The Sixteenth International Symposium on Artificial Life and Robotics 2011 (AROB 16th '11),
B-Con Plaza, Beppu,Oita, Japan, January 27-29, 2011

Improved Artificial Bee Colony Algorithm for Large-Scale Optimization Problems

Ryuta Gocho†, Akihide Utani‡, and Hisao Yamamoto‡

†*Graduate School of Engineering, Tokyo City University*

‡*Faculty of Knowledge Engineering, Tokyo City University*

1-28-1, Tamazutsumi, Setagaya, Tokyo 158-8557, Japan

{autani; yamahisa}@tcu.ac.jp

***Abstract*:** This paper proposes a new optimization algorithm based on Artificial Bee Colony（ABC）algorithm that has the good performance on large-scale optimization problems. We evaluate the proposed algorithm through numerical experiments on well-known benchmark functions, such as *Rosenbrock* function, *Rastrigin* function, *Schwefel* function, *Ackley* function and *Griewank* function, and discuss its development potential. In numerical experiments, the performance of the proposed algorithm is compared with those of the existing optimization algorithms.

***Keywords*: Wireless Sensor Networks, Particle Swarm Optimization, Query Dissemination, Long-Term Operation.**

I. INTRODUCTION

The minimization of multimodal functions with many local and global minima is a problem that frequently arises in diverse scientific fields and numerous engineering design problems. This problem is NP-hard in the sense of its computational complexity even in simple cases. As techniques of computing a global minimum of the objective function, many meta-heuristics, which are search algorithms for optimization based on heuristic knowledge, have been proposed. Some well-known representative meta-heuristics are Simulated Annealing and Tabu Search, which are the traditional optimization algorithms, Genetic Algorithms and Immune Algorithms, which are classified as evolutionary computation techniques, and Ant Colony Optimization algorithms and Particle Swarm Optimization algorithms, which belong to the category of swarm intelligence algorithms.

In meta-heuristics, Genetic Algorithms and Immune Algorithms, classified as evolutionary computation techniques, are generally techniques for combination optimization problems. In Genetic Algorithms and Immune Algorithms, the variables of continuous type are frequently translated into those of discrete（genetic）type. If there is a dependency between variables, therefore, a promising solution may be destroyed during the solution search process（genetic operation）and the solution search performance may deteriorate. To the contrary, Particle Swarm Optimization algorithm can directly handle the variables of continuous type. Even when there is a dependency between variables, therefore, an efficient and effective solution search can be realized. Recently, Particle Swarm Optimization algorithm is intensively researched because it is superior to the other algorithms on many difficult optimization problems. The ideas that underlie Particle Swarm Optimization algorithm are inspired not by the evolutionary mechanisms encountered in natural selection, but rather by the social behavior of flocking organisms, such as swarms of birds and fish schools. Particle Swarm Optimization algorithm is a population-based algorithm that exploits a population of individuals to probe promising regions of the search space. The algorithm is simple and allows unconditional application to various optimization problems. However, it has been confirmed that the performance of Particle Swarm Optimization algorithm on large-scale optimization problems is not always satisfactory.

This paper proposes a new optimization algorithm based on Artificial Bee Colony（ABC）algorithm that has the good performance on large-scale optimization problems. We evaluate the proposed algorithm through numerical experiments on well-known benchmark functions, such as *Rosenbrock* function, *Rastrigin* function, *Schwefel* function, *Ackley* function, and *Griewank* function, and discuss its development potential. In numerical experiments, the performance of the proposed algorithm is compared with those of the existing optimization algorithms. The rest of this paper is organized as follows. In Section II, the proposed algorithm is described. In Section III, experimental results are reported. Finally, the paper closes with conclusions and ideas for further study in Section IV.

II. PROPOSED ALGORITHM

In ABC algorithm, the colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts. In the initial state of "artificial bee colony", the colony consists of the employed bees and the onlookers.

At the initial stage of ABC algorithm, multiple solution points are randomly set in multidimensional solution search space. For every solution point, there is only one employed bee. In other words, the number of employed bees is equal to the number of solution points. The employed bee of an abandoned solution point becomes a scout. The search carried out by the artificial bees can be summarized as follows:

1. Each employed bee randomly searches a more suitable solution point within the neighborhood of the solution point in its memory.
2. Employed bees share their search information with onlookers and then onlookers select one of solution points by the following equations:

$$fit_i^k = \begin{cases} \dfrac{1}{1+f(\boldsymbol{x}_i^k)}, & f(\boldsymbol{x}_i^k) \geq 0 \\ 1+abs(f(\boldsymbol{x}_i^k)), & f(\boldsymbol{x}_i^k) < 0 \quad (i=1,\ldots,SN) \end{cases} \tag{1}$$

$$P_i^k = fit_i^k \Big/ \sum_{n=1}^{SN} fit_n^k \tag{2}$$

where $f(\boldsymbol{x})$ is the objective function of variables $(\boldsymbol{x})$. The subscript $i\,(i=1, \ldots, SN)$ and the superscript k indicates the solution point's index and the number of search iterations, respectively. SN is the number of solution points. Onlookers select one of solution points by referring to the probability (P_i^k) of each solution point based on the search information from employed bees.

3. Each onlooker randomly searches a more suitable solution point within the neighborhood of the solution point chosen by itself.
4. The employed bee of an abandoned solution point becomes a scout and starts to search a new solution point randomly.

The proposed algorithm is an advanced ABC algorithm. For effectively searching a global optimum solution, ABC algorithm is improved as follows:

1) Improvement of Eq.(1) to compute fitness (fit_i^k)
For improving the adaptability to various engineering problems, in the proposed algorithm, the fitness (fit_i^k) of each solution point is computed as follows:

$$fit_i^k = \begin{cases} \dfrac{1}{f(\boldsymbol{x}_i^k)-f_{bound}}, & f(\boldsymbol{x}_i^k)-f_{bound} \geq f_{accuracy} \\ \dfrac{1}{f_{accuracy}}, & f(\boldsymbol{x}_i^k)-f_{bound} < f_{accuracy} \end{cases} \quad (i=1,\ldots,SN) \tag{3}$$

where f_{bound} represents the boundary value of $f(\boldsymbol{x}^+)$ on $\boldsymbol{x}^+$ acceptable as a solution for every engineering problem and $f_{accuracy}$ shows the exactness of convergence to f_{bound}.

2) Improvement of Step1 and Step3 in ABC algorithm
For improving the performance of solution search, in the proposed algorithm, a more suitable solution point is determined by roulette or elite selection based on the probability (P_i^k) of Eq.(2).

3) Improvement of Step4 in ABC algorithm
The search by scouts corresponds to the mutation of Genetic Algorithms. In the proposed algorithm, the search by scouts is not executed to no effect.

III. EXPERIMENTAL RESULTS

Through numerical experiments on the following D dimensional benchmark functions, the performance of the proposed algorithm is investigated to verify its effectiveness.

- Rosenbrock function

$$\text{min. } f_1(\boldsymbol{x}) = \sum_{j=1}^{D-1}\left\{100(x_{j+1}-x_j^2)^2+(x_j-1)^2\right\}$$
$$\text{subj. to } -100 \leq x_j \leq 100, \quad j=1,\ldots,D$$
$$\boldsymbol{x}^* = (1,\ldots,1), \quad f_1(\boldsymbol{x}^*)=0$$

- Rastrigin function

$$\text{min. } f_2(\boldsymbol{x}) = \sum_{j=1}^{D}\left\{x_j^2-10\,cos(2\pi x_j)+10\right\}$$
$$\text{subj. to } -5.12 \leq x_j \leq 5.12, \quad j=1,\ldots,D$$
$$\boldsymbol{x}^* = (0,\ldots,0), \quad f_2(\boldsymbol{x}^*)=0$$

- Schwefel function

$$\text{min. } f_3(\boldsymbol{x}) = 418.98288727\,D + \sum_{j=1}^{D} -x_j\,sin(\sqrt{|x_j|})$$
$$\text{subj. to } -512 \leq x_j \leq 512, \quad j=1,\ldots,D$$
$$\boldsymbol{x}^* = (420.968750,\ldots,420.968750), \quad f_3(\boldsymbol{x}^*)=0$$

- Ackley function

$$\text{min. } f_4(\boldsymbol{x}) = 20+e-20\,exp\left(-0.2\sqrt{\frac{1}{D}\sum_{j=1}^{D}x_j^2}\right) - exp\left(\frac{1}{D}\sum_{j=1}^{D}cos(2\pi x_j)\right)$$
$$\text{subj. to } -30 \leq x_j \leq 30, \quad j=1,\ldots,D$$
$$\boldsymbol{x}^* = (0,\ldots,0), \quad f_4(\boldsymbol{x}^*)=0$$

- Griewank function

$$\text{min. } f_5(\boldsymbol{x}) = \frac{1}{4000}\sum_{j=1}^{D}x_j^2 - \prod_{j=1}^{D}cos\left(\frac{x_j}{\sqrt{j}}\right)+1$$
$$\text{subj.to } -600 \leq x_j \leq 600, \quad j=1,\ldots,D$$
$$\boldsymbol{x}^* = (0,\ldots,0), \quad f_5(\boldsymbol{x}^*)=0$$

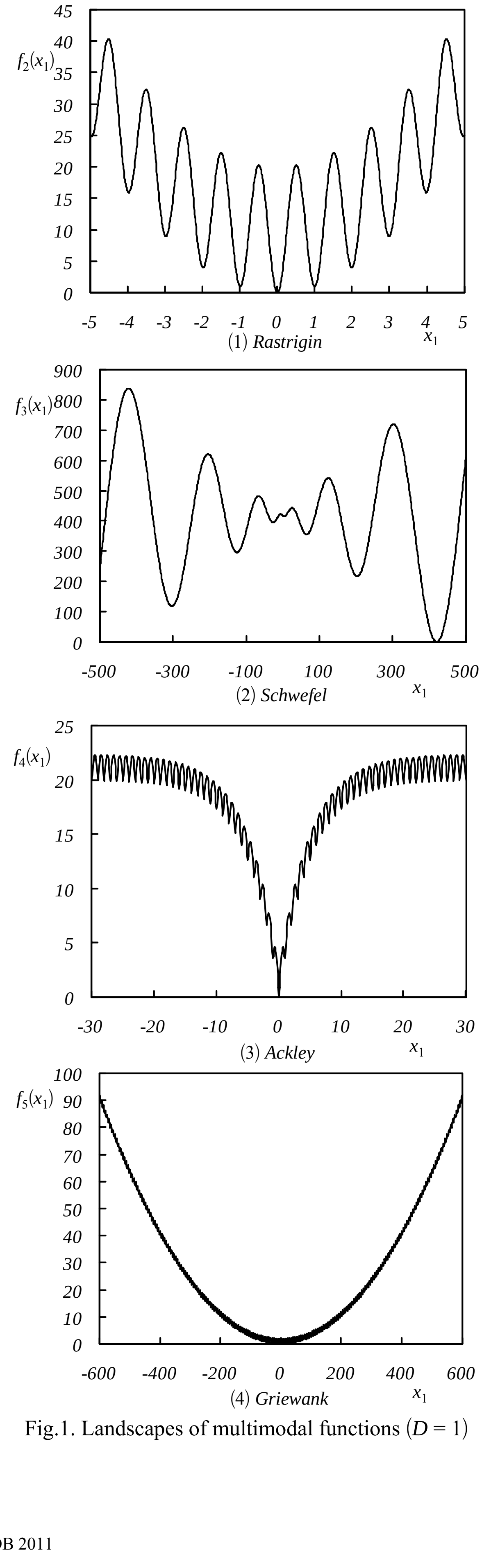

Fig.1. Landscapes of multimodal functions $(D = 1)$

x*** of each benchmark function represents a global optimum solution. Fig.1 shows landscapes of multimodal functions $(D = 1)$.

In experimental results reported, the proposed algorithm is evaluated through the comparison with the existing ones, which are Particle Swarm Optimization algorithm[1,2], Differential Evolution algorithm[3], and the original ABC algorithm[4]. The results of numerical experiments are shown in Tables 1, 2, 3, and 4. Experimental results indicate that the proposed algorithm is a promising one for large-scale optimization problems.

IV. CONCLUSIONS

In this paper, a new optimization algorithm based on Artificial Bee Colony (ABC) algorithm has been proposed. The proposed algorithm has been evaluated through numerical experiments on well-known benchmark functions, such as *Rosenbrock* function, *Rastrigin* function, *Schwefel* function, *Ackley* function, and *Griewank* function. From experimental results, it has been confirmed that the proposed algorithm has the development potential as an efficient one for large-scale problem.

REFERENCES

[1] Kennedy J and Eberhart RC (1995), Particle swarm optimization. Proc. IEEE International Conference on Neural Networks, pp.1942-1948

[2] Kennedy J and Eberhart RC (2001), Swarm intelligence. Morgan Kaufmann Publishers, San Francisco.

[3] Storn R and Price K (1997), Differential evolution - A simple and efficient heuristic for global optimization over continuous space. J. of Global Optimization, vol.11, pp.341-359

[4] Karaboga D and Basturk B (2007), A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Global Optimization, vol.39, pp.459-471.

Table 1 Experimental results on PSO algorithm

Problem	Dim.	Best	Ave.	Worst
Rosenbrock	50	4.38×10^{3}	7.00×10^{4}	2.26×10^{5}
	75	9.20×10^{3}	1.79×10^{5}	4.63×10^{5}
	100	5.08×10^{4}	7.28×10^{5}	3.18×10^{6}
	150	9.86×10^{4}	1.48×10^{6}	5.91×10^{6}
Rastrigin	50	1.05×10^{2}	1.81×10^{3}	3.14×10^{3}
	75	1.58×10^{2}	2.30×10^{3}	3.26×10^{3}
	100	2.72×10^{2}	3.71×10^{3}	6.10×10^{3}
	150	3.54×10^{2}	6.60×10^{3}	9.31×10^{3}
Schwefel	50	5.39×10^{3}	9.78×10^{3}	1.25×10^{4}
	75	9.20×10^{3}	3.48×10^{4}	4.52×10^{4}
	100	1.92×10^{4}	6.92×10^{4}	9.88×10^{4}
	150	3.35×10^{4}	4.07×10^{5}	5.65×10^{5}
Ackley	50	1.82×10^{0}	2.11×10^{1}	2.65×10^{1}
	75	2.54×10^{0}	4.05×10^{1}	6.02×10^{1}
	100	4.84×10^{0}	2.40×10^{2}	4.30×10^{2}
	150	5.79×10^{0}	2.61×10^{2}	8.90×10^{2}
Griewank	50	1.23×10^{0}	1.91×10^{1}	2.13×10^{1}
	75	3.89×10^{0}	4.93×10^{1}	9.83×10^{1}
	100	4.01×10^{0}	7.12×10^{1}	1.51×10^{2}
	150	5.99×10^{0}	7.05×10^{1}	1.87×10^{2}

Table 2 Experimental results on DE algorithm

Problem	Dim.	Best	Ave.	Worst
Rosenbrock	50	9.01×10^{1}	1.32×10^{2}	9.02×10^{2}
	75	7.09×10^{2}	1.38×10^{3}	1.92×10^{3}
	100	1.08×10^{4}	1.73×10^{4}	2.79×10^{4}
	150	8.45×10^{5}	1.97×10^{6}	4.12×10^{6}
Rastrigin	50	7.62×10^{-2}	4.55×10^{0}	4.23×10^{1}
	75	1.53×10^{1}	8.77×10^{1}	1.81×10^{2}
	100	1.54×10^{2}	2.21×10^{2}	3.22×10^{2}
	150	2.98×10^{2}	5.59×10^{2}	7.38×10^{2}
Schwefel	50	1.14×10^{3}	2.24×10^{3}	4.62×10^{3}
	75	6.27×10^{3}	9.87×10^{3}	1.11×10^{4}
	100	1.24×10^{4}	3.45×10^{4}	4.10×10^{4}
	150	5.72×10^{4}	6.70×10^{4}	7.54×10^{4}
Ackley	50	9.89×10^{-4}	2.44×10^{-3}	3.56×10^{-3}
	75	2.66×10^{-1}	8.52×10^{-1}	7.25×10^{0}
	100	5.26×10^{0}	7.81×10^{0}	8.01×10^{0}
	150	8.59×10^{0}	9.15×10^{0}	9.45×10^{0}
Griewank	50	5.42×10^{-6}	9.80×10^{-3}	7.30×10^{-2}
	75	5.72×10^{-2}	1.47×10^{-1}	3.89×10^{-1}
	100	4.19×10^{-1}	6.24×10^{-1}	3.83×10^{0}
	150	2.02×10^{0}	3.08×10^{0}	5.31×10^{0}

Table 3 Experimental results on ABC algorithm

Problem	Dim.	Best	Ave.	Worst
Rosenbrock	50	1.36×10^{0}	1.25×10^{2}	8.80×10^{2}
	75	2.30×10^{0}	5.36×10^{2}	9.79×10^{3}
	100	9.96×10^{1}	9.07×10^{2}	9.82×10^{3}
	150	9.76×10^{2}	2.85×10^{3}	1.40×10^{4}
Rastrigin	50	4.51×10^{-4}	3.53×10^{0}	1.25×10^{1}
	75	4.93×10^{0}	2.42×10^{1}	9.82×10^{1}
	100	6.22×10^{1}	7.16×10^{1}	1.02×10^{2}
	150	1.28×10^{2}	2.77×10^{2}	3.22×10^{2}
Schwefel	50	8.98×10^{2}	1.30×10^{3}	1.97×10^{3}
	75	2.73×10^{3}	6.01×10^{3}	6.56×10^{3}
	100	6.02×10^{3}	9.31×10^{3}	1.00×10^{4}
	150	9.60×10^{3}	1.62×10^{4}	3.20×10^{4}
Ackley	50	4.74×10^{-4}	1.71×10^{-3}	2.24×10^{-2}
	75	1.30×10^{-1}	7.96×10^{-1}	1.58×10^{0}
	100	1.89×10^{0}	2.49×10^{0}	4.12×10^{0}
	150	4.57×10^{0}	5.75×10^{0}	6.68×10^{0}
Griewank	50	8.50×10^{-7}	9.21×10^{-3}	3.73×10^{-2}
	75	4.26×10^{-4}	4.57×10^{-2}	1.28×10^{-1}
	100	8.17×10^{-3}	7.94×10^{-2}	5.34×10^{-1}
	150	1.79×10^{-2}	7.17×10^{-1}	9.97×10^{-1}

Table 4 Experimental results on the proposed algorithm

Problem	Dim.	Best	Ave.	Worst
Rosenbrock	50	1.39×10^{-2}	8.73×10^{0}	9.71×10^{1}
	75	7.01×10^{-1}	2.57×10^{1}	1.14×10^{2}
	100	5.75×10^{0}	5.34×10^{1}	2.10×10^{2}
	150	9.98×10^{1}	1.32×10^{2}	4.39×10^{3}
Rastrigin	50	2.08×10^{-10}	8.10×10^{-1}	2.99×10^{0}
	75	2.80×10^{0}	8.00×10^{0}	1.57×10^{1}
	100	1.21×10^{1}	1.39×10^{1}	2.92×10^{1}
	150	4.67×10^{1}	5.66×10^{1}	9.53×10^{1}
Schwefel	50	1.18×10^{2}	2.03×10^{2}	1.18×10^{3}
	75	1.43×10^{3}	2.39×10^{3}	3.06×10^{3}
	100	2.54×10^{3}	3.03×10^{3}	5.26×10^{3}
	150	6.78×10^{3}	9.02×10^{3}	9.91×10^{3}
Ackley	50	3.30×10^{-9}	5.70×10^{-5}	2.21×10^{-3}
	75	1.85×10^{-5}	2.44×10^{-4}	1.55×10^{-3}
	100	1.51×10^{-3}	7.17×10^{-2}	3.74×10^{-1}
	150	1.48×10^{-1}	1.96×10^{-1}	2.46×10^{0}
Griewank	50	1.14×10^{-16}	9.13×10^{-5}	4.56×10^{-3}
	75	7.92×10^{-11}	3.61×10^{-3}	6.84×10^{-2}
	100	4.00×10^{-8}	5.49×10^{-3}	3.76×10^{-2}
	150	1.74×10^{-4}	1.68×10^{-2}	2.56×10^{-1}