Far East Journal of Mathematical Sciences (FJMS)
© 2018 Pushpa Publishing House, Allahabad, India
http://www.pphmj.com
http://dx.doi.org/10.17654/MS103081303
Volume 103, Number 8, 2018, Pages 1303-1321 **ISSN: 0972-0871**

GLOBAL OPTIMIZATION WITH SIMPLEXED ARTIFICIAL BEES COLONY ALGORITHM

Goh Khang Wen[1,2,*], Yosza Bin Dasril[2] and Abdul Rani Bin Othman[2]

[1]Faculty of Integrative Science and Technology
Quest International University Perak
Jalan Raja Permaisuri Bainun
30250 Ipoh, Perak, Malaysia

[2]Faculty of Electronics and Computer Engineering
Universiti Teknikal Malaysia Melaka
Hang Tuah Jaya, 76100 Durian Tunggal
Melaka, Malaysia

Abstract

Artificial bees colony (ABC) algorithm is one of the most recent biological-inspired optimization algorithms proposed in 2005. ABC is inspired by the foraging behavior of honey bee swarm. The exploration and exploitation are two important mechanisms in ABC. Exploitation process starts when the employed bees approach to the food sources. However, ABC can perform better during exploration stage but weaker at exploitation stage. In this study, the exploitation search stage has been integrated with Nelder-Mead simplex method (NM). A new method called NM-ABC is developed. The numerical results show that the NM-ABC can perform better than the original ABC with less colony size and number of cycles allocated.

Received: May 31, 2017; Accepted: October 8, 2017
2010 Mathematics Subject Classification: 90C59, 65K10.
Keywords and phrases: artificial bees colony algorithm, Nelder-Mead simplex method, global optimization.
*Corresponding author

1. Introduction

Swarm intelligence is the natural or artificial system that consists of collective behavior of decentralization and self-organization. This expression was introduced by Beni and Wang in [3]. Swarm intelligence is usually based on the collective behaviors of decentralized and self-organized swarms such as social insects, flocks of birds or schools of fish with self-organization and division of labor as the main components. Self-organization is the interaction among its low-level components while the division of labor is the simultaneous task performed by cooperating specialized individuals. These two fundamental concepts are necessary and sufficient properties to obtain swarm intelligent behavior.

Artificial bees colony (ABC) algorithm is one of the most recent swarm intelligent based algorithms proposed by Karaboga in [6]. It is a biological-inspired optimization algorithm. ABC is inspired by the foraging behavior of honey bee swarm. It is very straightforward, robust and population-based on optimization algorithm. In ABC algorithm, the process begins by dividing the colony of artificial bees into two groups: employed bees and unemployed bees. Employed bees are associated with the particular food source which they are currently exploiting. There are two types of unemployed bees which are the groups of foragers that continually look out for a food source to exploit: scouts, randomly search the environment surrounding the nest for new food sources and onlookers waiting in the nest and establishing a food source through the information shared by the employed bees. The number of employed bees is equal to the number of food sources as the constraint because there is only one employed bee for every food source. Fitness value is calculated to check the quality of the food source and is associated with its position. The process of the swarm of bees searching for food source is the process used to find optimal solution (Karaboga [6]).

The exploration and exploitation are two important mechanisms in ABC. Exploitation process starts when the employed bees approach to the food sources. After determining the nectar amounts of the food sources by the employed bees, the onlooker bees will go to the highest probability value of source and determine the nectar amount. When the source is exhausted, it indicated the end of the exploitation process. Meanwhile, exploration process begins when scouts are sent to search for new food sources randomly.

However, there are some insufficiencies regarding the ABC. ABC can perform better during exploration stage but weaker at exploitation stage. In addition, the convergence speed is also an issue in some problem cases. The exploration process is the ability of searching for global optimum independently while the exploitation is about the distributions of the existing knowledge to look for better solutions (Li et al. [8], Banharnsakun et al. [2] and Zhu and Kwong [12]).

In this paper, we have improved the exploitation search of the original ABC by using Nelder-Mead simplex method (NM). The numerical results show that with the less colony size and number of cycles, the improved ABC still can perform better than the original ABC. This paper is organized as follows. In Section 2, we briefly introduce the description and the algorithm of the original ABC method. In Section 3, we summarize the analysis of several well-known modified ABC algorithms from fellow researchers in literature. Then the description of the process of integration between Nelder-Mead simplex method into the original ABC is presented in Section 4. Following by that are the numerical results which reflect the improvement in the performance of the modified ABC in solving global optimization problems. The conclusion and discussion which end this paper are presented in Section 6.

2. Artificial Bees Colony (ABC) Algorithm

Artificial bees colony (ABC) algorithm is inspired by the foraging behaviors of honey bees swarm. In fact, bees are divided into two groups: employed and unemployed (onlookers and scouts). The bees will first search for a food source and become employed bee when the bees manage to bring the nectars back to their hives. The employed bees can either go back to their discovered source site or spread it to the onlookers by performing a dance in dancing area. The onlookers will select one profitable source by watching the dance advertising according to the quality of the source. When a source is exhausted or abandoned, the employed bees become a scout and start to randomly search for a new source (Akay and Karaboga [1]).

Description

This mechanism is applied in ABC algorithm. First of all, randomly distributed initial food source positions are generated by the objective function values of the sampled point from each employed bees. The process can be represented by $f(x_i)$, $x_i \in R^D$, $i \in \{1, 2, 3, \ldots, SN\}$. x_i is a position of food source as D-dimensional vector, where D is the number of optimization parameters in the model. $f(x_i)$ is the objective function which determines the quality of the solution, and SN is the number of food sources. After the initialization, the population is subjected to repeated cycles of the three major steps that are updating feasible solutions, selecting feasible solutions and avoiding suboptimal solutions (Banharnsakun et al. [2]). In order to test the fitness value fit_i of the new food source, the employed bee could produce a modification on the solution in its memory:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}). \tag{1}$$

In equation (1), v_{ij} is the new feasible solution and SN is the size of food source which generated, ϕ_{ij} is the random number between [0, 1] which is used to randomly adjust the old solution becoming the new solution in the next iteration. $k \in \{1, 2, \ldots, SN\}$ and $j \in \{1, 2, \ldots, D\}$ are random generated indexes but k must be different from i. When the difference between x_{ij} and x_{kj} decreases, the perturbation of the position also decreases. Thus, the step size is reduced when the search approaches to the optimum solution in the search space (Li et al. [8]):

$$fit_i = \begin{cases} \dfrac{1}{1 + f(x_i)} & \text{if } f(x_i) \geq 0, \\ 1 + abs(f(x_i)) & \text{if } f(x_i) < 0. \end{cases} \tag{2}$$

The fitness value is proportional to the nectar amount of the food source in the *i*th position. If the fitness value is better than the previous one, the employed bee would memorize the new food position and forget the old one. Otherwise, it keeps the current food position in its memory. Information about nectar amount and positions of food will be shared by the employed bees when all of them completed the searching process to the onlookers. The onlookers will then evaluate the nectar information by all the employed bees and choose a food source according to the probability which is related to the nectar amount. Therefore, during onlooker bees phase, new solution v_{ij} is produced for the solutions x by means of their fitness values by using the formula of the probability of the fitness. The onlookers can produce a modification on the position in its memory as what employed bees do. The onlookers check the nectar amount of the candidate source. If the nectar amount is better than that of the previous one, the bee would memorize the new position instead of the previous one. An onlooker chooses a food source completely according to the probability value associated with the food source, p_i where fit_i, the fitness value of the *i*th solution is:

$$p_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j}. \tag{3}$$

The food source which is exhausted or abandoned by the bees would be replaced with a new food source found by scout bees. The function values will be identified as abandoned values when they undergo a specific number of trials and the solutions cannot be improved. Then a new solution will be generated randomly to replace the abandoned one. The new random position chosen will be calculated by using the equation as:

$$x_i^j = x_{\min}^j + rand(0, 1)(x_{\max}^j - x_{\min}^j), \tag{4}$$

where $x_{\max}^j$ is the upper bound of the food source position in dimension j while $x_{\min}^j$ is the lower bound of the food source position in dimension j.

The boundaries act as one of the constraints of the algorithm. When parameters generated exceed the boundaries, they will be shifted onto the boundaries. Besides that, $rand(0, 1)$ represents the random number between $[0, 1]$ and the maximum number of cycles (MCN) is used to control the number of iterations and it acts as a termination criterion.

Algorithm of artificial bees colony

Initialize the population of solutions x_{ij}, $i = 1, 2, \ldots, SB$, $j = 1, 2, ..., n$, $trial = 0$ is the non-improvement number of the solutions x_{ij}, used for abandonment

Evaluate the population

Set cycle = 1

Repeat

{Produce a new food source population for employed bees}

for $i = 1$ to SN **do**

Produce a new food source v_i for the employed bee of the food source x_i using **equation (1)** and evaluate its quality

Apply a greedy selection process between v_i and x_i and select the better one

If solution x_i does not improve $trial_i = trial_i + 1$, otherwise, $trial_i + 1 = 0$

end for

Calculate the probability values p_i by **equation (3)** for the solutions using fitness values

{Produce a new food source population for onlooker bees}

$t = 0$, $i = 1$

repeat

if $random < p_i$ then

Produce a new v_i food source by **equation (1)** for onlooker bee

Apply a greedy selection process between v_i and x_i and select the better one

If solution x_i does not improve $trial_i = trial_i + 1$, otherwise $trial_i = 0$, $t = t + 1$

end if

until $(t = SN)$

{Determine scout}

if $\max(trial_i) >$ limit **then**

Replace x_i with a new randomly produced solution by **equation** (4)

end if

Memorize the best solution achieved so far

$cycle = cycle + 1$

until (cycle = maximum cycle number)

end algorithm (Gao and Liu [4])

3. The Development of Artificial Bees Colony (ABC) Algorithm

Tereshko and Loengarov [10] had started to solve problems by using honey bee foraging dynamics. They were interested in seeing how the exchanging information interactions between the individual lead to globally intelligent selection of the food sources in an unpredictable environment (Tereshko and Loengarov [10]). Hence, they started to develop a model which will be able to quickly search for the "best" food source by considering the bee colony as dynamic system (Tereshko and Loengarov [10]). The system consisted of three essential components, which are the food sources, employed bees and unemployed bees. Meanwhile, the leading modes of the foraging behavior of the bees are recruitment to a nectar source and abandonment of the source.

In the same year, Dervis Karaboga was inspired by this idea and initiated artificial bees colony (ABC) which is also an algorithm which adapts to the honey bee swarm's foraging behavior. Similarly, the model included the three essential components as mentioned above. According to Karaboga [6], ABC is very simple and flexible compared to the existing swarm based algorithms. Recently, ABC algorithm had been reviewed by many professional researchers.

In [12], Zhu and Kwong did some modifications on the ABC into global best-guided ABC (Gbest ABC). The modification of ABC into Gbest ABC was inspired by population-based optimization algorithms (PSO), which, in order to improve the exploitation process, took advantage of the information of the global best solution, by modifying the equation:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{5}$$

becoming

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \varphi_{ij}(x_j - x_{ij}) \leftarrow GbestTerm, \tag{6}$$

where x_j is the jth element of global best solution, φ_{ij} is the uniform random number in $[0, c]$ and c is the non-negative constant. When the number c is increased, the efficiency of the exploitation process will be improved. At the same time, the number c cannot be too large because it weakens the exploitation process, at the same time causing the Gbest term driving the new candidate solution moves over the global best solution (Zhu and Kwong [12]). However, after some experiments had been carried out by both of the researchers, it is noticed that, GABC outperformed the original ABC in most of the experiments when $c = 1.5$. Therefore, this shows that GABC can perform better than the original ABC with appropriate parameter applied.

Banharnsakun and his fellow researchers had initiated best-so-far ABC in [2], by adding three things to the modification of ABC that are, best-so-far method, adjustable search radius and objective-value-based comparison method.

In the original ABC algorithm, all the onlookers choose a food source based on the probability of respective fitness function explored by a single employed bee and the new solutions are generated by using equation (5). On the contrary, in the best-so-far method, the onlookers make decision on new food sources by making use of all the information from all employed bees so that they can compare all the information that are available and are able to select the best-so-far food position. The modified equation to generate new food source is:

$$v_{id} = x_{ij} + \phi f_k (x_{ij} - x_{kj}), \tag{7}$$

where f_k is the fitness value of the best food source so far; x_{kj} is the best-so-far food source selected dimension j. This method is able to improve the local search ability compared to the original ABC algorithm. According to the results from numerical experiments conducted by Banharnsakun and his fellow researchers, best-so-far ABC obtained a better convergence rate than the original ABC. A smaller rate of convergence indicates that less iteration is needed for a function to converge to the optimal solution. Results showed that best-so-far ABC can produce the optimal solution more quickly on almost all benchmark functions (Banharnsakun et al. [2]).

After that, Li et al. [8] proposed an improved ABC called *I-ABC* and another *PS-ABC* with the ability of prediction and selection. The latter is the combination of the bright sides from ABC, GABC and I-ABC. Before knowing what is PS-ABC, best-so-far solution, inertia weight and acceleration coefficients are introduced to modify the searching process in I-ABC. I-ABC could not only find the global optimal values for many numerical functions, but also owns an extremely fast convergence speed. Yet, in some cases, I-ABC traps in local optimal and therefore not able to find better solutions than ABC or GABC. The equation is modified as the following form:

$$v_{ij} = x_{ij} w_{ij} + 2(\phi_{ij} - 0.5)(x_{ij} - x_{kj})\phi_1 + \varphi_{ij}(x_j - x_{kj})\phi_2, \qquad (8)$$

w_{ij} is the inertia weight which controls impacts of the previous solution x_{ij}. x_j is the *j*th element of global best solution ϕ_{ij}, and φ_{ij} are random numbers between $[0, 1]$, ϕ_1 and ϕ_2 are positive parameters that could control the maximum step size. Somehow, when the global fitness is very large, bees are further away from optimal solutions (Li et al. [8]).

For the purpose of producing high efficient ABC algorithm with the abilities to predict and select, the researchers produced the PS-ABC by gathering all the bright sights of ABC, GABC and I-ABC to form a hybrid ABC algorithm. The main difference between PS-ABC and any of ABC, GABC and I-ABC is how to determine the candidate solutions process. In PS-ABC, the employed bees will firstly work out 3 possible solutions with 3 types of search equations and then choose and determine the best one as the candidate solution (Li et al. [8]).

From original ABC, $v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$.

From I-ABC, $v_{ij} = x_{ij} w_{ij} + 2(\phi_{ij} - 0.5)(x_{ij} - x_{kj})\phi_1 + \varphi_{ij}(x_j - x_{kj})\phi_2$.

From GABC, $v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \varphi_{ij}(x_j - x_{ij})$.

Both the I-ABC and PS-ABC were tested with 13 classical functions comparing to the solutions from the ABC and GABC. It was found that I-ABC obtained faster convergence speed than ABC or GABC for most functions although it did not achieve better optimality ability than the ABCs in few of the functions. The results showed that the convergence and searching ability generated by using PS-ABC is better than the other methods for almost all functions. In PS-ABC, the global search ability had increased and convergence ability of this algorithm had been enhanced at the same time (Li et al. [8]). This shows that there is no specific algorithm to substantially achieve the best solutions for all the optimization problems. Some algorithms give best solutions in some cases and some not. Hence, researchers nowadays try to search for a well improved or new optimization method.

Table 3.1. The transformation of original ABC into PS-ABC

Original ABC (2005)	$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$
Best-so-far ABC (2010)	$v_{id} = x_{ij} + \phi f_k(x_{ij} - x_{kj})$
Gbest ABC (GABC) (2011)	$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \varphi_{ij}(x_j - x_{ij})$
I-ABC (2012)	$v_{ij} = x_{ij}w_{ij} + 2(\phi_{ij} - 0.5)(x_{ij} - x_{kj})\phi_1 + \varphi_{ij}(x_j - x_{kj})\phi_2$
PS-ABC (2012)	From original ABC, $v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$
	From I-ABC, $v_{ij} = x_{ij}w_{ij} + 2(\phi_{ij} - 0.5)(x_{ij} - x_{kj})\phi_1 + \varphi_{ij}(x_j - x_{kj})\phi_2$
	From GABC, $v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \varphi_{ij}(x_j - x_{ij})$

4. Modified Artificial Bees Colony Algorithm

Generally, this section is divided into two major subsections. First subsection discusses about how the Nelder-Mead simplex method works and the algorithm is provided in order to make it clearer. Then, in the second subsection, we will explain the integration between Nelder-Mead simplex methods into the original ABC.

Nelder-Mead simplex method

The Nelder-Mead method (Nelder and Mead [9]) can be changed in five different ways during an iteration, as illustrated here in two dimensions. The algorithm begins with a simplex and then one of the vertices will be replaced with the new point and so the progress repeats. Except in the case of a shrink, the worst vertex of the simplex at iteration k is replaced at iteration $k+1$ by one of the reflextion, expansion, or contraction points (Wright and Margaret [11]). The iteration of this algorithm will be terminated when a new simplex such that the function values at its vertices satisfy some form of descent condition compared to the previous simplex (Lagarias et al. [7]).

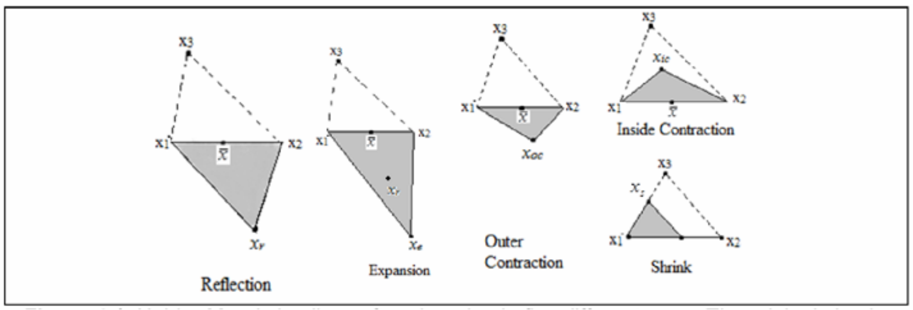

Figure 4.1. Nelder-Mead simplices after changing in five different ways. The original simplex is shown with a dashed line (Wright and Margaret [11]).

In short, this simplex method adapts itself to the local landscare, elongating down long inclined planes, changing direction on encountering a valley at an angle, and contracting in the neighbourhood of a minimum (Nelder and Mead [9]).

NM-ABC algorithm

The exploitation search of the original ABC and best-so-far ABC is obtained through another different bee as the guide for new search direction. Thereafter, the exploitation search direction equation used by G-ABC and I-ABC include another two bees as the guide for new search direction. These ideas have inspired the implementation of Nelder-Mead simplex method into original ABC via using the other two bees to form the initial triangle for Nelder-Mead simplex method. The new modified ABC we named as NM-ABC.

Basically, the algorithm of the NM-ABC is quite similar to the original ABC, only in the phase of producing new food source for employed bees and onlooker having integrated with NM method. Equation (1) in the algorithm has been replaced with Nelder-Mead simplex method.

5. Numerical Results

In this section, the original ABC and NM-ABC algorithm as presented in previous sections have been programmed into C++ and implemented for solving several benchmark global optimization problems (Wen [5]). Numerical results have been reported.

Results of original ABC

The following table shows the mean of optimal solution obtained from the original ABC by different colony size after 30 runs for each of the following problem. The results showed that the colony size and the number of cycles are directly proportional to the accuracy of the solution. This statement explained how the colony size and the number of cycles affect the results. When colony size and number of cycles is large at the same time, the result obtained will be better than that with small colony size and less number of cycles.

Function	Colony size	Number of cycles	Mean of the optimum value for 30 runs
Rosenbrock $f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2$	10	100	1.093553×10^{-1}
		1000	3.370321×10^{-2}
		10000	3.490372×10^{-3}
	20	100	1.023259×10^{-1}
		1000	3.834186×10^{-3}
		10000	9.330526×10^{-4}
	50	100	1.347883×10^{-2}
		1000	1.856689×10^{-3}
		10000	1.859279×10^{-4}
	100	100	3.078544×10^{-3}
		1000	8.871076×10^{-4}
		10000	1.204077×10^{-4}
Sphere $f(x_1, x_2) = x_1^2 + x_2^2$	10	100	3.28935×10^{-17}
		1000	4.25346×10^{-18}
		10000	2.194551×10^{-19}
	20	100	1.466006×10^{-17}
		1000	1.657043×10^{-18}
		10000	1.591113×10^{-19}
	50	100	5.714456×10^{-18}
		1000	4.792367×10^{-19}
		10000	6.293019×10^{-20}
	100	100	3.141911×10^{-18}
		1000	3.22466×10^{-19}
		10000	3.698771×10^{-20}

Treccani function $f(x_1, x_2) = x_1^4 + 4x_1^3 + 4x_1^2 + x_2^2$	10	100	5.874849×10^{-14}
		1000	1.904501×10^{-18}
		10000	2.670346×10^{-20}
	20	100	1.197651×10^{-14}
		1000	5.975909×10^{-19}
		10000	6.130373×10^{-21}
	50	100	1.290156×10^{-17}
		1000	5.82669×10^{-20}
		10000	8.304195×10^{-22}
	100	100	1.630461×10^{-18}
		1000	2.254478×10^{-20}
		10000	8.853060×10^{-23}
Booth function $f(x_1, x_2) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	10	100	6.800595×10^{-5}
		1000	1.262258×10^{-16}
		10000	7.780096×10^{-18}
	20	100	2.129839×10^{-5}
		1000	4.608728×10^{-17}
		10000	3.950874×10^{-18}
	50	100	3.681047×10^{-6}
		1000	1.689070×10^{-17}
		10000	1.050576×10^{-18}
	100	100	5.608005×10^{-7}
		1000	8.644657×10^{-18}
		10000	9.704818×10^{-19}

Beale's function $f(x_1, x_2) = [1.5 - x_1(1 - x_2)]^2 + [2.25 - x_1(1 - x_2^2)]^2 + [2.625 - x_1(1 - x_2^3)]^2$	10	100	6.255732×10^{-2}
		1000	4.615743×10^{-2}
		10000	2.028533×10^{-14}
	20	100	3.995236×10^{-3}
		1000	6.222835×10^{-6}
		10000	1.549315×10^{-15}
	50	100	1.724260×10^{-3}
		1000	2.570454×10^{-9}
		10000	8.534780×10^{-16}
	100	100	9.397632×10^{-4}
		1000	4.043348×10^{-10}
		10000	7.253231×10^{-16}

Results of NM-ABC with different numbers of cycles

(1) Rosenbrock function $f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2$

Colony size	Number of cycles	Mean of the optimum value for 30 runs
10	1	1.083445×10^{-26}
10	10	8.735773×10^{-27}
10	50	5.557070×10^{-27}

(2) Sphere function $f(x_1, x_2) = x_1^2 + x_2^2$

Colony size	Number of cycles	Mean of the optimum value for 30 runs
10	1	1.415157×10^{-26}
10	10	7.567237×10^{-27}
10	50	4.864613×10^{-27}

(3) Treccani function $f(x_1, x_2) = x_1^4 + 4x_1^3 + 4x_1^2 + x_2^2$

Colony size	Number of cycles	Mean of the optimum value for 30 runs
10	1	8.866011×10^{-27}
10	10	4.995395×10^{-27}
10	50	1.392662×10^{-27}

(4) Booth function $f(x_1, x_2) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$

Colony size	Number of cycles	Mean of the optimum value for 30 runs
10	1	1.160783×10^{-26}
10	10	7.577246×10^{-27}
10	50	4.768901×10^{-27}

(5) Beale's function

$$f(x_1, x_2) = [1.5 - x_1(1 - x_2)]^2 + [2.25 - x_1(1 - x_2^2)]^2 + [2.625 - x_1(1 - x_2^3)]^2$$

Colony size	Number of cycles	Mean of the optimum value for 30 runs
10	1	1.415157×10^{-26}
10	10	7.567237×10^{-27}
10	50	4.488875×10^{-27}

In the results, the modified NM-ABC has shown that better optimum values can be obtained even with the lower colony size and less number of cycles compared to the results shown by original ABC in Section 5. Besides that, the result also shows that the number of cycles does not illustrate any significant increase after the number of cycles has been increased from 1 to 50. Therefore, as to remain the foraging behavior of the bees, the colony size for NM-ABC will be set as 10 and limit to 10 cycles for each colony in foraging process.

6. Conclusion

In this paper, we have introduced a new developed ABC called *NM-ABC* which integrates NM into original ABC. The numerical results exhibit that, in the original ABC as the colony size and the number of cycles become larger, the optimum solution obtained is more precise. However, the modified artificial bees colony (ABC) in this study can obtain much better optimum solution even with the smaller colony size and number of cycles.

References

[1] B. Akay and D. Karaboga, A modified artificial bee colony algorithm for real-parameter optimization, Inform. Sci. 192 (2012), 120-142.

[2] A. Banharnsakun, T. Achalakul and B. Sirinaovakul, The best-so-far selection in artificial bee colony algorithm, Applied Soft Computing 11 (2011), 2888-2901.

[3] G. Beni and J. Wang, Swarm intelligence in cellular robotic systems, Proc. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, June 26-30, 1989.

[4] W.-F. Gao and S.-Y. Liu, A modified artificial bee colony algorithm, Comput. Oper. Res. 39 (2012), 687-697.

[5] Goh Khang Wen, Global optimization using nonparametric filled function method, M.Sc. Thesis, Universiti Malaysia Terengganu, 2009.

[6] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Technical Report-TR06, Erciyes University, October 2005.

[7] J. C. Lagarias, J. A. Reeds, M. H. Wright and P. R. Wright, Convergence properties of the Nelder-Mead simplex method in low dimensions, SIAM J. Optim. 9 (1998), 112-147.

[8] G. Li, P. Niu and X. Xiao, Development and investigation of efficient artificial bee colony algorithm for numerical function optimization, Applied Soft Computing 12 (2012), 320-332.

[9] J. A. Nelder and R. Mead, A simplex method for function minimization, Computer Journal 7 (1965), 308-313.

[10] V. Tereshko and A. Loengarov, Collective decision-making in honey bee foraging dynamics, Journal of Computing and Information Systems 9(3) (2005), 1-7.

[11] Margaret H. Wright, Nelder, Mead and the other simplex method, Documenta Math. ISPM (2012), 271-276.

[12] G. P. Zhu and S. Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization, Appl. Math. Comput. 217 (2010), 3166-3173.