Artificial Tribe Algorithm and Its Performance Analysis

Tanggong Chen, Youhua Wang and Jianwei Li
Province-Ministry Joint Key Laboratory of Electromagnetic Field and Electrical Apparatus Reliability, Hebei University of Technology, Tianjin, 300130, China
Email: tgchen@hebut.edu.cn

***Abstract*—In this paper, a novel intelligent optimization algorithm - Artificial Tribe Algorithm (ATA) is presented based on the analyses of the principle and uniform framework of the Bionic Intelligent Optimization Algorithms (BIOA). ATA simulates the existent skills of the natural tribes, and actualizes the optimization purpose through the propagation and migration behaviors of the tribes. The main factors which influence the performance of ATA have been discussed. ATA is used for unconstrained and constrained functions optimization and the results produced by ATA, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Artificial Fish-Swarm Algorithm (AFSA) have been compared. The results show that ATA is a powerful algorithm for global optimization problems.**

***Index Terms*—artificial tribe algorithm, bionic intelligent optimization algorithm, optimization, swarm intelligence, genetic algorithm, particle swarm algorithm, artificial fish-swarm algorithm**

I. INTRODUCTION

Most animals and insects show the amazing abilities of completing complex behaviors. Since 1940s, the optimization design problems in the engineering fields have been solved by using the inspiration of the biological systems, meantime the Bionic Intelligent Optimization Algorithm (BIOA) was designed. At present, the popular BIOAs are Genetic Algorithm (GA) [1], Ant Colony Algorithm (ACA) [2], Particle Swarm Optimization (PSO) [3], Artificial Fish-Swarm Algorithm (AFSA) [4], Shuffled Frog Leaping Algorithm (SFLA) [5], and Artificial Searching Swarm Algorithm (ASSA) [6]. These bionic algorithms have become a research focus in the fields of intelligent optimization.

In this work, a novel bionic intelligent optimization algorithm –Artificial Tribe Algorithm (ATA) is presented and discussed. ATA simulates the existent skill of the natural tribe, through the propagation and migration behaviors to renew the tribe and move the living region of the tribe if the existent situation is bad. The algorithm is used for optimizing multivariable functions (unconstrained or constrained optimization problems) and the results produced by ATA, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Artificial Fish-Swarm Algorithm (AFSA) have been compared. The results show that ATA is a powerful global optimization algorithm.

Manuscript received January 1, 2009; revised June 1, 2009; accepted July 1, 2009.
project number: 50877022, E2011202040

II. BIONIC INTELLIGENT OPTIMIZATION ALGORITHM

BIOA is a new kind of optimization method. Although the characters are different, BIOA manifests the great similarity in the structure, research content, research method and the operation model. It provides the possibility for the establishment of uniform framework.

The uniform framework of BIOA can be described as follows [7,8]: the swarm is formed by the individuals, relies on the specific evolution rules, goes with the iteration of the algorithm, and produces the reborn swarm (such as GA) or changes the individual position (such as PSO, AFSA, SFLA, ASSA). The optimal solution evolves unceasingly along with the change of the swarm and then appears finally. The process of the algorithms is shown as follows:

1. Set the parameters, initialize the swarm randomly and calculate the fitness values;
2. According to the set rules, renew swarm or change its position, generate one group of solutions, and calculate the fitness of individuals;
3. Through comparison, obtain the optimal fitness value and take note;
4. Judge whether or not the terminal condition is satisfied. If satisfy, then end the iteration; otherwise, return to 2.

In this uniform framework, the rules of renewing swarm and changing individual's position play an important role and decide the algorithm performance. These set rules represent the direct biology foundation of the algorithm, restrict the individual behaviors, and form the unique characteristics of the algorithm to be different from the other algorithms.

In BIOA the bulletin board is generally set up, which records the optimal individual's historical states. Through the iteration of the algorithm, each individual compares its own condition with bulletin board's condition, and replaces the information when its own value is better. Thus the bulletin board can record the historical best solution all along. After algorithm iteration finished, the optimal solution and the related information can be obtained from the bulletin board.

III. ARTIFICIAL TRIBE ALGORITHM

doi:10.4304/jsw.7.3.651-656

A. Basic Idea of ATA

According to the uniform framework of BIOA, ATA uses the tribe which composed of individuals as the main executive, and simulates the existent skills of the tribes to actualize the optimization purpose. With the uninterrupted iteration of the algorithm, the optimal solution shows finally, so the corresponding optimization problem is solved.

As the above analyses BIOA can be differentiated into two kinds: one is how to produce the next generation (such as GA); the other is how to change the swarm position (such as PSO, AFSA, ASSA). In this work, ATA tries to integrate the characters of the two kind algorithms. If the existent situation is adaptive, the tribe propagates to reborn the next generation through propagation behavior, otherwise, if the existent situation is bad, the tribe migrates to the new region through migration behavior, and propagates right along.

According to the above ideas, how to estimate the existent situation is the key problem. ATA sets existent criterion to indicate the situation of the tribe through comparing the optimal fitness value with that of anterior generation. If the intergenerational difference of the optimal fitness value is smaller than the existent criterion, it indicates that the existent situation is bad, and the tribe must migrate immediately.

The rules of the individual behavior can be described as follows:

1. Propagation: an individual uses coordinates sequence to form the gene code. If the existent situation is adaptive, the individual selects the other individual of the tribe randomly, and propagates to reborn the next generation through exchanging partial gene code.

2. Migration: If the existent situation is bad (that indicates the intergenerational difference of the optimal fitness value is smaller than the existent criterion), the individual moves the position according to it's and tribe's historical experience to implement the migration of the tribe.

Two rules of the ATA have their own characteristics. Propagation rule is similar with the crossover operator of the GA, and can find the better goals; Migration rule plays a main role in finding the better goals, pulls the algorithm away from the local solution, and has the global adjusting function. The above analyses have been confirmed by simulation tests.

The running flow of ATA is shown as follows:

1. Set the parameters, initialize the tribe randomly and calculate the fitness value;

2. Iteration counter add 1, estimate the existent situation of the tribe. If the existent situation is adaptive (that indicates the intergenerational difference of the optimal fitness value is bigger than the existent criterion), implement propagation behavior to the individuals in turn; Otherwise, implement the migration behavior (the migration behavior is prohibited to use continuously).

3. Calculate the fitness value. Compare with the best fitness of tribe and each individual respectively, if better, log on the bulletin board;

4. Determine whether or not to satisfy the conditions of termination, if satisfy, then end the iteration; otherwise, return to 2.

Because of including global information, the tribe's historical experience compared with that of the individual has more weight to find the better goals and enhance the searching ability. So the better effect can be expected if increase the weight of tribe's experience properly. So ATA introduces the global inertia weight to enhance the searching ability and accelerate the convergent speed.

B. Pseudo-Code Description of the ATA

For explaining the behavior rule clearly, by using object-oriented technology, an individual can be described as a C++ class as follows:

```
class Artificial_Tribe_Individuals
{
  float X[n];  //individual's position.
  float AT_fitness();  //the object function
  float AT_criterion; //the estimation criterion of the
                                  existent situation
  float AT_w; //the global inertia weight
  void AT_propagation();  //the behavior of propagation
  void AT_migration();  //the behavior of migration
  Artificial_Tribe_Individuals();
  virtual~Artificial_Tribe_Individuals();
};
```

Suppose r_1 , r_2 are the two random real number, $0 \le r_1, r_2 \le 1$, the individual's position is X_i in i-th iteration, the historical best solution of the individual is X_s, the historical best solution of the tribe is X_g, the position of the individual which be selected to propagate is Y_i, then the two behaviors can be described as follows:

void AT_propagation()

{

$$\begin{aligned} X_{i+1} &= r_1 * X_i + (1 - r_1) * Y_i \\ Y_{i+1} &= r_1 * Y_i + (1 - r_1) * X_i \end{aligned} \tag{1}$$

};

void AT_migration()

{

$$X_{i+1} = X_i + r_1 * |X_s - X_i| + AT\_w * r_2 * |X_g - X_i| \tag{2}$$

}.

IV. Simulation and Performance Analysis

A. Simulation Functions

To verify the effectiveness of ATA, some typical functions are chosen for simulation tests as follows:

$$F_1(x,y) = \frac{sin(x)}{x} \times \frac{sin(y)}{y} \quad x,y \in [-10,10] \tag{3}$$

$$F_2(x_1,x_2,\cdots,x_n) = \sum_{i=1}^{n-1} 100(x_i^2 - x_{i+1})^2 + (1-x_i)^2$$
$$x_i \in [-2.048, 2.048], \ i = 1,2,\cdots,n \tag{4}$$

$$F_3(x_1,x_2,x_3) = 30 + \sum_{i=1}^{3} \left(\frac{1}{10}(x_i - 20)^2 - 9cos\frac{2\pi}{5}x_i\right)$$
$$x_i \in [1,39] \quad i = 1,2,3 \tag{5}$$

$$F_4(x_1,x_2) = 0.5 + \frac{\sin^2\sqrt{x_1^2+x_2^2} - 0.5}{[1.0+0.001(x_1^2+x_2^2)]^2}$$
$$x_i \in [-100,100] \quad i=1,2 \tag{6}$$

$$F_5(x_1,x_2) = [1+(x_1+x_2+)^2(19-14x_1+3x_1^2-14x_2 + 6x_1x_2+3x_2^2)][30+(2x_1-3x_2)^2(18- 32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$$
$$x_i \in [-2,2] \quad i=1,2 \tag{7}$$

$$F_6(x_1,x_2) = (x_1-10)^3+(x_2-20)^3$$
$$s.t.\begin{cases} g_1(x_1,x_2) = 100-(x_1-5)^2-(x_2-5)^2 \le 0 \\ g_2(x_1,x_2) = -82.81+(x_1-6)^2-(x_2-5)^2 \le 0 \\ 13 \le x_1 \le 100, 0 \le x_2 \le 100 \end{cases} \tag{8}$$

$$F_7(x_1,x_2) = (x_1^{\ 2}+x_2-11)^2+(x_1+x_2^2-7)^2$$
$$s.t.\begin{cases} g_1(x_1,x_2) = (x_1-0.05)^2+(x_2-2.5)^2-4.84 \le 0 \\ g_2(x_1,x_2) = 4.84-x_1^2-(x_2-2.5)^2 \le 0 \\ 0 \le x_1,x_2 \le 6 \end{cases} \tag{9}$$

F_1 has a global maximum 1 at point [0, 0], and a lot of local maximums distributed around it. The usual algorithms are easy to fall into local maximums or vibrate between the local minimum and maximum. It is often chosen to verify the validity of the algorithm.

F_2 is the general Rosenbrock function. It has the global minimum value 0 at the points [1, 1,…,1]. As the global optimum is inside a long, narrow, parabolic shaped flat valley, it is difficult for global optimization, and often be used to detect the searching ability of algorithm.

F_3 has a global minimum value 3 at point [20,20,20], and a lot of local minimum value distributed around it, it is a multi-variable function.

F_4 is the Schaffer function and has a global minimum 0 at point [0, 0].

F_5 is Goldstein-Price function and has global minimum 0 at point [0, 0].

F_6 is a constrained optimization problem; it subjects to (s.t.) some constrained conditions. It has the minimum value -6961.81388 at point(14.095, 0.84296).

F_7 is a constrained optimization problem. It has the minimum value 13.59084 at point (2.2468, 2.3818).

B. Performance Analysis

ATA is used to solve function F_1. The size of the tribe is 10, the AT_criterion is 0.3, the AT_w is 1.46, and the iteration is 50. Fig. 1 shows the initial distribution of the tribe, the individuals distribute randomly in the defining region.

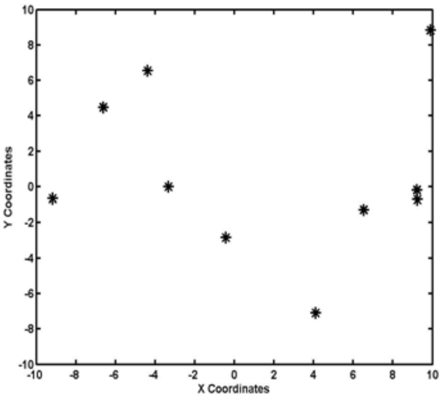

Figure1. Distribution of the initial tribe

After 5 iterations, the distribution of the tribe is shown in Fig. 2. Some individuals have moved near the global optimum position. The tribe has centralized near the point [0, 0] after 5 iterations only, and it is showed that ATA has quick convergent speed.

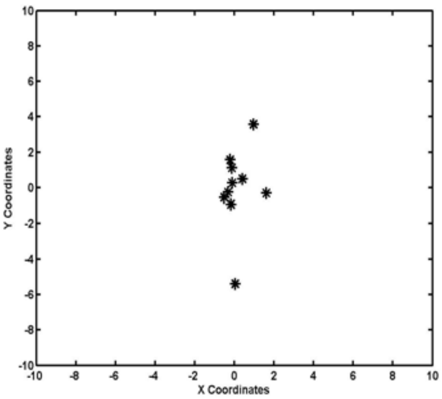

Figure2. Distribution of the tribe after 5 iterations

After 12 iterations, the tribe distribution is shown in Figure 3. For display clearly, the distribution region of the tribe reduces a half compared with figure 2. More individuals have centralized near the global optimal solution, and the best individual is very approach to the global optimal solution.

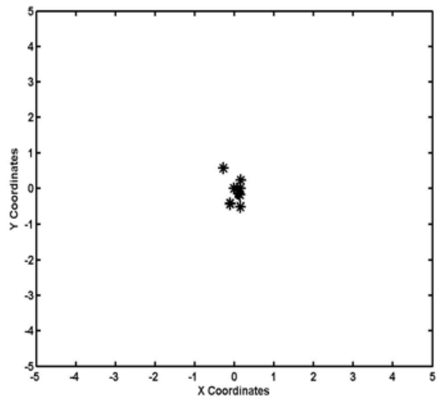

Figure3. Distribution of the tribe after 12 iterations

As Fig. 4 shown, after 19 iterations, almost all the individuals have centralized near the optimal solution, and the value of the best individual is equal to the optimal value, that concretely shows the effectiveness of the algorithm.

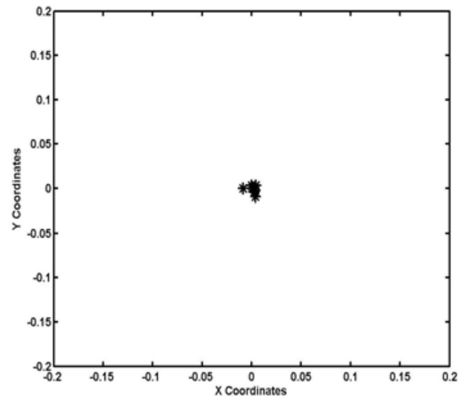

Figure4. Distribution of the tribe after 19 iterations

In order to describe the process of the tribe hunting for the optimal solution concretely, Fig. 5 shows the mutative track of the best individual's fitness value. With the algorithm iteration the individuals change quickly through propagation and migration behaviors, and the value of the best individual gets to the optimal value rapidly.

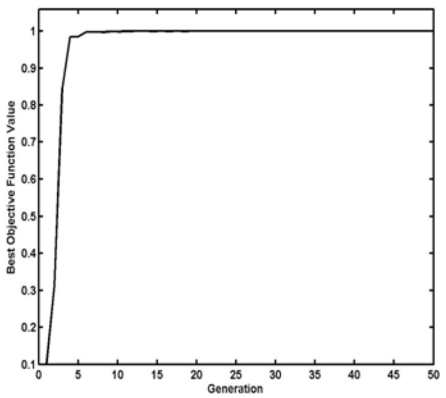

Figure5. Objective function Value of the best individual

As well known tribe size is one of the main factors which affect the performance of the ATA. Select F_3 to do simulation testing for discussing the influence. The ATA_criterion is 0.3, the ATA_w is 1.65, and the iteration is 50. The results are shown in Fig. 6.

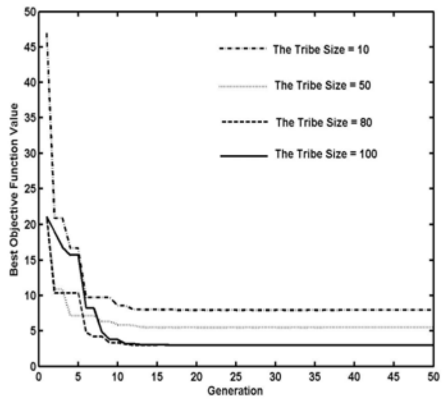

Figure6. Influence of tribe size

Although the ATA has the running ability of small tribe size and can obtain better searching results, but the tribe size continues to be one of the key factors that influence the performance of ATA. In general, the larger the tribe size, the better the performance can be got, but the efficiency of algorithm reduces; The smaller the tribe size, the faster the ATA searching speed is, but the diversity of population reduces. It does not meet the linear relationship between the tribe size and the performance of algorithm, so the tribe size should be moderate.

The inertia weight is the other factor that influences the performance of ATA. F4 is selected to do simulation test, the tribe size is 10, the AT_criterion is 0.3, the iteration is 100 and the AT_w are 2.6, 3.6, 4.6 and 5.6 respectively. The results are shown as Fig. 7.

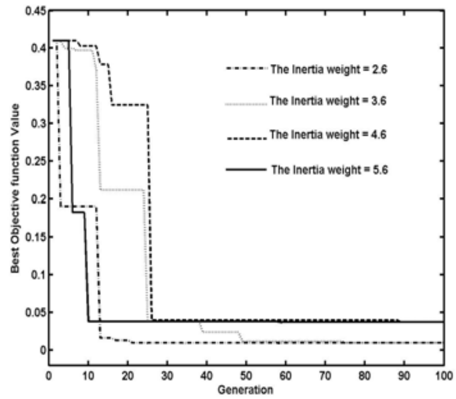

Figure7. Influence of inertia weight

From the results selecting the inertia weight properly can enhance the performance of ATA efficiently. Because ATA is a stochastic algorithm, there is no determinate result about the influence. The value of inertia weight should be moderate too.

C. Performance Comparison

ATA and GA are used respectively to solve function F_2.

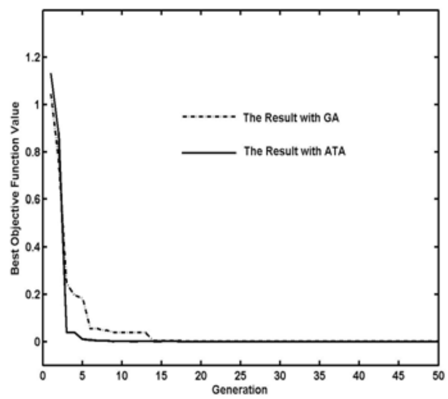

Figure8. Comparison of the best individual value between ATA and GA

Select n is 3, the tribe size is 10, the AT_criterion is 0.3, the AT_w is 1.2, and the iteration is 50. The GA uses binary code, the crossover probability is 0.6, the mutation probability is 0.001, and the length of code is 30. The test results are shown in Fig. 8. In the early running time, the

ATA is slower than the GA, but ATA exceeds the GA, and gets the better optimization results finally. GA gets the optimal value 0.001730 at 18-th generation, while ATA gets the same precision value at 9-th generation, and gets the optimal value 0.000053 at 47-th generation. The results show that ATA outperforms GA in global optimization.

For GA the premature convergence is easy to occur while the swarm size is small. Fig. 8 shows the better running example of the tests. ATA does not have the same phenomenon, and still has the good searching capability when the swarm size is small. So ATA has the running ability of small swarm size which can save a lot of computing resource.

Function F_1 is selected to do simulative experiment with ATA and AFSA respectively. The tribe size is 10, the iteration is 100, the ATA_criterion is 0.3, and the ATA_w is 1.46; the step of AFSA is 0.3, the visual of AFSA is 15, and the consistence of AFSA is 0.618. Experiments are 100 times, and the results are shown as Fig. 9. From the results, ATA has better performance than AFSA besides several running cases.

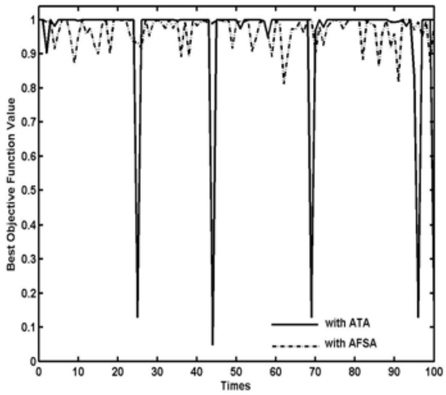

Figure9. Comparison of the best individual value of ATA and AFSA

In order to compare the performance of ATA and PSO, Function F_5 is selected to do simulative experiment.

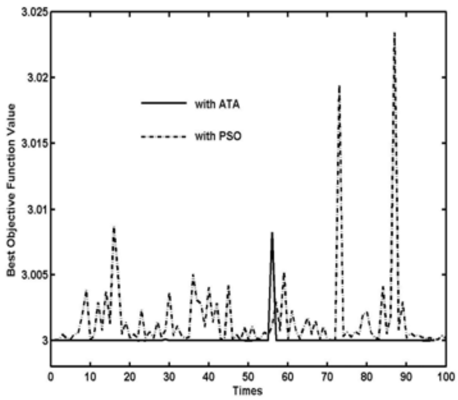

Figure10. Comparison of the best individual value of ATA and PSO

The tribe size is 50, the iteration is 100, ATA_criterion is 0.3, and the ATA_w is 1.22; the inertia weight of PSO is descending, the experiments are 100 times, and the results are shown as Fig. 10. At 14-th and 71-th experiment ATA obtains the value 30, the operations are failing; at 63-th experiment ATA obtains the value 3.585441, and at other one experiments ATA obtains the optimal value 3 (for showing clearly Figure 10 deletes above three cases). Generally ATA has better performance than PSO besides several running cases.

From the simulative results, for the multivariable unconstrained function optimization, ATA is an efficient global optimization algorithm.

D. Penalty Function Method

In scientific research and social practice fields many optimization problems have constrained conditions. For solving the constraint optimization problem, penalty function method is a useful method. Penalty function method converts the object function into the penalty function which included the constrained information, and the process of solving constraint optimization problem is transformed to the process of continuously solving unrestraint optimization problem. The basic idea of the penalty function method is shown as follows.

Generally the constrained optimization problem can be described as fellows:

$$\min f(X)$$
$$s.t. \begin{cases} g_m(X) \le 0, m = 1,\cdots,p \\ h_j(X) = 0, j = 1,\cdots,q \\ x_i^l \le x_i \le x_i^u, i = 1,\cdots,n \end{cases} \quad (10)$$

Here $x_i \in [x_i^l, x_i^u], i = 1,2,\cdots,n$ are the design variables, $X = (x_1, x_2, \cdots, x_n) \in R^n$, $f(X)$ is objective function, $g_m(x)$ are m-th non-equation constrained conditions, $h_j(x)$ are j-th equation constrained conditions, and $S = \prod_{i=1}^{n} [x_i^l, x_i^u]$ is the search space.

Because the equation constrained condition can be transformed to non-equation constrained condition, then the $h_j(x)$ can be take off from the model for simple.

Define the penalty function as

$$F(X, r_k) = f(X) + r_k \sum_{m=1}^{p} \{\max[g_m(X), 0]\}^2 \quad (11)$$

$$\max[g_m(X), 0] = \frac{g_m(X) + |g_m(X)|}{2} \quad (12)$$

Here r_k is the positive, penalty factor sequence, and increases by degrees. It is proved that if $r_k \to \infty$, the optimal solution of $F(X, r_k)$ will convergence to optimal solution of the objective function.

Penalty function method has the penalty factor sequence which makes difficulty to program. The penalty factor (has fixed value) usually be used for simple, and the effects are almost equivalent in practice.

About function F_6, with penalty function method, select population size is 500, iteration times is 500, AT_criterion=0.6, AT_w=8.6, testing times=20, and the penalty factor=10^8. The results are shown as table I.

TABLE I. THE RESULTS OF F_6

Method	The Optimal Value		
	Best	*Median*	*Worst*
Reference [10]	-6961.814	-6955.812	-6954.446
Reference [11]	-6961.801	-6954.734	-6951.622
This Paper	-6960.269	-6885.712	-6797.484

About function F_7, with penalty function method, select population size is 500, iteration times is 100, AT_criterion=0.3, AT_w=1.6, testing times=50, and the penalty factor=10^8. The results are shown as table II.

TABLE II. THE RESULTS OF F_7

Method	The Optimal Value		
	Best	*Median*	*Worst*
Reference [9]	13.59084	13.59087	13.59097
Reference [10]	13.59084	13.59084	13.59085
This Paper	13.59084	13.59083	13.59082

From simulative results ATA is an effective algorithm for solving constrained optimization problems.

V. CONCLUSIONS

Artificial Tribe Algorithm (ATA) is based on the simulation of the operation principle and unity framework of BIOA, so it is an abstract algorithm. ATA simulates the existent skills of the natural tribes, and actualizes the optimization purpose through the propagation and migration behaviors of the tribes. ATA can be used for solving both unconstrained and constrained optimization problems effectively.

ATA possesses little parameters, simple structure, insensitive to initial values, and small swarm running ability. The algorithm was used for optimizing multivariable functions and the results produced by ATA, Genetic Algorithm (GA), particle swarm optimization (PSO), and artificial fish-swarm algorithm (AFSA) have been compared. The experiment results showed that ATA is a powerful algorithm in global optimization.

ACKNOWLEDGMENT

This work was supported by Project 50877022 of the National Science Foundation of China and Project E2011202040 of the Natural Science Foundation of Hebei Province.

REFERENCES

[1] Holland J. H. Adaptation in Nature and Artificial System. MIT Press, 1992.

[2] A. Colorni, M. Dorigo, V. Maniezzo, et al. Distributed optimization by ant colonies. Proceedings of the 1st European Conference on Artificial Life, 1991, pp. 134-142.

[3] J. Kennedy, R. C. Eberhart. Particle Swarm Optimization. Proceedings of the IEEE International Conference on Neural Networks, 1995, IV, pp. 1942-1948.

[4] X. L. Li, Z. J. Shao, J. X. Qian. An Optimization Method Based on Autonomous Animats: Fish-swarm Algorithm. Systems Engineering-Theory & Practice, 2002, 22(11), pp. 32-38.

[5] M. Eusuffm, K. E. Lansey. Optimization of Water Distribution Network Design Using Shuffled Frog Leaping Algorithm. Journal of Water Resources Planning and Management, 2003, 129(3), pp. 210-225.

[6] T. G. Chen. A Simulative Bionic Intelligent Optimization Algorithm: Artificial Searing Swarm Algorithm and its performance Analysis. Proceedings of the Second International Joint Conference on Computational Sciences and Optimization, 2009, Volume 2: pp.864-866.

[7] H. Wang, F. Qian. Swarm intelligent optimization Algorithms. Chemical Automation and Instrumentation, 2007, 34 (5), pp.7-13.

[8] H. B. Duan, D. B. Wang, X. F. Yu. Research on some Novel Bionic Optimization Algorithms. Computer Simulation, 2007, 24 (3), pp.169-172.

[9] X. H. Wang, X. M. Zheng and J. M. Xiao, "Artificial fish-swarm algorithm for solving constrained optimization problems," Computer Engineering and Applications, vol. 43, No. 3, pp. 40–42,63, 2007.

[10] Y. R. Zhou, J. X. Zhou and Y. Wang, "An optimised evolutionary algorithms for nonparameter penalty function," Computer Engineering, vol. 31, No. 10, pp. 31–33, 2005.

[11] T. G. Chen, L. L. Pang, J. Du, Z. B. Liu and L. J. Zhang, "Artificial Searching Swarm Algorithm for Solving Constrained Optimization Problems," Proceedings of IEEE International Conference On Intelligent Computing and Intelligent Systems, 2009, Volume 1, pp.562-565.

TangGong Chen Shandong Province, China. Birth date: March, 1963. is Electrical Engineering Ph.D., graduated from school of Electrical Engineering Hebei University of Technology. And research interests on intelligent Optimization algorithm, electrical CAD technique, electromagnetic fields computing, machine learning.

He is a associate professor of school of Electrical Engineering Hebei University of Technology.

YouHua Wang Jiangxi Province, China. Birth date: November, 1964. is Electrical Engineering Ph.D., graduated from school of Electrical Engineering Fuzhou University. And research interests on global Optimization, electrical power and electrics, electromagnetic metallurgy.

He is a professor of school of Electrical Engineering Hebei University of Technology.

JianWei Li Hebei Province, China. Birth date: November, 1974. is Electrical Engineering Ph.D., graduated from school of Electrical Engineering Hebei University of Technology. And research interests on intelligent computing, support vector machine.

He is a associate professor of school of Computer and Software Hebei University of Technology.