wang, J. Wang and C. Wu, "The improved research on actual number genetic algorithm", Journal of Biomathematics, vol. 21, no. 1, **(2006)**, pp. 153-158.

Authors

Jiquan Wang, he received the BE degree in Agricultural electrification and automation from the Northeast Agricultural University, China, in 1996, the ME degree in Agricultural electrification and automation from the Northeast Agricultural University, China, in 2004, and the Ph.D. degree in Agricultural Equipment Engineering Technology from Shenyang Agricultural University, China, in 2011. I am currently an associate professor with the Engineering College, Northeast Agricultural University. My current research interests include Genetic algorithm theory and its application, Neural Network theory and its application, Parallel Computing and image recognition. I have published over 40 papers in domestic and international academic journals and conference proceedings.

Okan K. Ersoy received the BSEE from Robert College in 1967, the MSEE degree from University of California at Los Angeles in 1968, Ph.D. degree from University of California at Los Angeles. He is currently a professor of Electrical and Computer Engineering, Purdue University. His current research interests include Neural Networks, image processing and imaging, networking and information processing, genetic algorithm, decision trees and support

vector. He has published over 200 papers international academic journals and conference proceedings.

Xinxin Chen, he received the BE degree in Industry Engineering, Northeast Agricultural University, China in 2016. He is currently a postgraduate students of Engineering College, Northeast Agricultural University. His current research interest includes Genetic algorithm theory and its applications.

Fulin Wang received the BE degree in Agricultural mechanization and automation from the Northeast Agricultural University, China, in 1981, the ME degree in Agricultural mechanization and automation from the Northeast Agricultural University, China, in 1988, and the Ph.D. degree in Agricultural mechanization and automation from Northeast Agricultural University, China, in 1993. He is currently a professor with the Engineering College, Northeast Agricultural University. His current research interests include Genetic algorithm theory and its application, Neural Network theory and its application, Parallel Computing and image recognition. He has published over 100 papers in domestic and international academic journals and conference proceedings.