Comparison and Analysis of the Selection Mechanism in the Artificial Bee Colony Algorithm

Li Bao
Complex System and Computational Intelligence Laboratory, Taiyuan University of Science and Technology
Taiyuan, Shanxi, PR. China, 030024
libao@sina.com

Jian-chao Zeng
Complex System and Computational Intelligence Laboratory, Taiyuan University of Science and Technology
Taiyuan, Shanxi, PR. China, 030024
zengjianchao@263.net

Abstract—**Artificial bee colony (ABC) algorithm is a new global stochastic optimization algorithm based on the particular intelligent behavior of honeybee swarms, in which there exists many issues to be improved and solved. When onlooker bees exploit in ABC algorithm, they choose food source depending on the strategy of proportional selection that can result in the premature of the evolutionary process. In this paper, in order to improve the population diversity and avoid the premature, several selection strategies, such as disruptive selection strategy, tournament selection strategy and rank selection strategy, are compared and analyzed through simulation, and the results show that the modified algorithm outperforms the basic ABC algorithm.**

Keywords-artificial bee colony algorithm; selection strategy; premature convergence; population diversity

I. INTRODUCTION

Artificial bee colony (ABC) algorithm, a new swarm intelligent algorithm, was proposed by Karabog [1] in Erciyes University of Turkey in 2005. Since ABC algorithm is simple in concept, easy to implement, and has fewer control parameters, it has been widely used in many fields. ABC algorithm has applied successfully to unconstrained numerical optimization problems [2-4]. Karaboga and Basturk [5] proposed the extended version of the ABC for solving constrained optimization problems in 2007. The experiments show that the extended version of ABC algorithm has better performance than DE and PSO. It can efficiently solve constrained optimization problems. ABC algorithm was applied to train artificial neural networks [6-8]. Singh [9] proposed an artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem in 2009. DING Haijun et al. [10] proposed a modified ABC algorithm adapted to combinatorial optimization problems, which was used to solve TSP problems. Nowadays, ABC algorithm is being studied on hybrid algorithm, Multi-objective optimization, integer programming, engineering application and data mining techniques.

The rest of paper is organized as follow. Section 2 introduces ABC algorithm. In order to improve the performance of ABC algorithm, modified ABC algorithms based on three different selection strategies are proposed in section 3. The simulation results obtained are presented and discussed in section 4. Finally, the conclusion is in section 5.

II. ARITIFICIAL BEE COLONY ALGORITHM

A. Description of Honeybees Behaviours

The minimal model of foraging selection that leads to the emergence of collective intelligence of honey bee swarms consists of three essential components: food sources, employed foragers and unemployed foragers. There are two basic behaviors: recruitment to a food source and the abandonment of a food source [1].

1) Food sources: it represents a position of solution of optimization problem, the profitability of food source are expressed as fitness of the solution.

2) Unemployed foragers: there are two types of them, scouts and onlookers. Their main task is exploring and exploiting food source. At the beginning, there are two choices for the unemployed foragers: (i). it becomes a scout – randomly search new food sources around the nest; (ii). It becomes a onlooker – determine the nectar amount of food source after watching the waggle dances of employed bee, and select food source according to profitability.

3) Employed foragers: the honeybees found food source, which also known as the employed bees, are equal to the number of food sources. The employed bees store the food source information and share with others according to a certain probability. The employed bee will become a scout when food source has been exhausted.

B. Description of Aritificial Bee Colony Algorithm

In ABC algorithm, artificial honeybees consist of employed bees, onlookers and scouts, among which the number of employed bees and onlookers are equal. The process of bees look for food source is the process of find the optimum solution, each solution of optimization problem is considered as a food source position in the search space, the fitness of solution represents the profitability of food source. Solution Numbers (SN) is equal to the number of employed bees or the onlookers. First of all, the ABC randomly generates initial SN solutions of population, each x_i (i = 1, 2,..., SN) is a vector of D dimension. Then honeybees

978-0-7695-3745-0/09 $25.00 © 2009 IEEE
DOI 10.1109/HIS.2009.319

repeatedly search for all food sources, iterations of which is set to C (C=1, 2, ... , MCN). MCN is maximum number of cycle. Employed bee start neighborhood search firstly, the position of new sources replace the previous one if better than previous position, otherwise keep the position of previous one. After all the employed bees complete search process, they share the nectar information of the food sources with the onlooker bees on the dance area. An onlooker bee evaluates the nectar information taken from all employed bees, and then it chooses a food source by using a selection probability. The higher the fitness of solution is, the higher selection probability is. After selecting the food source, onlookers start to carry out the exploitation process.

Employed bees and onlookers produce a candidate position by Formula (1):

$$v_{ij} = x_{ij} + r_{ij}(x_{ij} - x_{kj}) \qquad (1)$$

where $k \in \{1, 2, \ldots, SN\}$ and $j \in \{1, 2, \ldots, D\}$ are randomly chosen indexes. Moreover, $k \neq i$. r_{ij} is a random number between [-1,1] .

In ABC algorithm, "*limit*" is an important control parameter, it controls the times of updates of a certain solution. If a solution cannot be improved further through a predetermined number of cycles called limit then that solution is assumed to be abandoned, and the employed bee becomes a scout. If the solution is x_i and $j \in \{1, 2, \ldots, D\}$ to be abandoned, then a new solution produced randomly would replace x_i by formula (2):

$$x_{ij} = x_{\min}^{j} + rand(0,1)(x_{\max}^{j} - x_{\min}^{j}) \qquad (2)$$

where $x_{\min}^{j}$ is the lower bound of the parameter j and $x_{\max}^{j}$ is the upper bound of the parameter j.

C. Selection Strategy of Aritificial Bee Colony Algorithm

In the exploitation process of onlooker bees in ABC algorithm, onlooker bees choose food source by a stochastic selection scheme. The selection strategy consists of three steps: first, calculating fitness value; second, calculating selection probability for each solution in population; third, selecting candidate solution according to selection probability by "roulette wheel selection" method, and then starting neighborhood search.

Fitness function is considered as evidence to distinguish the good or bad individuals. Selection a good fitness function is very important which can keep population diversity and avoid premature convergence. Fitness function F(x) is usually a mathematical transformation from objective function f(x). In ABC algorithm, the fitness function is defined as follows:

$$fit_i = \begin{cases} \dfrac{1}{1+f_i} & f_i \geq 0 \\ 1 + abs(f_i) & f_i < 0 \end{cases} \qquad (3)$$

where f_i is objective function and fit_i is the fitness function after transformation.

Onlookers are placed on the food sources by using a selection probability based on candidate solution. Thus, selection probability is determined by watching the waggle dance of employed bees. As the profitability of a food source increases, the probability value with which the food source is preferred by onlookers increases, too. The profitability can be expressed by fitness, while the selection probability is defined as follows:

$$P_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \qquad (4)$$

where fit_i is the fitness of the ith solution, SN is the number of solutions.

Selection strategy adopted by ABC is proportional selection. Selection probability is directly proportionate to the fitness of each solution. However, using this selection strategy has two problems: (i) A "super- individual" being too often selected the whole population tends to converge towards his position. The diversity of the population is then too reduced to allow the algorithm to progress. (ii) With the progression of the algorithm, the differences between fitness are reduced. The best ones then get quite the same selection probability as the others and the algorithm stops progressing. Therefore, this selection strategy is difficult to maintain diversity and avoid premature convergence. In order to palliate these problems, this paper has chosen three different selection strategies to improve ABC algorithm, simulation results show three modified ABC algorithm outperform ABC algorithm.

III. THREE SELECTION STRATEGIES

A. Rank Selection

The proportional selection in ABC is more like selection operator in genetic algorithm which reflects the idea of "survival of the fittest" in the process of nature evolution. This selection strategy makes more chance to select individuals with higher fitness. However, under certain circumstances, a worse solution also contains useful information. We proposed rank selection strategy in order that the worse ones have greater probability to be selected. In rank selection, each individual of the population is sorted according to the objective values. The fitness assigned to each individual depends only on its position in the individuals rank and not on the actual objective value. Selection probability is only related to ranking value of the individual. Rank selection provides a simple and effective way of controlling selective pressure, it has better robustness. A current worse solution may have a greater chance of "evolving" towards a better future solution by means of this method.

Rank selection calculates fitness value from formula (3) firstly, and then every individual is sorted according to their sizes from high to low, i.e. the best one has the index k=1 and the worst one k=n. Selection probability is defined as follows [11]:

$$\begin{aligned} P_k &= \frac{1}{n} + a(t)\frac{n+1-2k}{n(n+1)} \qquad k = (1,2,\cdots,n) \\ a(t) &= 0.2 + \frac{3t}{4N} \qquad t = (1,2,\cdots,N) \end{aligned} \qquad (5)$$

where a(t) is a self-adaptive parameters, N is the maximum iterations. At the early stage of evolution, a(t) should be lower value for keeping population diversity. At the later stage of evolution, a(t) should be higher value for preventing the trend of searching stagnation due to weakened competition in the population.

B. Disruptive Selection

The difference between disruptive selection [12] and the proportional selection is the definition of fitness function. The fitness function and selection probability are defined as follows:

$$fit_i = \left| f_i - \overline{f_t} \right| \qquad P_i = \frac{fit_i}{\sum_{i=1}^{n} fit_i} \qquad (6)$$

where f_i is the objective function, $\overline{f_t}$ is the average value of the objective function f_i of the individuals in the population.

Using this selection strategy, superior and inferior individual have more chances to be selected into the next generation, the individuals between high-fitness and low-fitness are eliminated. Disruptive Selection tends to preserve diversity somewhat longer because it favors both superior and inferior solutions.

C. Tournament Selection

The proportional selection in ABC algorithm requires the fitness function greater than zero. However, tournament selection [13] is different, it's a selection process based on local competition which only refers to the relative value of individuals. Its main idea is to randomly select individuals of k in the population and make comparison; the one with the best fitness will be selected. Parameter k is called tournament size, which is often held only between two individuals. In this paper, we select two individuals from the population and compare their fitness values, then assign one score to a better individual of the two, repeat such process and then the individual with the highest values wins the heaviest weight. This method of selection offers more chances for high-fitness individuals to survive. Meanwhile, this method avoids from the influence of the super individuals since it only standardizes relative value of fitness which is not in proportion to the size of fitness. To a certain extent, it also avoids from premature convergence and stagnation. The pseudocode of the tournament selection is listed as follows:

```
ai=0
for i=1:n
   for j=1:n
     if  fi ≤ fj
        ai = ai +1;
     end
   end
end
```

Selection probability of fi is $P_i = \frac{a_i}{\sum_{i=1}^{n} a_i}$ (7)

D. The Main Steps of Modified ABC Algorithm

The main steps of the modified ABC algorithm are listed as follows:

Setp1. Initializing each solutions of the population - x_{ij}, i=1,…,SN, j=1…D.

Setp2. Evaluating the population.

Setp3. Employed bees Produce new solution v_{ij} according to formula (1) in the neighbourhood of x_{ij}, and evaluate them.

Setp4. If fitness of v_i is better than x_i, v_i is replaced with x_i, otherwise x_i is maintained.

Setp5. Calculating the selection probability P_i for the solutions x_i according to corresponding selection strategy one, two or three (formula 5, 6 or 7, respectively).

Setp6. Onlookers select food sources depending on P_i , and produce new solution v_i according to formula (1), and evaluate them.

Setp7. The same as setp4.

Setp8. Determining the abandoned solution, if exists, replacing it with a new randomly produced solution xi for the scout according to formula (2).

Setp9. Memorizing the best solution achieved so far.

Setp10. If the stopping criterion is satisfied, output the best solution; otherwise, go to step 3.

IV. SIMULATION RESULTS AND DISCUSSION

A. Test Functions

In order to evaluate the performance of the modified algorithm, four famous test functions are selected in the simulation experiments: Sphere, Rosenbrock and two Penalized functions [14]. They are listed as follows:

1) Sphere function

$$f_1(x) = \sum_{i=1}^{n} x_i^2,$$
$$-100 \le x_i \le 100, \qquad \min(f_1) = f_1(0,...,0) = 0.$$

2) Rosenbrock function

$$f_2(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2],$$
$$-30 \le x_i \le 30, \qquad \min(f_2) = f_2(1,...,1) = 0.$$

3) Penalized function1

$$f_3(x) = \frac{\pi}{n}\{10\sin^2(\pi y_1) + +(y_n - 1)^2$$
$$+ \sum_{i=1}^{n-1}(y_i - 1)^2 \cdot [1 + 10\sin^2(\pi y_{i+1})]\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$$
$$-50 \le x_i \le 50, \quad \min(f_3) = f_3(1,...,1) = 0.$$

where

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x_i > a, \\ 0, -a \le x_i \le a, \\ k(-x_i - a)^m, x_i < -a. \end{cases}$$

$$y_i = 1 + \frac{1}{4}(x_i + 1)$$

4) Penalized function2

$$f_4(x) = 0.1\Big\{\sin^2(\pi 3x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\Big\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$$
$$-50 \le x_i \le 50, \quad \min(f_4) = f_4(1, \ldots, 1) = 0.$$

where

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x_i > a, \\ 0, -a \le x_i \le a, \\ k(-x_i - a)^m, x_i < -a. \end{cases}$$

Among the four functions above, n is the dimension of each function. Function f_1 is continuous, convex and unmoral. Function f_2 is a classical complicated optimization problem. The global optimum solution is inside a long, narrow, parabolic-shaped flat valley, so that there are very rare chances to converge to the global optima. Hence this function is often used in assessing the performance of the optimization algorithms. Function f_3 and f_4 are complex non-linear multimodal functions which have lots of local minima and can efficiently test the performance of global search.

B. Performance Analysis

The modified ABC algorithm based on three different selection strategies is compared with the basic ABC algorithm in order to identify above analysis. Three different modified ABC algorithms are called ABC algorithm based on rank selection (RABC), ABC algorithm based on disruptive selection (DABC) and ABC algorithm based on tournament selection (TABC), respectively. The number of colony size is set to 50, the numbers of onlooker bees and employed bees are 50% of the colony size. The dimensions are 30 and 50, for the maximum cycle number 1000 and 2000, respectively. In each experiment, the simulation runs 30 times randomly. The mean function values of the best solutions found by the algorithms for different dimensions have been recorded. The mean and the standard deviations of the function values obtained by ABC, RABC, DABC and TABC are given in Table 1−4. In order to show the performance of algorithm more clearly, Figs.1−4 show the progress of the mean best fitness when test function was taken as 30 dimensions, Figs. 5−8 show the progress of the mean best fitness value when test function was taken as 50 dimensions.

1) As shown in Figs. 1−8, all of three modified algorithms can find the global optima with nearly a line track for the unimodal function f_1. For function f_2, the performance of three modified algorithms is better than ABC. For function f_3 and f_4 with multimodal and many local minima, three modified algorithms are also better than ABC, and almost keep linear decreasing to find the global optima at the final stage. In each case, we find that the basic ABC algorithm performs well in the first period but fails to make further progress at the final stage. On the contrary, all of three modified algorithms are better than ABC at the final stage.

DABC algorithm maintains population diversity to a certain extent in the first period, but the performance of RABC and TABC are better than DABC at the final stage because the selection strategies had nothing to do with the size of fitness. Among the three modified algorithms, TABC outperforms the others at dimension 30, but RABC outperforms others at dimension 50. In a word, three modified ABC algorithms outperform the basic ABC algorithm.

2) Convergence rate and average convergence iterations are the very important indicator. The higher the convergence rate is, the better performance of the algorithm is, and otherwise it is difficult to find the global optima. Take dimension 50 as an example, the convergence rate (CR), average convergence iterations (ACI) and convergence precision (CP) of test function are given in Table 5. The results in Table 5 show that the convergence rates of the modified algorithms are all higher than ABC under the same iterations. The ABC algorithm can't achieve the convergence request with fixed precision, but three modified algorithms can meet the convergence request at different levels – where RABC is the highest convergence rate, average convergence iterations and convergence rate are better than the others. This indicates the validity of modified ABC algorithm from another angle.

TABLE I. SIMULATION RESULTS OF SPHERE FUNCTION

Alg.	Dim	Mean	Std
ABC	30	1.829458E-09	2.639270E-09
DABC	30	9.798135E-12	1.063763E-11
RABC	30	1.978102E-12	2.033197E-12
TABC	30	1.514169E-12	1.380414E-12
ABC	50	9.564480E-12	1.212776E-11
DABC	50	2.138416E-14	2.318155E-14
RABC	50	9.066074E-16	7.463074E-16
TABC	50	4.228150E-15	3.779910E-15

TABLE II. SIMULATION RESULTS OF ROSENBROCK FUNCTION

Alg.	Dim	Mean	Std
ABC	30	5.220163	5.278101
DABC	30	4.002284	2.780462
RABC	30	2.355272	1.769675
TABC	30	2.093640	1.698724
ABC	50	3.283130	3.986572
DABC	50	3.185141	2.264986
RABC	50	1.331720	1.301473
TABC	50	1.460538	1.191361

TABLE III. SIMULATION RESULTS OF PENALIZED FUNCTION 1

Alg.	Dim	Mean	Std
ABC	30	6.759335E-11	1.203652E-10
DABC	30	4.646641E-12	9.103033E-12
RABC	30	2.906279E-13	5.511228E-13
TABC	30	1.825101E-13	1.834557E-13
ABC	50	2.489125E-13	3.612222E-13
DABC	50	2.693182E-14	1.390646E-13
RABC	50	1.484326E-16	3.746631E-16
TABC	50	1.344679E-15	5.968968E-15

TABLE IV. SIMULATION RESULTS OF PENALIZED FUNCTION 2

Alg.	Dim	Mean	Std
ABC	30	4.318483E-09	5.857697E-09
DABC	30	8.047858E-11	1.535506E-10
RABC	30	1.894350E-11	6.586020E-11
TABC	30	1.766890E-12	5.587658E-12
ABC	50	5.238564E-10	1.882791E-09
DABC	50	1.376825E-12	5.685854E-12
RABC	50	3.758103E-15	5.988249E-15
TABC	50	6.883571E-15	1.085744E-14

TABLE V. SIMULATION RESULTS OF TEST FUNCTION UNDER THE SAME PRECISION (50 DIMENSIONS)

Fun	Alg.	ACI	CR (%)	CP
$f_1(x)$	ABC	/	0	10^{-14}
	DABC	1969.60	33.33	
	RABC	1888.63	100.00	
	TABC	1941.69	86.67	
$f_2(x)$	ABC	1496.79	63.33	0.5
	DABC	1604.48	76.67	
	RABC	1683.46	80.00	
	TABC	1677.91	73.33	
$f_3(x)$	ABC	/	0	10^{-15}
	DABC	1940.13	53.33	
	RABC	1862.56	96.67	
	TABC	1897.07	93.33	
$f_4(x)$	ABC	/	0	10^{-14}
	DABC	1962.42	40.00	
	RABC	1920.48	90.00	
	TABC	1933.58	86.67	

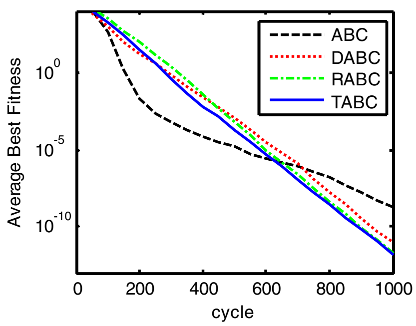

Figure 1. $f_1(x)$ Sphere Function

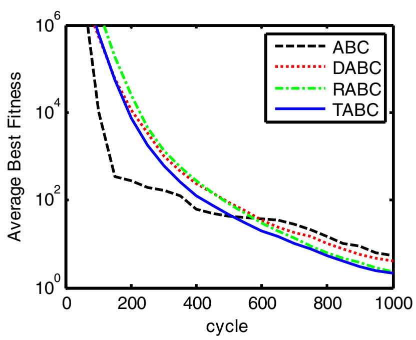

Figure 2. $f_2(x)$ Rosenbrock Function

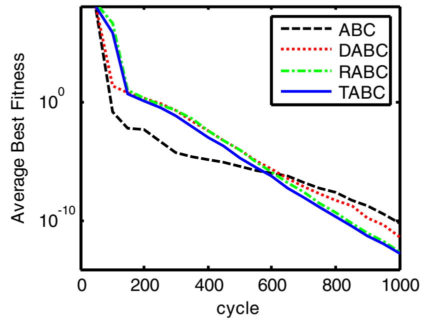

Figure 3. $f_3(x)$ Penalized Function 1

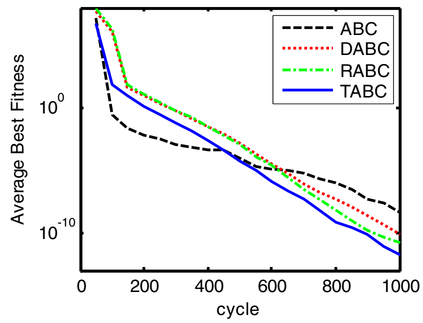

Figure 4. $f_4(x)$ Penalized Function 2

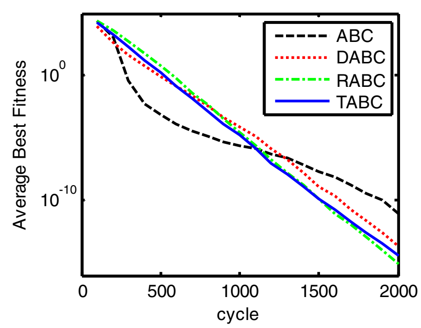

Figure 5. $f_1(x)$ Sphere Function

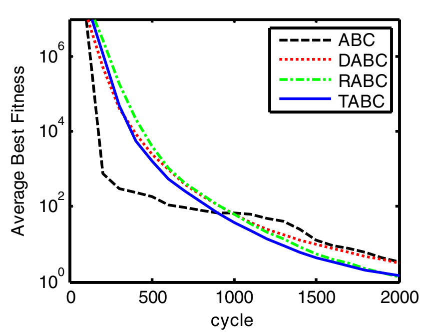

Figure 6. $f_2(x)$ Rosenbrock Function

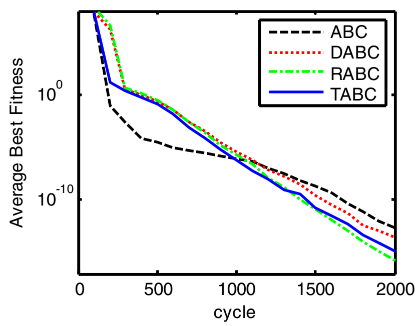

Figure 7. f_3(x) Penalized Function 1

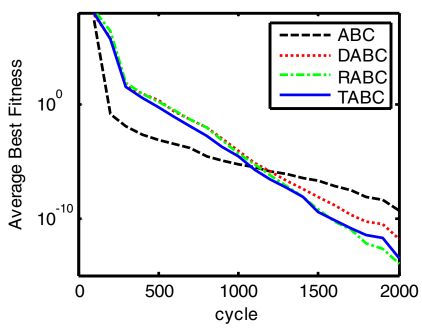

Figure 8. f_4(x) Penalized Function 2

V. CONCLUSION

In this paper, we compare and analyze the selection mechanism in ABC algorithm, and propose the modified ABC algorithm based on three different selection strategies. The performance of the modified ABC algorithm and the basic ABC was tested for four high dimensional numerical optimization functions. The simulation results show that modified ABC algorithm by three different selection strategies increase population diversity and avoid premature convergence. In general, the modified ABC algorithms based on three different selection strategies perform better than the basic ABC algorithm.

REFERENCES

[1] D. Karaboga, "An Idea Based on Honey Bee Swarm for Numerical Optimization", Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[2] D. Karaboga, B. Basturk, "A Powerful And Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm", Journal of Global Optimization, Springer Netherlands, vol.39, no.3, 2007, pp.459-471.

[3] B. Basturk, D. Karaboga, "An Artificial Bee Colony (ABC) Algorithm for Numeric function Optimization", IEEE Swarm Intelligence Symposium 2006, Indianapolis, Indiana, USA. May 12-14, 2006.

[4] D. Karaboga, B. Basturk, "On The Performance of Artificial Bee Colony (ABC) Algorithm", Applied Soft Computing, January 2008, vol.8, no.1, pp.687-697.

[5] D. Karaboga, B. Basturk, "Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems", LNCS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing, Springer- Verlag, 2007, IFSA 2007. vol. 4529/2007, pp.789-798.

[6] D. Karaboga, B. Basturk Akay, "Artificial Bee Colony Algorithm on Training Artificial Neural Networks", Signal Processing and Communications Applications, SIU 2007, IEEE 15th, June 11-13, 2007, pp.1 - 4.

[7] D. Karaboga, B Basturk Akay, C Ozturk, "Artificial Bee Colony (ABC) Optimization Algorithm for Training Feed-Forward Neural Networks", LNCS: Modeling Decisions for Artificial Intelligence, MDAI 2007, Springer-Verlag, 2007, vol.4617, pp.318-319.

[8] D. Karaboga, C. Ozturk, B. Akay, "Training Neural Networks with ABC Optimization Algorithm on Medical Pattern Classification", International Conference on Multivariate Statistical Modelling & High Dimensional Data Mining, Erciyes University, Kayseri, TURKEY, June 19-23, 2008.

[9] A. Singh, "An Artificial Bee Colony Algorithm for the Leaf-constrained Minimum Spanning Tree Problem", Applied Soft Computing, 2009, vol.9, no.2, pp.625-631.

[10] H. J. DING, F. L. LI, "Bee Colony Algorithm for TSP Problem and Parameter Improvement", China Science and Technology Information, 2008, no.3, pp.241-243.

[11] A.G. SONG, J.R. LUO, "A Ranking Based Adaptive Evolutionary Operator Genetic Algorithm", Acta Electronica Sinica, 1999, vol.27 no.1, pp.85-88.

[12] T. Kuo, S. Y. Huang, "Using Disruptive Selection to Maintain Diversity in Genetic Algorithms", Applied Intelligence, 1997, vol.7, no.3, pp. 257~267.

[13] T. Blickle, and L. Thiele, "A Mathematical Analysis of Tournament Selection", Proceedings of the Sixth International Conference on Genetic Algorithms, San Francisco, California, 1995, pp.9-16.

[14] X. Yao, Y. Liu, G.M. Lin, "Evolutionary Programming Made Faster", IEEEE Transactions on Evolutionary Computation, 1999, pp.82-102.