ISSN 1749-3889 (print), 1749-3897 (online)
International Journal of Nonlinear Science
Vol.15(2013) No.3,pp.212-219

Solving the Constrained Nonlinear Optimization based on Imperialist Competitive Algorithm

Mahdi Abdollahi[1] *, Ayaz Isazadeh[2], Davoud Abdollahi[3]
[1] University of Tabriz, Aras International Campus, Department of Computer Sciences, Tabriz, Iran
[2] University of Tabriz, Department of Computer Sciences, Tabriz, Iran
[3] University College of Daneshvaran, Department of Mathematics, Tabriz, Iran
(Received 24 December 2012, accepted 24 April 2013)

Abstract: In this work a strong framework is presented for solving the constrained nonlinear optimization problem that is a relatively complicated problem. These problems arise in a diverse range of sciences. There are a number of different approaches have been proposed. In this work, we employ the imperialist competitive algorithm (ICA) for solving constrained nonlinear optimization problems. Some well-known problems are presented to demonstrate the efficiency of this new robust optimization method in comparison to other known methods.

Keywords: ICA; constrained nonlinear optimization; evolutionary multi-objective optimization; meta-heuristics

1 Introduction

Constrained Nonlinear Optimization problem (CNOP) arise via a diverse range of sciences such as economics, engineering, chemistry, mechanics, medicine and especially in management. A constrained optimization problem is usually written as a nonlinear programming (NLP) problem of the following form:

$$minimize\ f(X),$$

$$s.t. \begin{cases} g_l(X) \geq 0, & l = 1, \ \ldots, \ K; \\ h_j(X) = 0, & j = 1, \ \ldots, \ M; \\ X \in R^n, \ x_i^{low} \leq x_i \leq x_i^{up}, & i = 1, \ \ldots, \ N. \end{cases} \tag{1}$$

In the above NLP problem, there are N variables and the function $f(x)$ is the objective function. $g_l(X)$ is the $l-th$ constraints of K inequality constraints and $h_j(X)$ is the $j-th$ constraints of M equality constraints. A selection like $X \in R^n$ which satisfy all the above constraints and variable bounds is called a feasible solution, others are infeasible ones. However, a number of different approaches have been proposed such as R. Le Riche and C. Knopf [18] used double penalty strategy and other methods such as Differential Evolution (DE) [12, 13], Genetic Algorithm (GA) [8], Stochastic Ranking (SR) [17], the Extended Hybrid Genetic Algorithm (EHGA) [14], Particle Swarm Optimization (PSO) algorithms [18], [7], high penalty [21], lower penalty [21], neural network [15] and Greedy Evolution [5].

In this paper, we introduce Imperialistic Competition Algorithm (ICA) for solving the constrained nonlinear optimization problem. The results of ICA are compared with other methods found in [2, 5, 7, 8, 18,19,21,] and [15].

In section 2, we will briefly overview the ICA, and in section 3, how to apply ICA for solving a constraint nonlinear optimization problem. In section 4, the obtained numerical results will be presented as a comparison and finally in section 5, we have the conclusions.

* **Corresponding author**. E-mail address: *m.abdollahi89@ms.tabrizu.ac.ir*

IJNS.2013.06.15/718

2 Imperialist Competitive Algorithm

Like other evolutionary algorithms, the proposed algorithm starts with an initial population as initial country. Some of the best countries among the population are selected to be the imperialists. The rest of the population is divided among the mentioned imperialists as colonies. Then the imperialistic competition begins among all the empires. The weakest empire will be eliminated from the competition. As a result, all colonies move toward their relevant imperialists along with competition among empires. Finally, the collapse mechanism will cause all the countries to converge to a state which there exists just one empire in the domain of the problem, and all the other countries are colonies of that one empire. The robust empire would be our optimal solution [1].

3 Solving the Constrained Nonlinear Optimization Problem

Let the form of the problem as (1). In order to transform it to a form which is possible to apply ICA, we will use slack variables in the inequality constraints to make them equality constraints, so we will have the following problems:

$$minimize\ f(X),$$
$$s.t. \begin{cases} g_l(X) - y_l = 0, & y_l \in R,\ l = 1,\ \ldots,\ K; \\ h_j(X) = 0, & j = 1,\ \ldots,\ M; \\ x_i^{low} \leq x_i \leq x_i^{up}, & i = 1,\ \ldots,\ N,\ X \in R^n. \end{cases} \tag{2}$$

Which all the slack variables (y_l) are non-negative variables and $x^{low},\ x^{up}$ are the lower and upper bounds that are posed on the variables of the problem. In order to use ICA for (2), we will use the auxiliary function:

$$min\ F(\bar{X}) = f^2(X) + \sum_{l=1}^{K}(g_l(X) - y_l)^2 + \sum_{j=1}^{M} h_j^2(X) \tag{3}$$
$$X \in R^N,\ y_l \in R\ \ (l = 1,\ 2,\ \ldots,\ K),\ \bar{X} \in R^{N+K}.$$

Therefore, our problem should be solved in R^{N+K}. In the rest of the paper, the minimization problem will be considered:

$$\begin{array}{l} min\ F(\bar{X}) \\ subject\ to \\ \bar{X} \in D = \{(X,Y)|X \in R^N\ and\ Y \in R^K;\ x^{low} \leq x \leq x^{up},\ 0 \leq Y\}. \end{array} \tag{4}$$

where the $F(\bar{X})$ is the new objective function and our original objective function $f(X)$ is on of its terms. Then, we generate our countries which are the randomized solutions as population [1]. In an $N + K$-dimensional problem, a country is an $1 \times (N + K)$ array defined as follow:

$$country = (x_1,\ x_2,\ \ldots,\ x_N,\ x_{N+1},\ x_{N+K}).$$

We should generate N_{pop} of them. The cost of each country is the cost of $F(\bar{X})$ at the variables $(x_1,\ x_2,\ \ldots,\ x_{N+K})$, then

$$Cost = F(country) = F(x_1,\ x_2,\ \ldots,\ x_{N+K}) \tag{5}$$

4 Experiment and results

In this section, we have investigated the performance of ICA with eight benchmark functions.

Test 1: [19]

$$\begin{array}{ll} min\ f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \\ s.t. & x_1^2 + x_2^2 \geq 0.25 \\ & \frac{1}{3}x_1 + x_2 \geq -0.1 \\ & -\frac{1}{3}x_1 + x_2 \geq -0.1 \\ & -10 \leq x_1 \leq 10,\ 0 \leq x_2 \leq 10 \end{array}$$

The solution is $f(1,\ 1) = 0$. Apply ICA to optimize it with 1000 iterations, with parameters are shown in Table 1.

Table 1: Parameters used in Test Problems

Parameter	Test 1	Test 2	Test 3	Test 4
The number of countries	1000	100	500	50
Empires	60	10	50	5
Revolution Rate	0.3	0.3	0.3	0.3
Iteration	1000	35	300	100
ζ	0.02	0.02	0.02	0.02
θ	0.5	0.5	0.5	0.5
β	2	2	2	2

Table 2: Results of GE, DE, PSO and ICA

Test Problem	Methods	Best Solution
Test 2	Greedy Evolution	0.25
	Differential Evolution	0.25
	PSO	0.25
	ICA	0.25

where ζ is a position coefficient, and to get different points around the imperialist we have to add a random amount of deviation to the direction of movement like θ, and β causes the colony to get closer to the imperialist in both direction. Figure 1 shows the convergence history of Test 1.

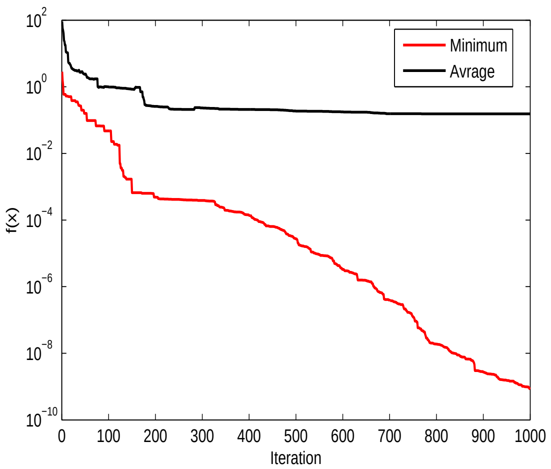

Figure 1: The convergence history of Test 1.

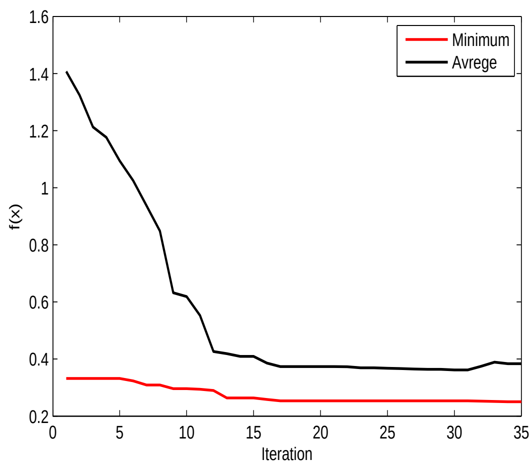

Figure 2: The convergence history of Test 2.

Test 2: [5]

$$\begin{aligned} &min\ f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \\ &s.t. \quad -x_1 - x_2^2 \le 0 \\ &\qquad -x_1^2 - x_2 \le 0 \\ &\qquad -0.5 \le x_1 \le 0.5,\ x_2 \le 1.0 \end{aligned}$$

where $min\ f(0.5,\ 0.249998) = 0.25$. The optimal solution found by ICA is $x = (0.5000,\ 0.2499)$. The results of Greedy Evolution [5], Differential Evolution [12, 13], PSO [9, 4, 10] and ours are shown in Table 2.

The convergence history of Test 2 is shown in Figure 2.

Table 3: Results of GA methods and ICA

Test Problem	Methods	Best Solution
Test 3	double penalty strategy [11]	0.998
	two-step selection [9]	0.987
	High penalty [21]	1.00
	Lower penalty [21]	0.87
	MPGA [21]	0.998
	ICA	1.00

Test 3: [21]

$$
\begin{aligned}
& max\ f(x) = (100 - (x_1-5)^2 - (x_2-5)^2 - (x_3-5)^2)/100 \\
& s.t. \quad (x_1-p)^2 - (x_2-q)^2 - (x_3-r)^2 \leq 0.25 \\
& \qquad for \quad p,\ q,\ r = 1,\ 3,\ 5,\ 7,\ 9 \\
& \qquad 0 \leq x_i \leq 10, \quad (1 \leq i \leq 3)
\end{aligned}
$$

The solution is $max\ f(5,\ 5,\ 5) = 1$. The optimal solution found by ICA is x=(4.9837, 5.0145, 5.0026). The results of Genetic Algorithm methods [11], [9] and [21] and ICA are shown in Table 3. with the convergence history shown in Figure 3.

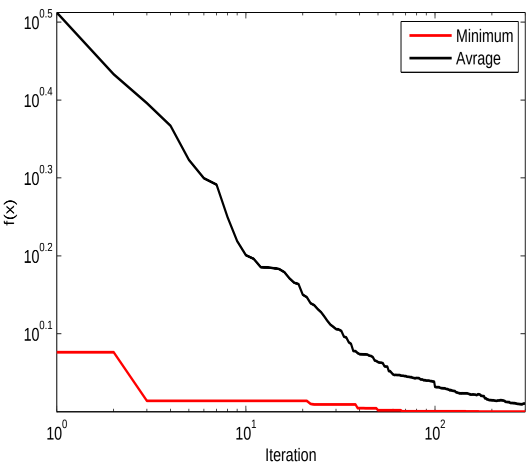

Figure 3: The convergence history of Test 3.

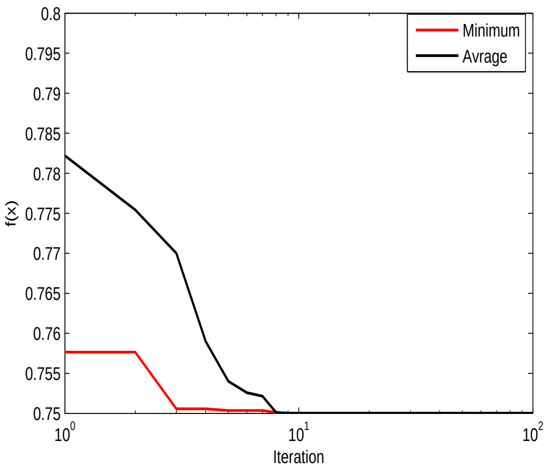

Figure 4: The convergence history of Test 4.

Test 4: [7]

$$
\begin{aligned}
& min\ f(x) = x_1^2 + (x_2-1)^2 \\
& s.t. \quad h(x) = x_2 - x_1^2 = 0 \\
& \qquad -1 \leq x_1 \leq 1, \quad -1 \leq x_2 \leq 1
\end{aligned}
$$

The solution of this function is $min\ f(\pm\frac{1}{\sqrt{2}},\ \frac{1}{2}) = 0.75$. The optimal solution found by ICA is $x = (-0.7071,\ 0.5000)$. The results of the homomorphous maps (HM), stochastic ranking (SR), constraint-handling mechanism PSO (CPSO), improved particle swarm optimization (IPSO) and ICA are comparable in Table 4. See Figure 4 too.

Now, we analyze four other problems as case studies. The used parameters for solving the following cases are shown in Table 5.

Table 4: Results of Test 4

Test Problem	Methods	Best Solution
	HM [16]	0.750000
	SR [17]	0.750000
Test 4	CPSO [7]	0.749999
	IPSO [7]	0.750000
	ICA	0.750000

Table 5: Parameters used in Case Problems

Parameter	Case 1	Case 2	Case 3	Case 4
The number of countries	300	1000	100	100
Empires	20	50	10	10
Revolution Rate	0.3	0.01	0.3	0.3
Iteration	200	1000	150	50
ζ	0.02	0.02	0.02	0.02
θ	0.5	0.5	0.5	0.5
β	2	2	2	2

Case 1: [18]

$$\begin{aligned} & max\ f(x) = sin^3(2\pi x_1) sin^3(2\pi x_2) \\ & s.t. \quad x_1^2 - x_2 + 1 \le 0 \\ & \qquad 1 - x_1 + (x_2 - 4)^2 \le 0 \\ & \qquad 0 \le x_1 \le 10,\ \ 0 \le x_2 \le 10 \end{aligned}$$

The ICA method got $x = (1.249981704542397,\ 4.250005076533529)$. The results of the genetic algorithm (GA) [20, 3, 6], competition PSO (CPSO) [18] and ICA are shown in Table 6. Figure 5 shows the convergence history of Case 1.

Case 2: [8]

$$\begin{aligned} & min\ f(x) = epx(x_1 x_2 x_3 x_4 x_5) \\ & s.t. \quad x_1^2 x_2^2 x_3^2 x_4^2 x_5^2 = 10 \\ & \qquad x_2 x_3 - 5 x_4 x_5 = 0 \\ & \qquad x_1^3 + x_2^3 = -1 \\ & \qquad -2.3 \le x_i \le 2.3,\ \ i = 1,\ 2 \\ & \qquad -3.2 \le x_i \le 3.2,\ \ i = 3,\ 4,\ 5 \end{aligned}$$

The best solutions obtained by the ICA method has been listed in Table 7 and compares them with the genetic algorithm (GA) [8]. It is obvious from Table 7 that the result of the ICA method is better than GA.

Figure 6 shows the convergence history of Case 2.

Table 6: Results of Case 1

Problem	Methods	Best Solution
	GA	0.95825
Case 1	CPSO	0.9999
	ICA	0.999999978

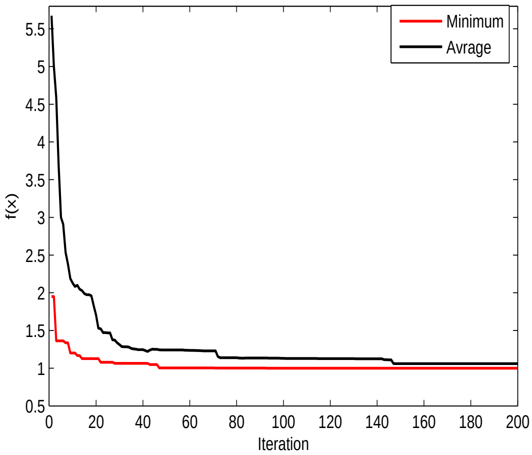

Figure 5: The convergence history of Case 1.

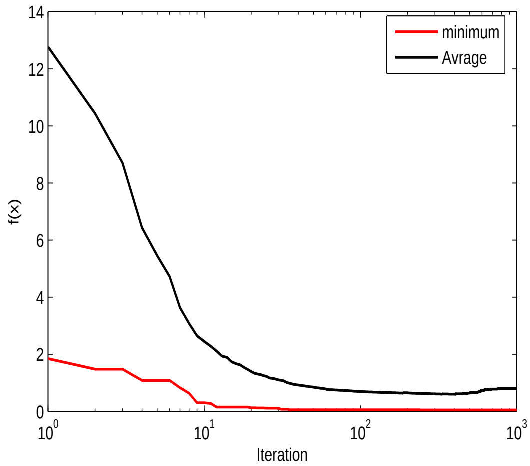

Figure 6: The convergence history of Case 2.

Table 7: Results of Case 2

Problem	Methods	$(x_1, ..., x_5)$	$f(x)$
Case 2	GA	-1.717143 1.595709 1.827247 -0.7636413 -0.7636450	0.053950
	ICA	-1.716945 1.595806 -1.830147 -0.766524 0.766855	0.052467

Table 8: Results of Case 2

Problem	Methods	Best Solution
Case 3	EHGA	31.8508
	LUDE	40.56241
	ICA	40.56241

Case 3: [2]

$$\begin{aligned} & max\ f(x) = 4x_1^3 - 3x_1 + 6x_2 \\ & s.t. \quad -x_1 - x_2 + 4 \geq 0 \\ & \qquad -x_1 + 4x_2 \geq 0 \\ & \qquad 2x_1 + x_2 - 5 = 0 \\ & \qquad 0 \leq x_i \leq 100, \quad i = 1,\ 2 \end{aligned}$$

The optimal solution found by ICA is $x = (2.2222,\ 0.5555)$. The results of the extended hybrid genetic algorithm (EHGA) [14], the line-up differential evolution algorithm (LUDE) [2] and ICA are compared in Table 8.

The convergence history of Case 3 are shown in figure 7 .

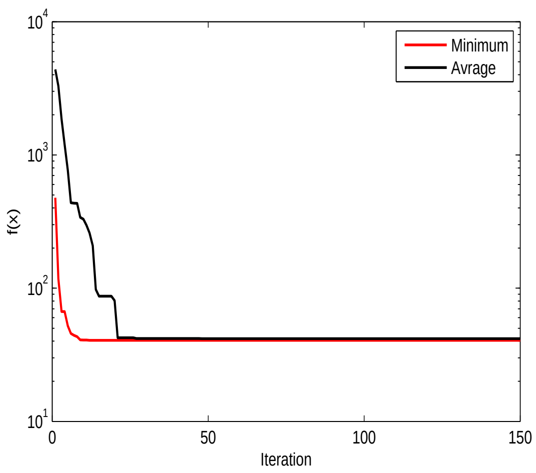

Figure 7: The convergence history of Case 3.

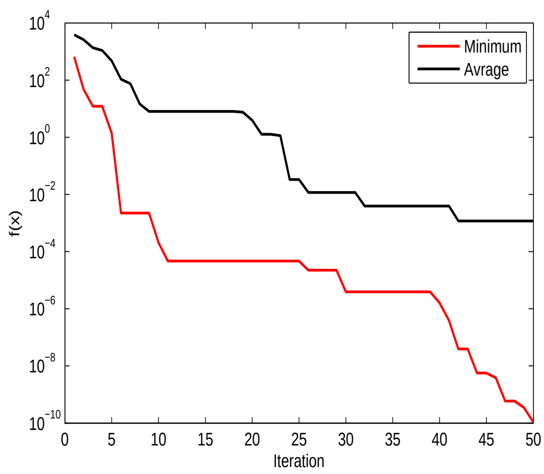

Figure 8: The convergence history of Case 4.

Case 4: [15]

$$\begin{aligned} & min\ f(x) = (\lambda^2 - \lambda - 3)x_1 + (\lambda^2 + 2\lambda - 6)x_2 \\ & s.t. \quad x_1 \leq 4 + \lambda^2 \\ & \qquad 3x_1 + 2x_2 \leq 18 - 2\lambda^2 \\ & \qquad x_1,\ x_2 \geq 0 \\ & \qquad \lambda \in [0,\ 3] \end{aligned}$$

The optimal solution is $x = (0,\ 0)$. The ICA finds it the same as the neural network method in [15]. Figure 8 shows the convergence history of Case 4.

5 Conclusions

In this paper, a new efficient approach for solving the constrained nonlinear optimization problem was presented. We transformed the constrained nonlinear optimization into a multi-objective optimization problem. The goal was to obtain the error as close to zero as possible for each of the involved objectives. Some well known problems were presented to demonstrate the efficiency of finding the best optimal solution using the Imperialist Competitive Algorithm (ICA). We should mention that when the optimal solution is negative, our method give the positive form of it. Because we use the square of the main objective function in the form (3). It doesn't matter, to find out the sign of your main optimal solution, you have to put values of your selection $X \in R^N$ in the main objective function to make sure.

References

[1] E. Atashpaz-Gargari and C. Lucas, Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition, *IEEE Congress on Evolutionary Computation.*, CEC(2007): 4661-4667.

[2] H. Sarimveis and A. ikolakopoulos, A line up evolutionary algorithm for solving nonlinear constrained optimization problems, *ELSEVIER Computers & Operations Research.*, 32(2005): 1499-1514.

[3] J. F. Tang, D. Wang and et al, A hybrid genetic algorlthm for a type of nonlinear programming problem, *Computers Math. Apphe.*, 36(1998): 11-21.

[4] J. H. Kim and H. Myung, Evolutionary Programming Techniques for Constrained Optimization Problems, *IEEE Transactions on Evolutionary Computation.*, 1(1997): 129-140.

[5] J. Si, K. Chen, S. Zhang and et al, Solving the Constrained Nonlinear Optimization based on Greedy Evolution Algorithm, *IEEE Seventh International Conference on Natural Computation.*, 2(2011): 1121-1125.

[6] J. Su, A. Hu and Z. He, Solving a kind of nonhnear programming problems via analog neural networks, *Neurocomputing.*, 18(1998): 1-9.

[7] J. Zheng, Q. Wu and W. Song, An Improved Particle Swarm Algorithm for Solving Nonlinear Constrained Optimization Problems, *IEEE Third International Conference on Natural Computation (ICNC).*, 4(2007): 112-117.

[8] K. Deb, An efficient constraint handling method for genetic algorithms, *ELSEVIER Comput. Methods Appl. Mech. Engrg.*, 186(2000): 311-338.

[9] K. Deb, Efficient Constraint Handling Method for Genetic Algorithms, *Computer Methods in Applied Mechanics and Engineering.*, 186(2000): 311-338.

[10] L. B. Zhang, C. G. Zhou, X. H. Liu, M. Ma, Y. H. Lv and Z. Q. Ma, A Novel Evolutionary Algorithm for Solving Constrained Optimization Problems, *Journal of Jilin University (Science Edition).*, 42(2004): 534-540.

[11] R. Le. Riche, C. Knopf-Lenoir and R. T. Haftka, A Segregated Genetic Algorithm for Constrained structural Optimization, *Proceedings of the 6th International Conference on GA, San Francisco, USA.*, 6(1995): 558-565.

[12] R. Storn and K. Price, DE - A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Space, *International Computer Science Institute (ICSI), Technical Repor.*, 11(1997): 341359.

[13] R. Storn and K. Price, Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, *Journal of Global Optimization.*, 11(1997): 341-359.

[14] R. Y. K. Fung, J. Tang and D. Wang, Extension of a hybrid genetic algorithm for nonlinear programming problems with equality and inequality constraints, *Computers and Operations Research.*, 29(2002): 261-274.

[15] S. Effati and M. Jafarzadeh, A new nonlinear neural network for solving a class of constrained parametric optimization problems, *ELSEVIER Applied Mathematics and Computation.*, 186(2007): 814-819.

[16] S. Koziel and Z. Michalewicz, Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization, *Evolutionary computation.*, 1(1997): 19-44.

[17] T. P. Runarsson and X. Yao, Stochastic Ranking for Constrained Evolutionary Optimization, *IEEE Transactions on Evolutoinary Computation.*, 4(2000): 284-294.

[18] Y. Dong, J. Tang, B. Xu and D. Wang, An Application of Swarm Optimization to Nonlinear Programming, *ELSEVIER An International Journal computers and mathematics with applications.*, 49(2005): 1655-1668.

[19] Y. H. Gu and Z. Y. Wu, A new filled function method for nonlinear integer programming problems, *ELSEVIER Applied Mathematics and Computation.*, 173(2006): 938-950.

[20] Y. Takao, M. Gen, T. Takeaki and Y. Li, A method for interval 0-1 number non-hnear programming problems using genetic algorithm, *Computers and Industmal Engmeemng.*, 29(1995): 531-535.

[21] Y. Wul, J. Lul and Y. Sunl, An Improved Multi-Population Genetic Algorithm for Constrained Nonlinear Optimization, *IEEE Proceedings of the 6th World Congress on Intelligent Control and Automation.*, 1(2006): 1910-1914.