International Journal of Hybrid Information Technology
Vol. 9, No.12 (2016), pp. 127-140
http://dx.doi.org/10.14257/ijhit.2016.9.12.12

A Cooperative Coevolution Algorithm Based on ABC and NMSM

Hongsheng Su, Kaile Yin and Zihan Wu

School of Automation and Electrical Engineering, Lanzhou Jiaotong University, Lan zhou 730070, China
shsen@163.com

Abstract

Considering that the existing artificial bee colony (ABC) algorithm can not give simultaneously attention to evolution speed and solution quality, an improved ABC algorithm is proposed based on nelder-mead simplex method (NMSM) in this paper, and defined ad NMSM-ABC. In the process of iteration, the algorithm periodically passes the best individual vertex from NMSM operator into ABC, or migrates the optimal food source information from ABC to NMSM can get away from local minimum with aid of ABC. Hence, the proposed algorithm can realize cooperative co-evolution of the two so that the desired properties are obtained. In addition, the sensitivity analysis of the key parameter of NMSM-ABC is also conducted and the best value is suggested. Finally, Numerical experiments and comparisons on 6 benchmark functions are conducted with other ABC algorithms, and the results indicate that the proposed algorithm effectively overcomes the local minimum, and dramatically enhances the global searching ability and convergence speed, and is a good cooperative co-evolution algorithm.

Keywords: *artificial bee colony (ABC); Nelder-mead simplex method (NMSM); cooperative co-evolution; sensitivity analysis; global searching*

1. Introduction

Artificial bee colony algorithm (ABC) is a new-type swarm intelligence optimization algorithm based on honey mechanism of bees, and proposed in 2005[1,2]. The algorithm simulates the honey-searching behavior of bees in nature, according to their respective labor division and food sources information sharing, to find the optimal problem solution. The algorithm is well known with simple structure and less parameter settings, and attracts more and more attention from scholars, and has been become one of research hotspots in the field of intelligent algorithms. The algorithm has been widely applied in the field of function optimization or engineering optimization [3-6]. Comparing with other intelligence optimal algorithms, for instance, differential evolution (DE) [7], and particle swarm optimization (PSO) [8], ABC shows a better performance in solving many optimization problems [9,10]. In order to further improve the convergence of ABC, many scholars put forward the improved methods. Dervis proposed a q-ABC algorithm which models the behavior of onlooker bees to improve local search ability [11]. Li proposed an improved ABC algorithm in which inertia weight and acceleration coefficients are introduced into the searching process of ABC to improve convergence speed [12]. Sharma and Pant suggested the incorporation of DE operators in the structure of basic ABC algorithm [13]. Gao proposed an improved method which employs a Gaussian evolution equation to generate a new candidate individual at the onlooker bee phase to improve the exploitation ability[14]. Kiran proposed an improved method to apply five searching strategies and counters to update the solutions so as to improve search capability[15].

For many ABC algorithms improved, the strategy applied is that the conducted improvements are embedded to the original algorithm only to improve the performance of the algorithm. Clearly, to do so will have some limitations, such as large amount of

ISSN: 1738-9968 IJHIT

calculation or poor ability of global optimization. In fact, although some local search algorithms fall into local optimum easily, they have many advantages such as strong ability of local development, and faster convergence speed, and as well as higher precision.

Based on it, to improve the evolution speed and searching precision of ABC algorithm, further, on the basis of ABC and nelder-mead simplex method (NMSM), a new cooperative co-evolution algorithm is proposed in this paper, and denoted as NMSM-ABC. Other than the improved ABC algorithm mentioned above, the algorithm periodically migrates simplex optimal vertex from NMSM to ABC, or conducts the reverse migration from ABC to NMSM. And thus, the algorithm not only realizes parallel searching, but also balances the global and local exploration ability, and is an effective cooperative co- evolution algorithm.

2. Standard ABC Algorithm

The ABC algorithm finds the solution of the function optimization problem based on the honey mechanism of bees. In the algorithm, the artificial bee colony is divided into three groups, one is employed bees, another one is onlooker bees, and the third is scout bees. Initialized colony is randomly set in range of the specified area, and then sorts them according to their respective fitness values. The half of colony with fitness values better is selected as employed bee colony, and the rest half then acts as onlooker bee colony. Clearly, the number of employed bees is same with the one of onlooker bees. The goal to do so is to ensure that they can find the solution on the same space. Each employed bee corresponds to a food source, whose position represents a feasible solution of optimization problem, whereas food source quality is determined by the fitness value of associated solution. During bee honey-searching, the employed bees exploit their surrounding near food sources on each dimension, whereas an onlooker bee adopts the roulette method to choose the optimal individual, and implements exploiting at its surrounding. The optimum-searching ability is dramatically enhanced through successive iterations between the employed bees and onlooker bees. An employed bee becomes a scout bee as its food source has been exhausted, and then finds food sources once again. In order to overcome loss of colony diversity, ABC algorithm adopts mutation operator of scout bees to form new variation colony, whose size is same as initial colony. The essence of evolution rule of ABC algorithm is to keep excellent individuals and eliminate inferior ones, constantly, and approximate to global optimal solution.

The function of optimization problem is described by

$$\min f = f(\boldsymbol{X}), \boldsymbol{X} = (x_1, x_2, \cdots, x_D), \boldsymbol{X} = \left[\boldsymbol{X}^L, \boldsymbol{X}^H \right]$$

where $\boldsymbol{X}$ is a D-dimension vector, and $\boldsymbol{f}$ is the aim function, and $\boldsymbol{X^L}$,$\boldsymbol{X^H}$ are the lower and upper bound of $\boldsymbol{X}$.

The fitness of the food source $\boldsymbol{X_i}$ can be calculated by

$$fit(\boldsymbol{X}_i) = \begin{cases} \dfrac{1}{1+f(\boldsymbol{X}_i)} & f(\boldsymbol{X}_i) \geq 0 \\ 1+\left| f(\boldsymbol{X}_i) \right| & f(\boldsymbol{X}_i) \leq 0 \end{cases} \tag{1}$$

where $fit(\boldsymbol{X_i})$ is the fitness value of the $\boldsymbol{i}^{\text{th}}$ food source, and $f(\boldsymbol{X_i})$ is the aim function value of $\boldsymbol{X_i}$.

Standarded ABC algorithm is described as follows.

1) Initialization stage. To randomly generate the initialization colony composed of N individuals by

$$\boldsymbol{X}_i = \boldsymbol{X}_L + rand() \times (\boldsymbol{X}_H - \boldsymbol{X}_L), i = 1, 2, \cdots, N \tag{2}$$

where the $\boldsymbol{X_i}$ is the position of the $\boldsymbol{i}^{\text{th}}$ individual of the population, and *rand*( ) generates a random number between zero to one, and N is the number of the colony.

The total colony is then classified as the two groups with same number of individual, one group serves as employed bee colony with fitness values better, and the other is onlooker bee colony.

2) Employed bees exploiting stage. Employed bee exploits new honey source near its surrounding by

$$V_{ij} = x_{ij}^t + \varphi_{ij} \times \left(x_{ij}^t - x_{kj}^t \right), j = 1, 2, \cdots D \tag{3}$$

where V_{ij} expresses the position of new honey source, and x_{ij} represents the j^{th} coordination of the honey source x_i, and x_{kj} represents the j^{th} coordination of the honey source x_k to be selected randomly, and φ_{ij} is a random number between -1 to 1, and the superscript t presents the t^{th} iteration.

To then compare the quality of the honey sources between the original and the new in accordance with (1), and replace the original using the new one if the latter is better than the former.

3) Onlooker bees exploiting stage. Onlooker bees choose a food source region depending on the shared information supplied by employed bees, and which one food source region can be selected finally according to the roulette law described by

$$P_i = \frac{fit(\boldsymbol{X}_i)}{\sum_{i=1}^{N} fit(\boldsymbol{X}_i)} \tag{4}$$

where P_i represents the probability of the i^{th} honey source selected, and larger probability value means that the chance selected is more.

After onlooker bees reach their selected regions they then exploit them, and try to find a richer food sources like employed bees according to (3).

4) Scout bees searching stage. If the solution of an employed bee can not be improved through a predetermined number of trials controlled by parameter *limit*, and then its solution is abandoned and becomes a scout bee. Henceforth, the scout bee starts to search for new solution generated randomly according to (2).

5) To memorize the best solution achieved so far, if the terminated conditions are met, then export the best solution, and regenerate the two types of bee colony for all bees according to their current fitting values like step 1 and continue to perform iterations operation otherwise.

3. Simplex Method

NMSM is a local exploiting algorithm with higher efficiency[16]. The core idea of NASM is to make use of the concept of a simplex, that is, a special polytope constituted by D+1 vertices connecting one another in a D-dimension space[17]. For D-dimension-variable function minimization problems, NMSM adopts the means such as reflection, expansion, contraction, and compression operations to obtain a new point, and compares objective function values of the simplex (D+1) vertices, and replaces the objective function maximum value with a new point, such that the simplex can be updated through continuous iteration, and finally the problem optimal solution can be found [18]. Figure 1 shows a two-dimension space simplex map, and Figure 2 shows the NMSM flow chart.

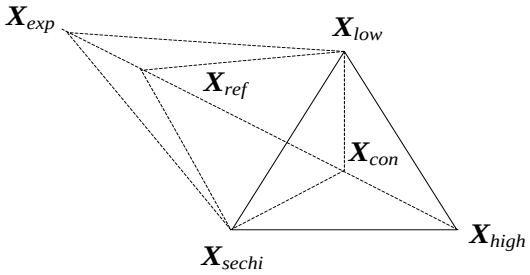


Figure 1. Two-Dimension Simplex Map

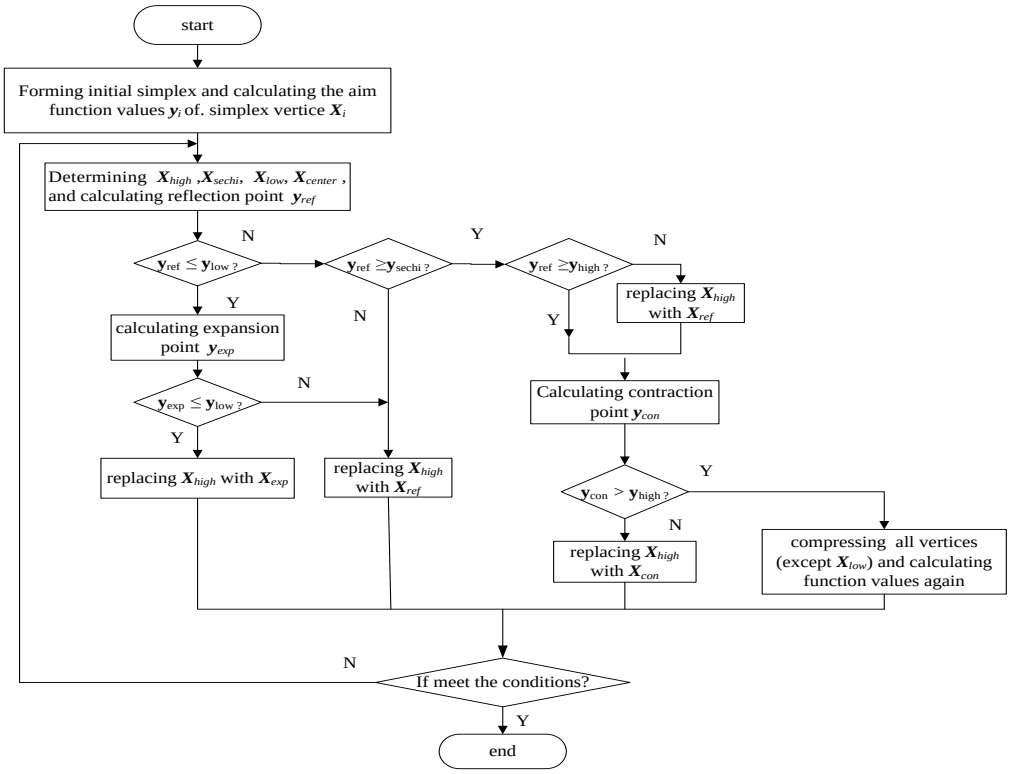


Figure 2. Function Optimum Flow Chart Based On NMSM

Function optimum steps based on NMSM are described as follows[19].

1) Initialization stage. For function minimization problem with *D*-dimension-variable, let $\boldsymbol{X}_0$, $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, …, $\boldsymbol{X}_D$ be *D*+1 points of *D*-dimension space, and they form the initial simplex. To then calculate the aim function value yi of each simplex vertice $\boldsymbol{X}_i$.

2) Reflection stage. In each iteration, to firstly determine the vertex $\boldsymbol{X}_{high}$ with maximum aim function value and the second maximum value vertex $\boldsymbol{X}_{sechi}$, and as well as the minimum value vertex $\boldsymbol{X}_{low}$. And then to calculate the center point $\boldsymbol{X}_{center}$, which the center coordinate of all vertices except $\boldsymbol{X}_{high}$. Finally, the reflection point $\boldsymbol{X}_{ref}$ is worked out by

$$\boldsymbol{X}_{ref} = \boldsymbol{X}_{center} + \alpha(\boldsymbol{X}_{center} - \boldsymbol{X}_{high}) \tag{5}$$

where α is the reflection coefficient larger than zero. If $y_{low} \le y_{ref} \le y_{sechi}$, then replace $\boldsymbol{X}_{high}$ with $\boldsymbol{X}_{ref}$ and enter into the next iteration, directly.

3) Expansion stage. If $y_{ref} \leqslant y_{low}$, then expand the reflection point and get the expansion point $\boldsymbol{X}_{exp}$ by

$$\boldsymbol{X}_{\exp} = \boldsymbol{X}_{center} + \gamma(\boldsymbol{X}_{ref} - \boldsymbol{X}_{center}) \tag{6}$$

where γ is the expansion coefficient larger than one. If $y_{exp} \leq y_{low}$, then replace $\boldsymbol{X}_{high}$ with $\boldsymbol{X}_{exp}$, and expansion failure ($y_{exp} > y_{low}$) otherwise, then replace $\boldsymbol{X}_{high}$ with $\boldsymbol{X}_{ref}$, go back to step 2) and enter into next iteration.

4) Contraction stage. If $y_{sechi} \leq y_{ref} \leq y_{high}$, then replace $\boldsymbol{X}_{high}$ with $\boldsymbol{X}_{ref}$, and shrink the reflection point by (7). If $y_{ref} > y_{high}$, then shrink the maximum point directly by (7).

$$\boldsymbol{X}_{con} = \boldsymbol{X}_{center} + \beta(\boldsymbol{X}_{high} - \boldsymbol{X}_{center}) \tag{7}$$

where β expresses the contraction coefficient($0<\beta<1$). If $y_{con} \leqslant y_{high}$, then replace $\boldsymbol{X}_{high}$ with $\boldsymbol{X}_{con}$, go back to step 2) and enter into next iteration.

5) Compression stage. In the step 4), if $y_{con} > y_{high}$, then contraction failure. To compress simplex all vertices (except $\boldsymbol{X}_{low}$) by (8).

$$\boldsymbol{X}_i = \boldsymbol{X}_{low} + \delta(\boldsymbol{X}_i - \boldsymbol{X}_{low}) \tag{8}$$

where δ is respectively the compression coefficient($0<\delta<1$). To calculate simplex vertices function values again, go back to step 2) and enter into next iteration.

4. NMSM-ABC

Like other swarm intelligent algorithm, ABC algorithm possesses global and local searching ability, simultaneously, but more focus on the global, and the local ability is poorer. When the fitness value of an employed bee closes to the local optimal point, many onlooker bees will reach that region and exploit it. If one of employed bees reaches the optimal point now, the algorithm will stop searching. However, this point is not the global optimal point. Under many cases, say, multi-modal function optimization, the optimal solution may be near a food source. At this very moment, the food source to be searched locally can increase the probability to find the global optimal point, clearly. In the view of fast optimization of ABC, the optimal point mentioned above is replaced by the position of a new bee. When the new bee is better than the optimal bee of swarm, the swarm will lose the opportunity to explore potential areas.

Figure 3 shows a maximum function optimization problem. A bee($\mathbf{X}_1$) is influenced by current optimal bee information and may be attracted to the area, but ignores the location of $\mathbf{p}_1$ more near the highest peak. If one local exploration algorithm is introduced to implement deeply searching around the p1, and then the probability to find the global optimal point will be improved. In order to overcome the shortcomings, based on NMSM this paper proposes an improved ABC algorithm. This algorithm periodically applies reflection, contraction, expansion and so on NMSM law to update simplex vertices for bee locations obtained by ABC algorithm, and thus the search scope is narrowed so that the depth development is achieved, and so the local search ability is improved and the probability to search optimal solution near potential areas increases. In another hand, NMSM also can get away from local minimum using ABC global ability. In the end, the algorithm proposed realizes the cooperative co-evolution searching of the ABC and NMSM.

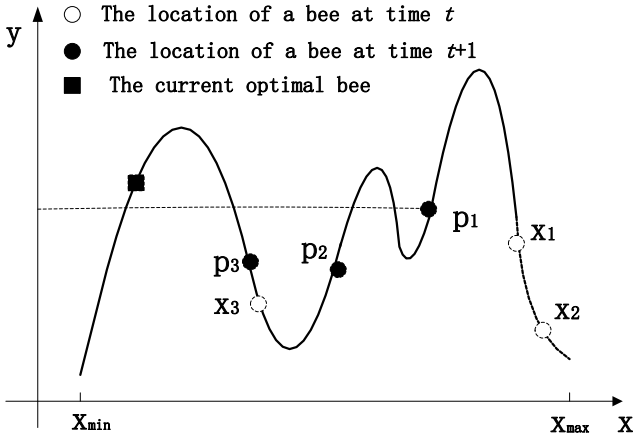


Figure 3. Effects of the Current Optimal Bee on Algorithm Performance

Figure 4 shows the NMSM-ABC cooperative co-evolution model. In this algorithm, the initialized swarm is divided into two subgroups being equal in number: one group implements ABC algorithm and denoted as A-colony, and another one performs NMSM algorithm and is B-colony. When A-colony meets the migration condition, the optimal food source location information migrates to NMSM from ABC to get away from local minimum. Meanwhile, the best individual vertex migrates from NMSM to A-colony for deeper exploration. The migration between the two continues all along until the algorithm converges.

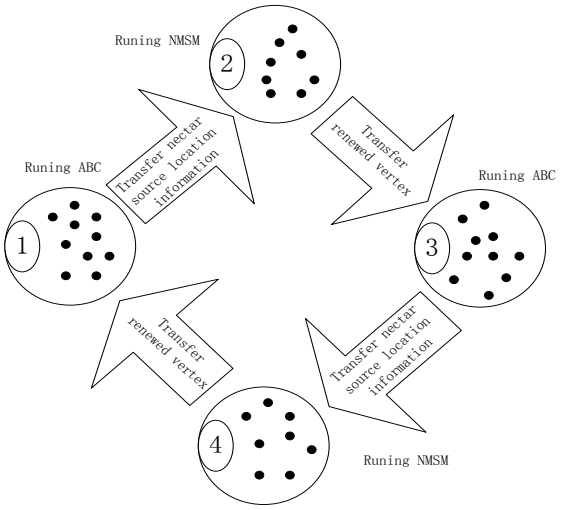


Figure 4. NMSM-ABC Algorithm Model

Figure 5. Flow Chart of NMSM-ABC

The algorithm consists of three parts: ABC algorithm, and NMSM algorithm, and as well as individual migration. Individual migration plays an important role in terms of the ability to balance global search and local exploitation. Figure 5 shows the NMSM-ABC flow program.

NMSM-ABC algorithm steps are described as follows.

1) To initialize two colony of A and B mentioned as above, and make them possesses to same colony size of N separately, and set the maximum iterations times, and the value of parameter *limit*, and simplex operators such as reflection coefficient α, contraction coefficient β, expansion coefficient γ, reduction coefficient δ, and contraction times, and as well as individual migration time interval Δ, and the dimension D.

2) To calculate the aim function value and fitness value of the individual of two populations.

3) To loop the following steps, until the termination conditions is met.

(1) To run ABC and NMSM at the same time.

(2) To choose the optimal point of ABC or the best simplex vertex of NMSM.

(3) To migrate the individual selected by unidirectional ring way.

(4) To replace the selected vertex in NMSM or the optimal point randomly in ABC by the migration one.

5. Examples

In this paper, we design two examples to verify the performance of NMSM-ABC algorithm as follows.

***Example1*.** There are six classical testing functions being of single modal and multimodal characteristics to be selected to test the performance of the improved ABC, and the testing results are compared with other ABC algorithm such as standard ABC and Self-Adaptive Mutation Article Bee Colony (SAM-ABC)[20].

Table 1 shows the selected functions, and as well as their corresponding searching spaces and the global optimum values, wherein f_1 is a high-dimensional single-mode state function, and f_2, f_3, f_4 are high-dimensional multimodal functions having many local minimum points, and f_5 is a smooth high-dimensional parabolic function with its global optimal point being difficult to find, and f_6 is morbid high-dimensional function which is hard to be minimized.

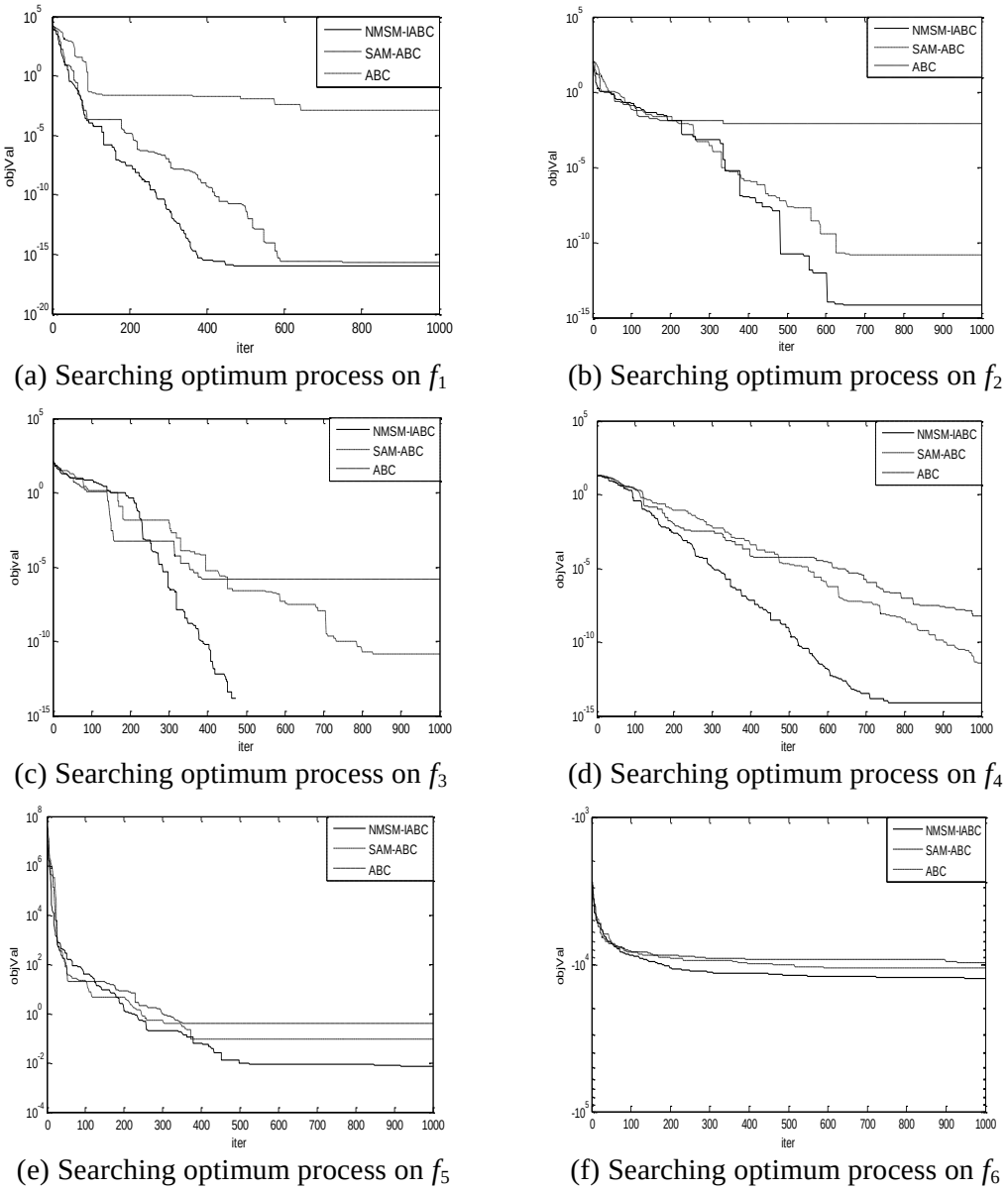
Table1. Classical Functions to Be Tested

Function	Interval	Dimension	Global Optimum
$f_1 = \sum_{i=1}^{D} x_i^2$	[-100,100]	50	0
$f_2 = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600,600]	60	0
$f_3 = \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i) + 10 \right]$	[-5.12,5.12]	60	0
$f_4 = -20e^{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}} - e^{\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)} + 22.71282$	[-32,32]	60	0
$f_5 = \sum_{i=1}^{D-1} \left[100\left(x_i^2 - x_{i+1}\right)^2 + \left(x_i - 1\right)^2 \right]$	[-30,30]	60	0
$f_6 = \sum_{i=1}^{D} \left(-x_i \sin\sqrt{\lvert x_i \rvert} \right)$	[-500,500]	30	-12569.49

NMSM-ABC parameters setting are as follows.

Let the initialized colony size *N* be 100, and *limit* be 100, and simplex operators α be 1, and β be 0.5, and γ be 2, and δ be 0.5, and contraction times *t* be10, and individual migration interval Δ be 10, and the maximum iteration times equal 1000.

Figure 6 shows the convergence performances of three types of algorithms. In each sub-Figure, the real line represents the proposed algorithm in this paper, and dotted line is standard ABC, and dashed line means SAM-ABC, wherein horizontal means iteration time, and expressed by symbol "iter", and vertical coordination is the optimal solution and presented by "objval". Table 2 shows the optimized results of six-test functions, wherein the best results have been marked in bold. Seen from Figure 6 and Table 2, NMSM-ABC gets the optimal solution with faster evolution speed and higher precision, and possesses better convergence performance than the standard ABC and SAM-ABC. For the high-dimensional single-mode function f_1 in sub-Figure (a), ABC does not conduct detailed search in the feasible region, and SAM-ABC can get the global optimal solution by adaptive adjustment, and NMSM-ABC can do a detailed search by NMSM around the feasible region so as to get more accurate solution with faster convergence speed. For high-dimensional multimodal functions f_2, f_3, f_4 in sub-Figure (b) to (d), SAM-ABC does not make full use of ABC global search ability such that the precision is not high, and ABC is easy to fall into local optimum, but NMSM-ABC makes full use of local development and global search ability such that its solution precision and convergence speed is improved quite dramatically. For f_5 in sub-Figure (e), the optimal solution is in a smooth narrow parabolic valley, where SAM-ABC improves the solution precision in later period by increasing adaptive coefficient, and NMSM-ABC gets better solution by exploring local region. For the morbid high-dimensional function f_6 in sub-Figure (f), NMSM-ABC can get the ideal optimal solution with higher precision, and therefore, NMSM-ABC is more suitable to solve high-dimensional complex optimization problems.



(a) Searching optimum process on f_1 (b) Searching optimum process on f_2

(c) Searching optimum process on f_3 (d) Searching optimum process on f_4

(e) Searching optimum process on f_5 (f) Searching optimum process on f_6

Figure 6. Searching Optimum Processes of Diverse Functions with Different Algorithms

Table 2. The Optimal Solutions Obtained Under Diverse Algorithms

Algorithm	f_1	f_2	f_3	f_4	f_5	f_6
NMSM-IABC	**9.33E-18**	**7.33E-15**	**0**	**7.99E-15**	**6.76E-3**	**12568.48**
SAM-ABC	1.06E-16	2.38E-12	8.26E-11	4.13E-12	2.35E-2	11350.26
ABC	6.98E-4	7.84E-2	9.68E-6	8.56E-7	0.13	-9986.39

Table 3 shows average running time, and average iteration time of each step, and total average iterations times of the three algorithms. For function f_1, f_3, f_4, f_6, NMSM-ABC average running time and iterations number are less than ones of SAM-ABC and ABC. Especially, in terms of function f_4, SAM-ABC and ABC have no convergence as being in 1000 times. For function f_2 and f_5, SAM-ABC average iteration time is less than NMSM-

ABC and ABC, but the average running time larger than one of NMSM-ABC. For function f_2 to f_6, NMSM-ABC each average iteration time is least. For function f_1, NMSM-ABC one step average iteration time is slightly more than SAM-ABC due to the amount of calculation of the simplex algorithm. However, if considering the optimization precision in Table 2, NMSM-ABC is better than other algorithms in composite index.

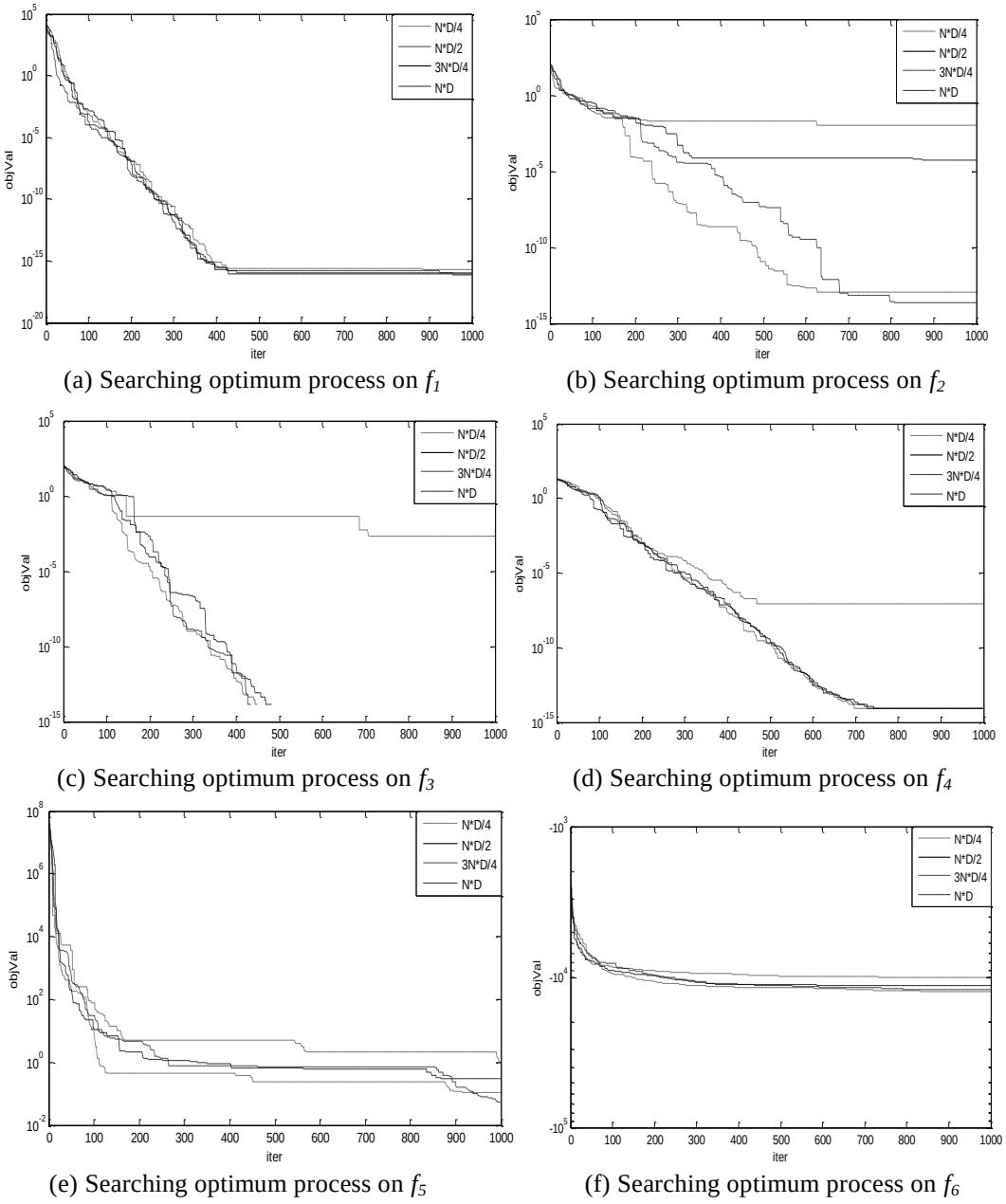
Table 3. Average Running Time, Average Each-Step Iteration Time and Average Number of Iterations under Diverse

Algorithm	Times and time	f_1	f_2	f_3	f_4	f_5	f_6
NMSM-IABC	Running time(s)	7.3	7.2	12.5	13.3	15.1	8.2
	Each iteration time (s)	1.72E-2	1.16E-2	2.70E-2	1.74E-2	2.70E-2	2.64E-2
	Average number of times	423.4	620	462.6	762.5	560.3	310.5
SAM-ABC	Running time(s)	10.1	7.9	25.6	18.5	16.2	13.6
	Each iteration time (s)	1.68E-2	1.29E-2	3.10E-2	1.85E-2	4.06E-2	2.69E-2
	Average number of times	602.3	613	820.5	1000	398.6	506
ABC	Running time(s)	53	75.6	81	97.6	93.3	62.8
	Each iteration time (s)	7.34E-2	2.44E-1	1.88E-1	9.76E-2	2.67E-1	6.96E-2
	Average number of times	721.6	310	430.2	1000	350	902

***Example2*.** Sensitivity analysis of key parameter *limit*.

In NMSM-ABC algorithm, the parameter *limit* has a great influence on the convergence speed and global optimization abilities, which is applied to generate and control the number of the scout bees. Figure 7 shows the experiment results as *limit* takes different values.

From Figure 7, we can see that the algorithm indicates diverse convergence performance with the change of parameter *limit*. As *limit* takes $N\times D/2$, the algorithm possesses the best stability superior to all other value-taking. As the *limit* equals $N\times D/4$, the algorithm stability is the worst, especially in terms of function f_2, f_3, f_4, but as taking $3N\times D/4$, the stability of the algorithm is better one of the former. When the *limit* equals $N\times D$, the algorithm falls into local optimum in function f_2. Table 4 shows the detailed results, and the best results have been marked in bold. Clearly, as *limit* takes $N\times D/2$, the algorithm possesses the best property, this is because the employed bees($N/2$) search dimension by dimension such that the maximum searching times does not exceeding $N\times D/2$. If the maximum searching times is less than $N\times D/2$, and then it is possible that there are some directions where the optimal value exists but not to be exploited such that it is ignored. And conversely, if the maximum searching times is larger than $N\times D/2$, and possibly, there exist some directions where the bees perform searching again such that the equivalent possibility condition of the searching for the employed bees in each dimension is destroyed, and the effect to do so is similar to the former, that is, to reduce the unexplored areas. If the maximum searching times is equal to $N\times D$, and then although the searching times increases, it is possible that the obtained areas in the second circle are no better than ones in the first, and so the searching efficiency is reduced. To sum up, as the value of *limit* takes $N\times D/2$, the NMSM-ABC shows the best properties.



(a) Searching optimum process on f_1 (b) Searching optimum process on f_2

(c) Searching optimum process on f_3 (d) Searching optimum process on f_4

(e) Searching optimum process on f_5 (f) Searching optimum process on f_6

Figure 7. Influence of Parameter *Limit* on NMSM-ABC Algorithm

Table 4. Sensitivity Analysis on Parameter *Limit* Under Diverse Functions

Parameter *limit*	f_1	f_2	f_3	f_4	f_5	f_6
$N\times D/4$	8.06E-18	4.97E-3	2.24E-3	8.36E-8	0.846	-12003.76
$N\times D/2$	**9.27E-18**	**7.33E-15**	**0**	**7.99E-15**	**0.036**	**-12568.48**
$3N\times D/4$	**9.27E-18**	8.29E-14	**0**	**7.99E-15**	0.053	**-12568.48**
$N\times D$	8.45E-18	9.38E-5	**0**	**7.99E-15**	0.303	-12526.83

6. Conclusions

This paper proposes a cooperative co-evolution algorithm called as NMSM-ABC based on ABC and NMSM algorithms. The presented algorithm NMSM-ABC possesses prominent properties both in global and local searching abilities since it effectively combines ABC and NMSM together, that is, the algorithm periodically gets the best individual vertex from NMSM operator and migrates to ABC, or obtains the optimal nectar source information from ABC and migrates to NMSM during iterations, such that ABC can improve its local exploiting capability applying NMSM, and NMSM can get away from local minimum by ABC. Simulation results show that the proposed algorithm possesses better convergence performance and computing accuracy than ones of SAM-ABC and ABC. The presented algorithm can be applied to solve the all kinds of function optimization problems and engineering application puzzles, and possesses excellent practical applications prosperity.

References

[1] D. Karaboga, "An idea based on bee swarm for numerical optimization", Erciyes University, Engineering Faculty, Computer Engineering Department, **(2005)**.
[2] D. Karaboga and B. Akay, "A comparative study of artificial bee colony(ABC) algorithm", Applied Mathematics Computation, vol. 8, no. 1, **(2009)**, pp. 108-132.
[3] F. Jin, "Path Planning of Free-Flying Space Robot Based on Artificial Bee Colony Algorithm", Proceedings of 2012 2nd International Conference on Computer Science and Network Technology, December 29-31; Changchun, China, **(2012)**.
[4] D. Aydına, "Composite artificial bee colony algorithms: From component-based analysis to high-performing algorithms", Applied Soft Computing, vol. 32, **(2015)**, pp. 266-285.
[5] A. Fathy, "Reliable and efficient approach for mitigating the shading effect on photovoltaic module based on Modified Artificial Bee Colony algorithm", Renewable Energy, vol. 81, **(2015)**, pp. 78-88.
[6] S.K. Agrawal, "Artificial bee colony algorithm to design two-channel quadrature mirror filter banks", Swarm and Evolutionary Computation, vol. 21, **(2015)**, pp. 24-31.
[7] A. Esmailzadeh and S. Rahnamayan, "Enhanced Differential Evolution Using Center-Based Sampling", Proceedings of 2011 IEEE Congress of Evolutionary Computation, Neworleans, Louisiana, USA, **(2011)**.
[8] J. Zhang and Z. Zhang, "Improved particle swarm optimization algorithm and its application for complex function", Advances in intelligent and soft computing, vol. 143, **(2012)**, pp. 683-690.
[9] Y. Xiang, "An elitism based multi-objective artificial bee colony algorithm", European Journal of Operational Research, vol. 245, **(2015)**, pp. 168-193.
[10] Y. Huo, "Elite-guided multi-objective artificial bee colony algorithm", Applied Soft Computing, vol. 32, **(2015)**, pp. 199-210.
[11] D. Karaboga, "A Quick Artificial Bee Colony -QABC- Algorithm for Optimization problems", Proceeding of Innovations in Intelligent Systems and Applications(INISTA), Istanbul, Turkey, **(2012)**.
[12] G. Li, "Development and investigation of efficient artificial bee colony algorithm for numerical function optimization", Applied Soft Computing, vol. 12, no. 1, **(2011)**, pp. 320-332.
[13] T.K. Sharma and M. Pant, "Differential operators embedded artificial bee colony algorithm", Appl. Evol.Comput., vol.2, no.3, **(2011)**, pp.1-14.
[14] W. Gao, "Bare bones artificial bee colony algorithm with parameter adaptation and fitness-based neighborhood", Information Sciences, vol. 316, **(2015)**, pp. 180-200.
[15] M. Servet Kiran, "Artificial bee colony algorithm with variable search strategy for continuous optimization", Information Sciences, vol. 300, **(2015)**, pp. 140-157.
[16] P. C. Wang, "Parameter sensitivity study of the Nelder–Mead Simplex Method", Advances in Engineering Software, vol. 42, **(2011)**, pp. 529-533.
[17] D. Kamiyama, "Down-hill Simplex Method Based Differential Evolution", Proceeings of SICE Annual Conference 2010, Taipei, Taiwan, **(2010)**.
[18] K. H. Chang, "Stochastic Nelder–Mead simplex method – A new globally convergent direct search method for simulation optimization", European Journal of Operational Research, vol. 220, **(2012)**, pp. 684-694.
[19] M.T. Vakil Baghmisheh, "A hybrid particle swarm–Nelder–Mead optimization method for crack detection in cantilever beams", Applied Soft Computing, vol. 12, **(2012)**, pp. 2217-2226.
[20] M. Sharfiul Alam, "Artificial Bee Colony Algorithm With Self-Adaptive Mutation: A Novel Approach for Numeric Optimization", Proceedings of TENCON 2011-2011 IEEE Region 10 Conference, November 21-24; Bali, Indonesia, **(2011)**.

Authors

Hongsheng Su, he obtained his Master in Traffic Information Engineering and Control, Lanzhou Jiaotong University in 2001. He acquired his PhD in Power Systems and Its Automation, Southwest Jiaotong University. Now he is serving as a full-time professor at school of Automation and Electrical Engineering, Lanzhou Jiaotong University. His research interest includes System Security and Reliability, Intelligent Control, Power Systems and Its Automation, and *etc*.

Kaile Yin, he is now a master graduate student, Lanzhou Jiaotong University. His research interest includes artificial bee colony algorithm and Power System stability.

Zihan Wu, he is now a master graduate student, North China Electric Power University. His research interest includes power generation ability of wind turbine and wind turbine fault detection.