10.1109/CEC.2016.7743783

A New Metaheuristic Algorithm for Real-Parameter Optimization: Natural Aggregation Algorithm

Fengji Luo, Junhua Zhao, and Zhao Yang Dong

[1]*Abstract*—**This paper proposes a new evolutionary algorithm (EA), which is called the natural aggregation algorithm (NAA). NAA is inspired by the collective decision making intelligence of the group-living animals. Distinguished from other EAs, NAA distributes individuals to several sub-populations (called 'shelters'), and uses a stochastic migration model to dynamically mitigate the individuals among the shelters. The *inter-individual attraction effect* and *crowding effect* are considered in the migration model to balance the exploration and exploitation. In each generation, both of the located search and generalized search are performed simultaneously, and the distributions of the individuals are self-adaptively updated. 7 benchmark functions with different dimensionality settings are used to validate the efficiency of NAA, and the results clearly show that NAA has strong performance for solving the real-parameter optimization problems.**

Index Terms—**Natural aggregation algorithm, evolutionary algorithm, collective decision making, heuristic optimization, computational intelligence**

I. Introduction

OPTIMIZATION problems rise in various fields of people's real life. Traditionally deterministic optimization techniques, such as the linear programming (LP), dynamic programming (DP), nonlinear programming (NLP), etc., have been widely applied. All these methods have certain requirements on the forms of the optimization models, which would not always be satisfied in real applications. In order to overcome the limitations of mathematical programming methods, another branch of the optimization is developed, which is the heuristic optimization. Basically, heuristic optimization techniques perform stochastic search in the problem spaces. Comparing with the mathematical programming methods, heuristic optimization techniques are robust and global, and often have no special constraints on the form of the objective functions (e.g., differentiable requirement). Over the last many years, several heuristic optimization algorithms have been proposed, such as genetic algorithm (GA) [1], evolutionary programming (EP) [2], simulated annealing (SA) [3], particle swarm optimization (PSO) [4], differential evolutionary algorithm (DE) [5], quantum-inspired evolutionary algorithm (QEA) [6], etc.

This paper proposes a new metaheuristic optimization algorithm with the name of natural aggregation optimization (NAA). NAA is inspired by the self-organization behaviors of the group-living animals in resource sharing & competition process. Biologists have observed that in a limited resource environment, a small number of individuals who find high quality resources (food, shelter, etc.) will attract other individuals to aggregate as different groups to possess the resources [7], [8]. Such aggregation can reduce different physical stresses, such as predation or increase the food intake, etc. For group-living animals, the decision making of group formation depends on both social interactions and assessment of the environmental opportunities. To deeply reveal this group formation mechanism, biologists specifically established experiments to study the cockroach's aggregation behaviors under multiple shelters located at a bounded container [9]. They found that the decision making of an individual cockroach on entering or leaving a shelter is influenced by both of the shelter quality and crowding degree of that shelter. After the evolution of exploration and interaction processes, the distribution of cockroaches among the shelters will automatically reach a dynamical stable state.

The collective decision making intelligence of group-living animals is mapped to the design philosophy of NAA. One of the most important considerations of EAs is to balance the exploration and exploitation of the searching process. In the group-living animals' world, aggregation can make them move towards the high quality resource (exploration), but the over-crowded aggregation will reduce the exploitation ability of the swarm. The evolution mechanism of NAA naturally inherits from such biological aggregation phenomenon. In NAA, a population of individuals is distributed to multiple sub-populations, which are called 'shelters'. A stochastic migration model, whose prototype is the cockroach aggregation model [9], is designed. In each generation, each individual uses the stochastic migration model to decide whether enter or leave a shelter temporarily. Then, the individuals in the same shelter move towards the site of that shelter (located search), and the individuals which are not belonged to any shelter do the free search in the unexplored space (generalized search). The multiple sub-populations coordinately evolve and the algorithm finally converges to the global/near-global optimum.

This paper will be organized as below. Section II introduces

[1] Fengji Luo is with the School of Civil Engineering, The University of Sydney, Australia (email: fengji.luo@sydney.edu.au)

Junhua Zhao is with the Chinese University of Hong Kong (Shen Zhen), China (email: zhaojunhua@cuhk.edu.cn)

Zhao Yang Dong is with the School of Electrical and Information Engineering, The University of Sydney, Australia (email: zydong@ieee.org)

the cockroach aggregation experiment and its mathematical aggregation models; Section III presents the procedures of NAA; Section IV validates the performance of NAA on 7 benchmark functions, compared with some other EAs; Section V gives some control parameter selection guidelines of NAA; Section VI explains the naming philosophy of the proposed algorithm; finally, conclusions and future works are drawn in Section VII.

II. Introduction of the Cockroach Aggregation Behavior

One nature of the cockroaches is that they prefer to hide in dark shelters. If a swarm of cockroaches is released in an open space, they will quickly find and aggregate in the shelters. This aggregation process involves the cooperation & competition games of the swarm members. On one hand, the cockroaches cooperate with each other to find and share the shelters; one the other hand, they compete with each other due to the limited capacity of the shelters.

The cockroach's aggregation process is completely distributed and autonomous. There is no centralized controller to control the behavior of the swarm. Basically, at the beginning every cockroach randomly explores the space to find the shelters. Then, the cockroaches interact with each other to exchange information and independently make decisions on entering or leaving a shelter based on their local knowledge.

In 2005, J. Ame et al. [10] organized a series of experiments to detailed reveal the cockroach's aggregation mechanism. They setup various numbers of shelters in a container, released the cockroaches into the container, and recorded their behaviors. Finally, they established a stochastic model whose prediction results can well match the experimental results. They found that for each cockroach in shelter *s*, it had probability Q_s to leave the shelter to do explore,

$$Q_s = \frac{\theta}{1+\rho\left(\frac{x_s}{S}\right)^n} \tag{1}$$

where parameters θ depends on the shelter quality; S is the shelter capacity, in terms of the number of cockroaches that can rest in the shelter; parameter ρ is the reference surface ratio for estimating the shelter capacity; parameter n controls the social interaction response rate; x_s is the number of cockroaches present in shelter *s*.

Eq. (1) represents the *inter-individual attraction effect* of the cockroaches. That is, Q_s increases with the increase of x_s. The biologists also found the probability for an exploring cockroach to join in a shelter *s* (R_s) decreases with linear *crowding effect*,

$$R_s = \mu\left(1-\frac{x_s}{S}\right) \tag{2}$$

where parameter μ is the maximal kinetic constant for entering a shelter.

Models (1) and (2) show that in the cockroach's aggregation process, no long-range interactions among cockroaches (such as the chemical marking or memory effect) occur. Every cockroach dynamically and independently makes its choice, under the influences of the i*nter-individual attraction effect* and *crowding effect.* Using the ideal free-distribution theory [10], biologists also mathematically proved that the behaviors under models (1) and (2) are optimal for each individual cockroach, and can best balance the cooperation and competition of the swarm.

III. Natural Aggregation Algorithm

Like other EAs, NAA maintains a population of individuals, where each individual represents a potential solution for the given optimization problem. Two positive integers, N^S and Cp^S , are pre-specified to represent the number of shelters and the number of individuals each shelter can contain, respectively. Note that in NAA, the shelter capacity (Cp^S) is assumed to be same for all shelters. At the beginning of the algorithm, the individuals randomly search in the problem space, and the positions of N^S individuals which have best fitness values are selected as the shelter leaders, and are associated with the shelter indexes 1, 2,…, N^S . The *shelter site* of each shelter is also set to be the position of corresponding shelter leader. The rest individuals are then distributed into the shelters as the *shelter followers*, by being associated with the corresponding shelter index. This actually divides the whole population into multiple sub-populations. After that, the algorithm starts the iteratively search process for the global optimum in the problem space. During the evolution, each individual could switch between 2 statuses:

Exploit Individual: If an individual is belonged to a certain shelter, it is an *exploit individual*;

Explore Individual: If an individual does not belong to any shelter, it is an *explore individual*.

In each generation of the search, for each exploit individual belonged to a certain shelter *s*, it calculated its probability Q_s of leaving the current shelter, and decides whether leave or not. Once it decides to leave, it changes to be the explore individual. Also, for each explore individual, it randomly chooses a shelter *s* and calculates the probability R_s of entering it. Once it decides to enter that shelter, it will become the exploit individual by being associated with the shelter index *s*. After making the decision, each individual of the population performs following search strategies:

Located Search: For each exploit individual, it moves towards its shelter site;

Generalized Search: For each explore individual, it randomly chooses other individuals over the population and moves towards their adjacent areas.

At the end of each generation, the fitness values of each individual are evaluated, and the shelter leaders are updated. This iteration process continues until the preset termination criterions are met. The conceptual diagram of NAA is shown in Fig. 1.

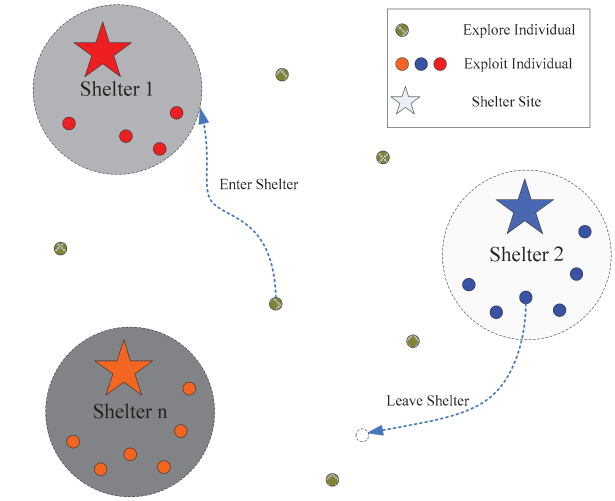

Fig. 1. Conceptual diagram of NAA

It can be seen that during the evolution process, each individual can dynamically migrate among different shelters and the unexplored spaces. The exploration and exploitation of the whole population are balanced by the stochastic migration model consisting of the calculations of Q_s and R_s.

Next, we present the detailed procedures of NAA as below.

A. Population Initialization

Given a minimization problem over D-dimensional continuous space, each individual $\boldsymbol{c}_i$ in the population with the size of N_{pop} is coded as a D-dimensional vector,

$$\boldsymbol{c}_i = [c_{i1}, c_{i2}, \ldots, c_{iD}] \tag{3}$$

For each individual, its initial value is randomly generated within the initial bounds $\boldsymbol{B}_l = [B_{l1}, B_{l2}, \ldots, B_{lD}]$ and $\boldsymbol{B}_u = [B_{u1}, B_{u2}, \ldots, B_{uD}]$. That is,

$$c_{ij} \approx U(B_{lj}, B_{uj}) \tag{4}$$

where j is the dimension index; $U(\bullet)$ denotes the uniform distribution function. Each individual is also associated with a fitness value which can be evaluated as,

$$fit_i = \text{fitness}(\mathbf{c}_i) \tag{5}$$

where $\text{fitness}(\bullet)$ denotes the fitness evaluation function.

B. Shelter Initialization

After generating the populations, the fitness value of each individual is evaluated. The N^S individuals with minimum fitness values are selected as the shelter leaders, and are associated with shelter indexes 1,…, N^S. And the positions of shelter leaders are copied to be the shelter sites (denoted as $\textbf{site}_s^S$. The fitness value of the individual with (N^S+1) th minimum fitness value is also stored (denoted as fit_{base}), which will be used in the shelter quality normalization process described later. The rest individuals are sequentially distributed to each shelter based on Cp^S, by marking them with corresponding shelter indexes. The pseudo codes of this distribution process are shown in Table I.

TABLE I
PSEUDO CODES OF THE SHELTER INITIALIZATION

begin:
/* set the initial shelter leaders */
(1) **for** each $\boldsymbol{c}_i$, $i = 1, 2, \ldots N_{pop}$ in sequential order,
(2) evaluate $fit_i = \text{fitness}(\boldsymbol{c}_i)$;
(3) **end for**
(3) sort the individuals with the ascendant order of fit_i
(4) **for** each of the first N^S sorted individuals, $\boldsymbol{c}_s^l$, in sequential order
(4) associate $\boldsymbol{c}_s^l$, with the shelter index s ($s = 1, 2, \ldots N^S$);
(5) copy $\boldsymbol{c}_s^l$ to $\textbf{site}_s^S$ /* set the shelter sites */
(5) **end for**

/* set the initial shelter followers */
(6) set variable pt=0;
(7) **for** i=1: N^S
(8) set variable $count$ = 1;
(8) **for** k=pt: N_{pop}
(9) if $\boldsymbol{c}_i$ has not been assigned the shelter index
(10) assign $\boldsymbol{c}_i$ with the shelter index i;
(11) set $count$=$count$+1;
(12) **if** count> Cp^S
(13) pt=k;
(14) break;
(15) **end if**
(16) **end if**
(17) **end for**
(18) **end for**
end

After initializing the shelters, NAA starts the iteration process to evolve the sub-populations.

C. Stochastic Migration Model

In each generation, each individual of the population makes decision about leaving or entering a certain shelter.

1) Decision Making of The Exploit Individual

If an individual $\boldsymbol{c}_i$ is an exploit individual belonged to shelter s, it firstly evaluate its probability of leaving current shelter based on model (6),

$$Q_s = \frac{\overline{\theta_s}}{1 + \left(\dfrac{x_s}{Cp^S}\right)^2} \tag{6}$$

In model (6), $\overline{\theta_s}$ is the normalized quality of shelter s. The smaller value of $\overline{\theta_s}$ indicates the better shelter quality; x_s is the number of individuals currently in s. $\overline{\theta_s}$ is calculated by Eqs. (7) and (8),

$$\theta_s = \text{fitness}(\boldsymbol{site}_s^S) - fit_{base} \tag{7}$$

$$\overline{\theta_s} = 1 - \frac{\theta_s}{\sum_{j=1}^{N^S} \theta_j} \tag{8}$$

After calculating Q_s, c_i decides to leave s or not,

$$leave = \begin{cases} 1 & \text{if } rand() \le Q_s \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $rand()$ is a generator which generates a random number uniformly distributed within [0, 1]. Once $\mathbf{c}_i$ decides to leave its current shelter, it becomes an explore individual and its associated shelter index is set to be -1.

2) Decision Making of Explore Individual

If an individual $\mathbf{c}_i$ is an explore individual, it firstly randomly select a shelter s,

$$s = rint(1, N^S) \tag{10}$$

where $rint(a,b)$ is a generator which generates a random integer uniformly distributed between the two integers a and b ($b>a$). Then, $\boldsymbol{c}_i$ evaluates the probability of entering s by Eq. (11), and makes decision based on Eq. (12).

$$R_s = \left(1-\overline{\theta_s}\right)\left(1-\frac{x_s}{Cp^S}\right) \tag{11}$$

$$enter = \begin{cases} 1 & if\ rand() \le R_s \\ 0 & otherwise \end{cases} \tag{12}$$

Once $\boldsymbol{c}_i$ decides to enter shelter s, its associated shelter index is set to be s. Comparing Eqs. (6) and (11) with the original biological equations (1) and (2), it can be seen that in NAA, the crowding response rate n is fixed to be 2, and the constant parameter μ is replaced by $(1-\overline{\theta_s})$, so as to take the shelter quality into account.

After making the migration decisions, each individual of the population will take the search action. Specifically, the exploit individuals will do the located search, and the explore individuals will do the generalized search.

D. Located Search

In each generation, for the exploit individuals $\boldsymbol{c}_i$, the shelter leaders and followers use different located search strategies. If $\boldsymbol{c}_i$ is a shelter leader, it tries to search its neighbored area $[-|\mathbf{c}_i|\delta, |\mathbf{c}_i|\delta]$. To do this, a mutation M_i is firstly generated,

$$M_i = [m_{1,i}, m_{2,i}, ..., m_{D,i}] \tag{13}$$

$$m_{j,i} = -|c_{j,i}|\delta + rand(j)\cdot 2\cdot |c_{j,i}|\delta \quad j=1,2,...,D \tag{14}$$

Then, the crossover operation is performed to generate a new shelter leader candidate,

$$\boldsymbol{c}_i^{new} = \begin{cases} m_{j,i} & if\ rand(j) \le Cr_{local} \text{ or } j == rint_i(1,D) \\ \boldsymbol{c}_i & otherwise \end{cases} \tag{15}$$

where Cr_{local} is the located search crossover factor; δ is the scaling factor; $rand(j)$ is the jth generated uniform random number; $rint_i(1,D)$ is a randomly selected number between [1, D] which is generated in advance for $\boldsymbol{c}_i$.

If $\boldsymbol{c}_i$ is a shelter follower of shelter s, it will move towards the shelter site. Firstly, a mutation is generated,

$$M_i = [m_{1,i}, m_{2,i}, ..., m_{D,i}] \tag{16}$$

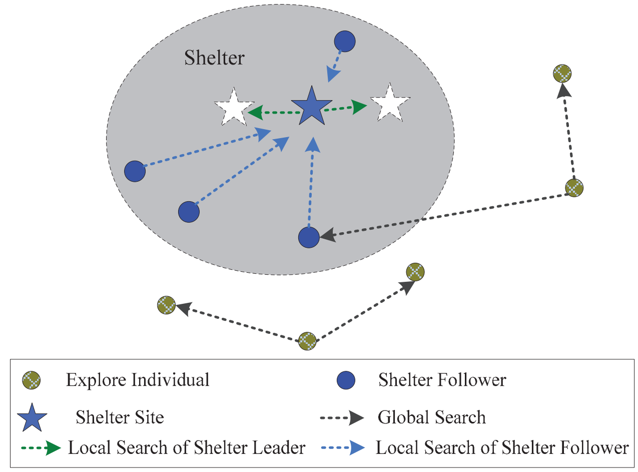

Fig. 2. Search strategies of NAA

$$m_{j,i} = \boldsymbol{c}_{j,i} + 2\cdot \mathrm{rand}()\cdot\left(\mathbf{site}^S_{j,s} - \boldsymbol{c}_{j,i}\right) \quad j=1,2,...,D \tag{17}$$

Then, the crossover operation is performed to generate a new shelter follower candidate,

$$\boldsymbol{c}_i^{new} = \begin{cases} m_{j,i} & if\ rand(j) \le Cr_{local} \text{ or } j == rint_i(1,D) \\ \boldsymbol{c}_i & otherwise \end{cases} \tag{18}$$

E. Generalized Search

If an individual $\boldsymbol{c}_i$ is an explore individual, firstly it randomly selects two different individuals $\boldsymbol{c}_{r1}$ and $\boldsymbol{c}_{r2}$ to generate a mutant,

$$M_i = [m_{1,i}, m_{2,i}, ..., m_{D,i}] \tag{19}$$

$$\begin{aligned} m_{1,i} = {} & c_{j,i} + \alpha\cdot rand_1(j) + (c_{j,r1} - c_{j,i}) + \\ & \alpha\cdot rand_2(j)(c_{j,r2} - c_{j,i}) \end{aligned} \tag{20}$$

where $rand_1(j)$ and $rand_2(j)$ are two uniform random numbers of the jth dimension; α is the generalized movement amplification. Then, the crossover operation is performed,

$$\boldsymbol{c}_i^{new} = \begin{cases} m_{j,i} & if\ rand(j) \le Cr_{global} \text{ or } j == rint_i(1,D) \\ c_{j,i} & otherwise \end{cases} \tag{21}$$

F. Movement Trial

For each individual $\boldsymbol{c}_i$ in the population, after generating the new candidate $\boldsymbol{c}_i^{new}$, it will make a movement trial based on its local knowledge. The fitness value of the new position $\boldsymbol{c}_i^{new}$ will be compared with $\boldsymbol{c}_i$. If the fitness value of $\boldsymbol{c}_i^{new}$ is larger than $\boldsymbol{c}_i$, then $\boldsymbol{c}_i$ will move to the new positions of $\boldsymbol{c}_i^{new}$; otherwise, $\boldsymbol{c}_i$ will be retained.

G. Shelter Update

At the end of each generation, after doing the searches, all individuals are sorted by ascendant order of the fitness values. The N^S individuals with minimum fitness values are updated as the new shelter leaders, and are assigned the shelter indexes

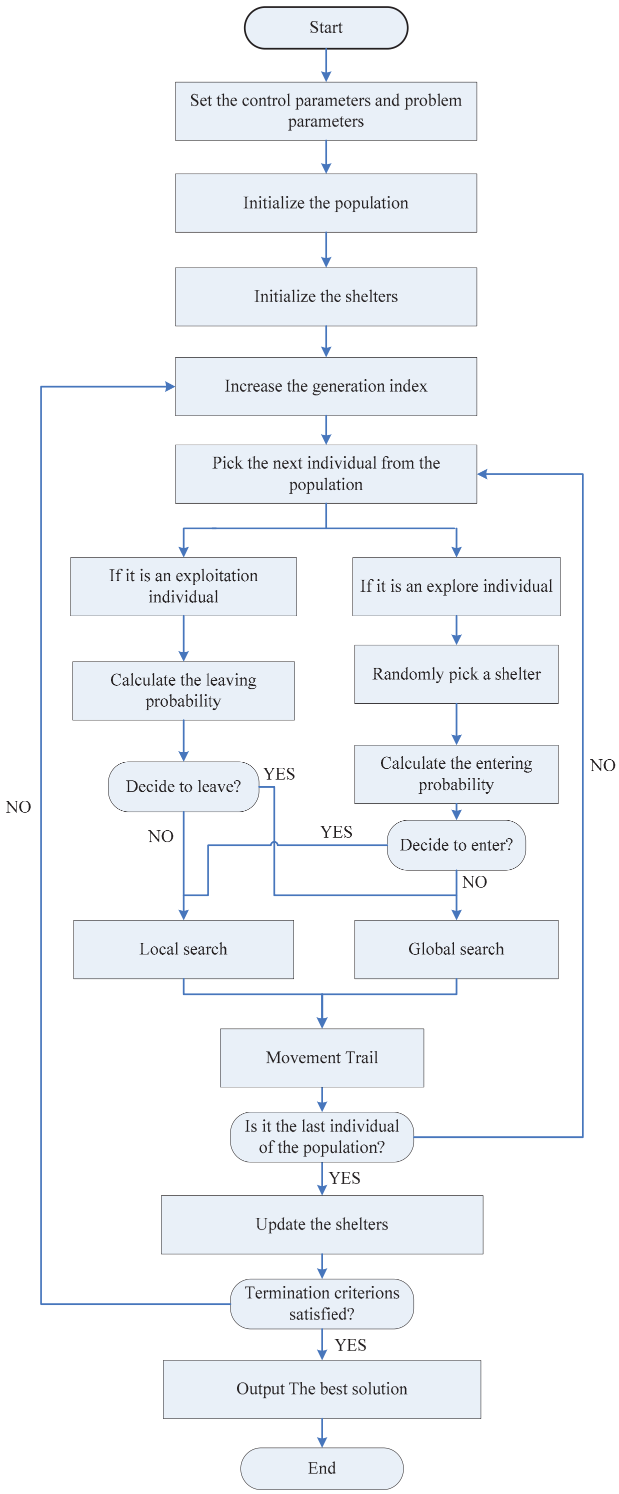

Fig. 3. Overall procedures of NAA

1, 2,…, N^S sequentially. The corresponding shelter sites and fit_{base} are also updated.

H. Termination

The algorithm terminates when one of the following termination criterions is satisfied:

- The found best solution is acceptable;
- The maximum generation time is reached.

The whole procedures of NAA are also depicted in Fig. 3.

IV. Validation

In this Section, we are going to test NAA on 7 different benchmark functions. All 7 functions are minimization problems with variable dimensional settings. These 7 functions include both unimodal and multimodal problems, and both separable and non-separable functions. For each function we test two cases: low dimensional setting (D=3) and high dimensional setting (D=100).

To validate NAA's search capability, in each case, we compare NAA with 2 state-of-the-art widely used EAs: DE/rand/1/bin (refers as 'DE' in the remaining of the paper) and standard PSO. We choose these two algorithms is because they are the most widely applied evolutionary algorithms in the last several years. Moreover, DE has been reported that it outperforms many other algorithms over the continuous spaces [5], [10]. For all the 3 algorithms, we fix the population size and maximum generation number to be 20 and 1,000 for the cases of D=3 and 40 and 6,000 for the cases of D=100. The results are obtained by averaging 30 independent trials. And all the results below 1.E-50 are reported to be 0.

For all the 7 functions, for each algorithm, we use two sets of control parameter settings for D=3 and D=100, respectively, shown in Table II. We set the values of the control parameters of DE and PSO based on the suggestions in [5], [11], [12].

A. *Benchmark Functions*

The first 4 benchmark functions are basic functions which have been widely used for testing EAs. These basic functions are separable and can be solved by using D 1-D searches. The other 3 functions rotate the basic functions by using an orthogonal matrix. For the newly rotated problems, once 1 dimension in the original decision variables is changed, all dimensions in the new decision variables are affected, which makes the rotated functions cannot be solved by just D one-dimensional searches [13].

1) First De Jong Function (Sphere)

The first De Jong function is with following form:

$$f_1(x)=\sum_{j=1}^{D} x_j^2 \tag{22}$$

The shape of the first De Jong function is a sphere. It is a unimodal function and often considered as a typical problem for testing the efficiencies of optimization methods. Its global minimum is $f_1(0,\ldots0)=0$. The 3D plot of this function is shown in Fig. 4. The initial bounds of the variables are set as $x_j \in [-100, 100]$.

2) Ackley's Function

The Ackley's function is formulated as Eq. (23), and its 3D plot is shown in Fig. 5. Ackley's function is multimodal. To find the global optimum, the optimization algorithm needs to cross the numerous local optima to search for a wider region.

$$\begin{aligned} f_2(x)= & -20\cdot\exp\left(-0.2\cdot\sqrt{D^{-1}\cdot\sum_{j=1}^{D}x_j^2}\right)- \\ & \exp\left(D^{-1}\cdot\sum_{j=1}^{D}\cos(2\pi\cdot x_j)\right)+20+\exp(1) \end{aligned} \tag{23}$$

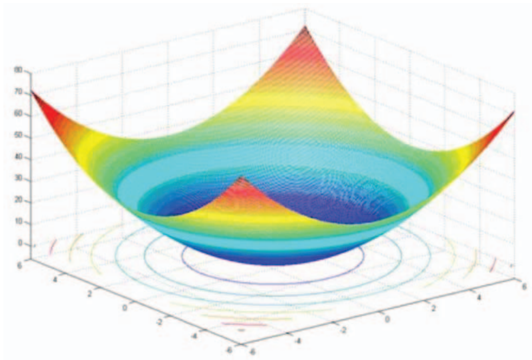

Fig. 4. Plot of the first De Jong function with 2-dimension variables

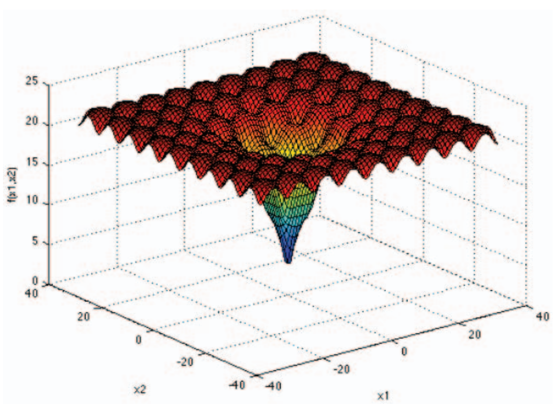

Fig. 5. Plot of the Ackley's function with 2-dimension variables

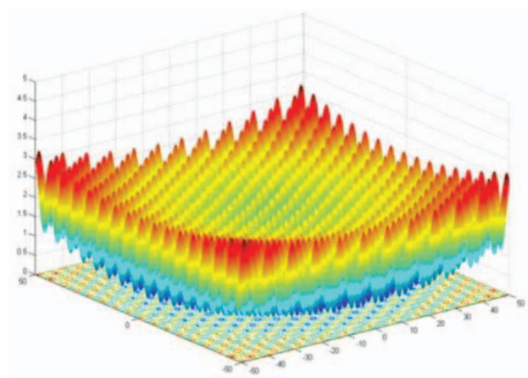

Fig. 6. Plot of the Griewank's function with 2-dimension variables

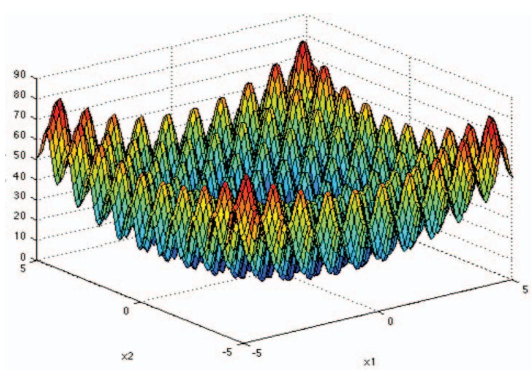

Fig. 7. Plot of the Rastrigin's function with 2-dimension variables

The global minimum of the Ackley's function is $f_2(0,...0)=0$. The initial bounds of the variables are set as $x_j \in [-32, 32]$.

3) Griewank's Function

The Griewank's function is represented as Eq. (24), and its 3D plot is shown in Fig. 6. Griewank's function is highly multimodal and is considered to be difficult to optimize.

$$f_3(x)=\sum_{j=1}^{D}\frac{x_j^2}{4000}-\prod_{j=1}^{D}\cos\left(\frac{x_j}{\sqrt{j}}\right)+1 \tag{24}$$

The global minimum of Griewank's function is $f_3(0,...0)=0$. The initial variables bounds are $x_j \in [-600, 600]$.

4) Rastrigin's Function

The Rastrigin's function is represented as Eq. (25), and its 3D plot is shown in Fig. 7. Rastrigin's function is also multimodal, and its local optima are regularly distributed.

$$f_4(x)=\sum_{i=1}^{D}\left(x_i^2-10\cos\left(2\pi x_i\right)+10\right) \tag{25}$$

The global minimum of Rastrigin's function is $f_4(0,...0)=0$. The initial variables bounds are $x_j \in [-5.12, 5.12]$.

5) Rotated Ackley's Function

The rotated Ackley's function rotates the basic Ackley's function as,

$$f_5(x)=-20\cdot\exp\left(-0.2\cdot\sqrt{D^{-1}\cdot\sum_{j=1}^{D}y_j^2}\right)-\exp\left(D^{-1}\cdot\sum_{j=1}^{D}\cos(2\pi\cdot y_j)\right)+20+\exp(1) \qquad \mathbf{y}=\mathbf{M}*\mathbf{x} \tag{26}$$

where $\mathbf{x}$ is the original decision variable vector $\mathbf{x}=[x_1,...,x_D]^T$; $\mathbf{y}$ is the rotated decision variable vector $\mathbf{y}=[y_1,...,y_D]^T$; $\mathbf{M}$ is the orthogonal matrix generated by the Salomon's method [14], which is in the form of,

$$\mathbf{M}=\begin{bmatrix} m_{11} & m_{12} & \dots & m_{1D} \\ m_{12} & m_{22} & \dots & m_{2D} \\ \dots & \dots & \dots & \dots \\ m_{D1} & m_{D2} & \dots & m_{DD} \end{bmatrix}$$

then $y_i=m_{i1}x_1+m_{i2}x_2+...+m_{iD}x_D$, $i=1,2,...,D$. The global minimum of rotated Ackley's function is $f_5(0,...0)=0$. The initial bounds of the variables are set as $x_j \in [-32, 32]$.

6) Rotated Griewank's Function

The rotated Griewank's function rotates the basic Griewank's function as,

$$f_6(x)=\frac{1}{4000}\sum_{i=1}^{D}y_i^2-\prod_{i=1}^{30}\cos\left(\frac{y_i}{\sqrt{i}}\right)+1 \qquad \mathbf{y}=\mathbf{M}*\mathbf{x} \tag{27}$$

The global minimum of this function is $f_6(0,...0)=0$. The initial bounds of the variables are set as $x_j \in [-600, 600]$.

7) Rotated Rastrigin's Function

The rotated Griewank's function rotates the basic Rastrigin's function as,

$$f_7(x)=\sum_{i=1}^{D}\left(y_i^2-10\cos\left(2\pi y_i\right)+10\right) \qquad \mathbf{y}=\mathbf{M}*\mathbf{x} \tag{28}$$

The global minimum of this function is $f_7(0,...0)=0$. The initial bounds of the variables are set as $x_j \in [-5.12, 5.12]$.

TABLE II

AVERAGED OPTIMIZATION RESULTS OF THE 3 ALGORITHMS

Dimensionality	DE-Settings		PSO-Settings			NAA-Settings					
	F	Cr	w	Cp	Cg	N^S	Cp^S	δ	Cr_{local}	α	Cr_{global}
D=3	0.9	0.1	0.729	1.49	1.49	4	$floor(\mathrm{NP}/N_{sh})$	1	0.9	1.2	0.1
D=100	0.5	0.1	-0.209	-0.079	3.764	8	$floor(\mathrm{NP}/N_{sh})$	1	0.9	1.2	0.1

Note: G_{max} denotes the maximum generation number; $floor(\bullet)$ denotes the round down operation.

TABLE III

AVERAGED OPTIMIZATION RESULTS OF THE 3 ALGORITHMS

Test Function	D	NFE			Mean Results					
					DE		PSO		NAA	
		DE	PSO	NAA	MBF	SD	MBF	SD	MBF	SD
First De Jong	3	9,709	–	**4,692**	**0**	**0**	19.88	16.17	**0**	**0**
	100	–	–	**195,770**	0.19	0.03	1.17E05	5.10E03	**0**	**0**
Ackley's	3	6,608	–	**3,309**	**0**	**0**	4.19	0.71	**0**	**0**
	100	–	–	**110,884**	0.08	0.01	19.41	0.13	**0**	**0**
Griewangk's	3	–	–	**44,555**	2.47E-04	0.0013	0.87	0.30	**0**	**0**
	100	162,060	–	**74,384**	**0**	**0**	1031.7	49.74	**0**	**0**
Rastrigin's	3	3,541	–	**2,181**	**0**	**0**	2.72	0.92	**0**	**0**
	100	–	–	–	276.77	11.36	1161.2	26.30	**152.71**	**60.52**
Rotated Ackley's	3	7,364	–	**2,954**	**0**	**0**	4.47	0.86	**0**	**0**
	100	–	–	**11,530**	2.53E-11	2.90E-11	19.56	0.12	**0**	**0**
Rotated Griewank's	3	–	–	–	0.0219	0.0096	0.9083	0.2352	**0.0061**	**0.0069**
	100	–	–	–	8.87E-5	1.2E-04	1032.3	58.06	**2.21E-13**	**8.16E-13**
Rotated Rastrigin's	3	–	–	–	0.1540	0.3375	2.7223	0.9727	**0.0670**	**0.2506**
	100	–	–	–	933.80	31.38	1176.6	28.02	**733.63**	**26.21**

Note: 'NFE' means the averaged fitness function evaluation times to find the global minimum; 'MBF' means the mean best fitness value; 'SD" means the standard deviation; '-' means fail to ensure to find the global minimum;

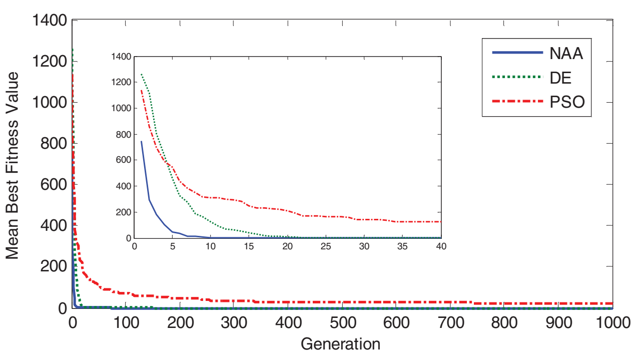

Fig. 8. Convergence profiles on the first De Jong function (D=3)

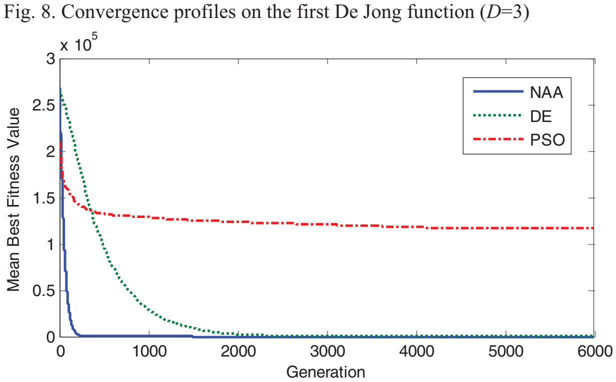

Fig. 9. Convergence profiles on the first De Jong function (D=100)

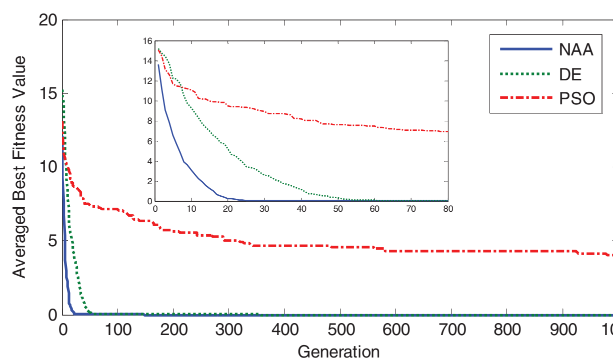

Fig. 10. Convergence profiles on the Ackley's function (D=3)

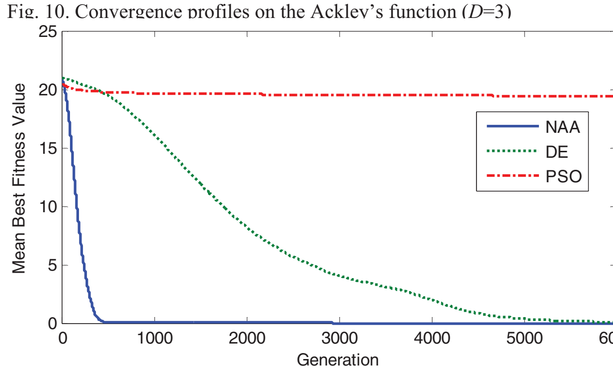

Fig. 11. Convergence profiles on the Ackley's function (D=100)

B. Experiment Results

The numerical results are reported in Table III. For each case, the base results are marked as bold texts. The mean fitness value profiles of the 3 algorithms under each test case are shown in Figs. 8-21, where the small figures are the zoomed proportions of the plotting.

Generally, for all the cases, PSO's performance is the worst. NAA outperforms DE and PSO on all the 14 cases. When D=3, for the first De Jong, Ackley's, Griewank's, Rastrigin's and rotated Ackley's functions, NAA ensures to find the global optimum on every individual trial. DE also ensures to find the global optima on these functions except for the Griewank's function. However, from Table III we can see that NAA conv-

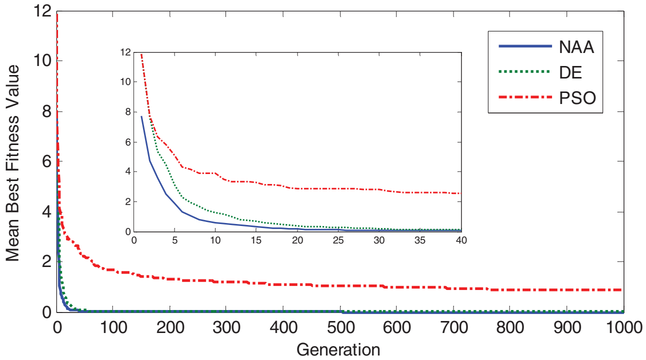

Fig. 12. Convergence profiles on the Griewank's function (D=3)

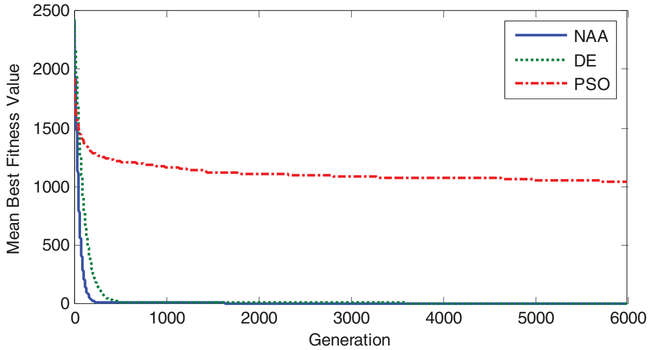

Fig. 13. Convergence profiles on the Griewank's function (D=100)

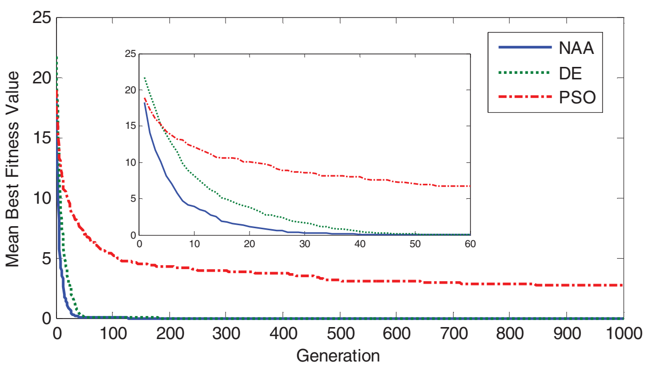

Fig. 14. Convergence profiles on the Rastrigin's function (D=3)

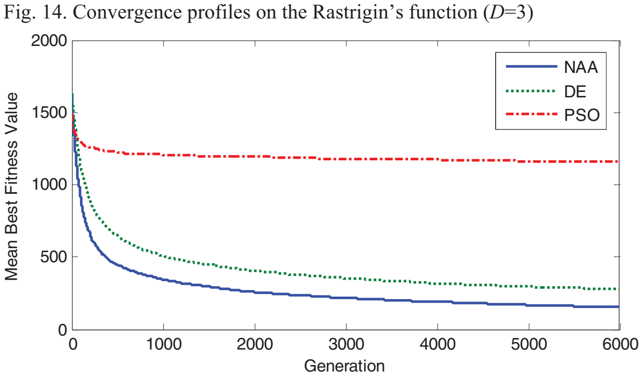

Fig. 15. Convergence profiles on the Rastrigin's function (D=100)

-erges significantly faster than DE even if both algorithms ensures to find the global optima. For example, for the Ackley's function, NAA only uses 3,309 fitness evaluation times to find the global optima, while that of DE is 6,608 times. For the Griewank's function, DE fails to find the global optima on every trial, and averagely NAA uses 44,555 fitness evaluation times to find the global optima. For the rotated Griewank's a-

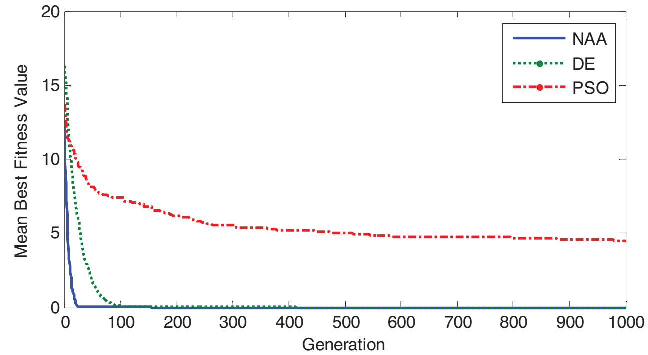

Fig. 16. Convergence profiles on the rotated Ackley's function (D=3)

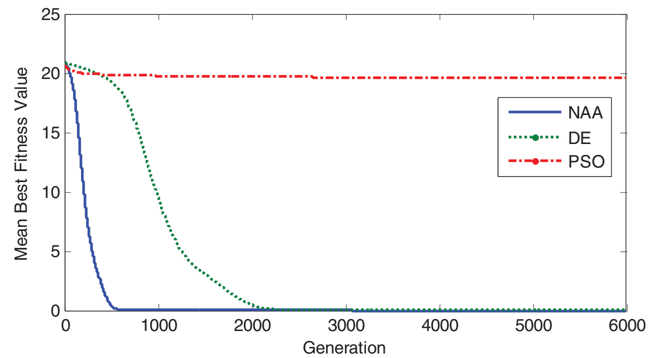

Fig. 17. Convergence profiles on the rotated Ackley's function (D=100)

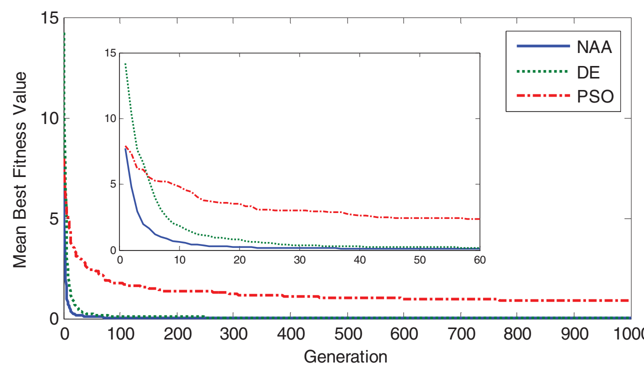

Fig. 18. Convergence profiles on the rotated Griewank's function (D=3)

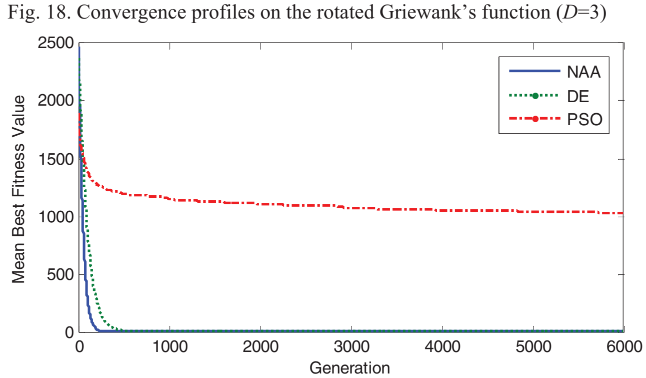

Fig. 19. Convergence profiles on the rotated Griewank's function (D=100)

-nd rotated Rastrigin's functions, all the 3 algorithms fail to ensure the finding of global optima. But it is clear that NAA obtains much better results than other 2 algorithms.

When D=100, NAA ensures to find the global optima on the First De Jong, Ackley's, Griewank's, and rotated Ackley's functions. DE only ensures to find the global optima on the Griewank's function. On the Griewank's function, DE averagely uses 162,060 fitness evaluation times to find the global

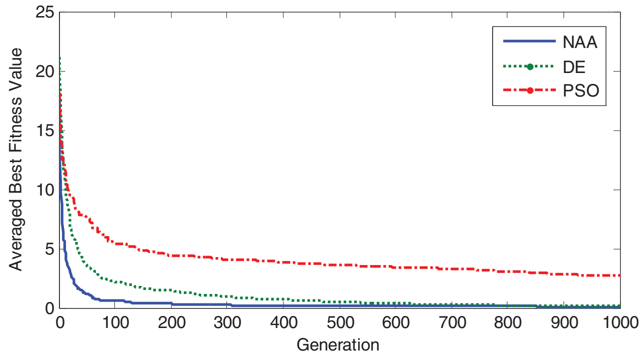

Fig. 20. Convergence profiles on the rotated Rastrigin's function (D=3)

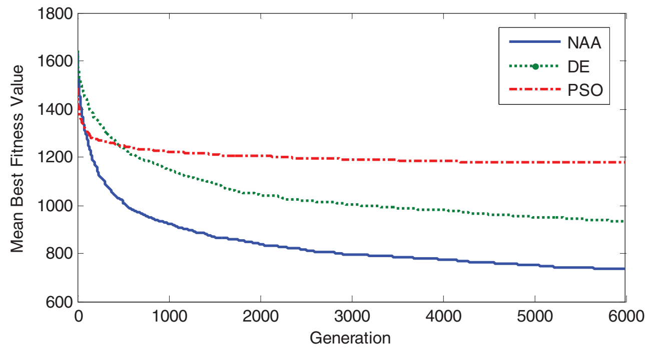

Fig. 21. Convergence profiles on the rotated Rastrigin's function (D=100)

TABLE IV
CONTROL PARAMETERS OF NAA

	Name	Meaning	Type	Empirical Range
Sub-Population Control	N^S	Number of shelters	Integer	$[1, \overline{N}]$
	Cp^S	Shelter capacity	Integer	$[2, 2\cdot avg]$
Located Search Control	δ	Scaling factor	Real	[0, 2]
	Cr_{local}	Located crossover factor	Real	[0, 1]
Generalized Search Control	α	Movement Amplification	Real	[0, 2]
	Cr_{global}	Generalized crossover factor	Real	[0, 1]

Note: $\overline{N} = floor(N_{pop}/2)$; $avg = floor(N_{pop}/N^S)$.

TABLE V
SEVERAL TYPICAL PARAMETER SETTINGS OF NAA UNDER MODERATE POPULATION SIZE

	N^S	Cp^S	δ	Cr_{local}	α	Cr_{global}
Setting 1	4	avg	1	0.9	1.2	0.1
Setting 2	2	$avg/2$	1	0.8	1	0.1
Setting 3	8	avg	2	0.8	1.5	0.5

optima while that of NAA is only 74,384. For the Rastrigin's, rotated Griewank's, and rotated Rastrigin's functions, all the 3 algorithms fail to ensure to find the global optima. But again, NAA finds significantly better results than DE and PSO.

To sum up, the experiment results clear show that NAA significantly outperforms the other two algorithms on all of the benchmark functions. NAA shows not only the strong global optimization capability, but also the fast convergence speed. It is also worth noting that Table II shows that the control parameter settings of NAA are kept the same for all the test cases, which indicates the robustness of NAA.

V. EMPIRICALLY PARAMETER SELECTION GUIDELINES

NAA has 6 control parameters, as listed in Table IV. There are two parameters for balancing the sub-populations, two parameters for controlling the located search behaviors, and two parameters for controlling the generalized search behaviors, respectively. In this sense, NAA provides a highly flexible framework to generate different search patterns by combing different control parameter values of the 3 modules.

Some empirical ranges of the control parameters are drawn by the authors. These empirically control parameter ranges are given in Table IV. Some empirical parameter selection guidelines are also drawn by the authors. Normally, the smaller value of N^S and larger value of Cr_{local} can make the algorithm fast converge to an acceptable solution, but the algorithm might be trapped to the local optima. For a moderate dimensionality problem (D=10 to 50) and moderate population size, set N^S to be 4-8 and Cr_{local} to be 0.7-1 would be a good choice. In the authors' tests, setting Cp^S to be $floor(N_{pop}/N^S)$ can yield good performance on most of the test cases. But if the value of N^S is too small (e.g., $N^S = 1$ or 2), it would be reasonable to reduce the shelter capacity (e.g., set $Cp^S = avg/2$) to ensure the adequate generalized search activities of the individuals. The authors' experiences also show that the algorithm's performance would be deteriorated if α is set to be larger than 2.

Table V gives 3 typical settings of NAA. In the authors' tests these settings are proved to be able to yield good convergence on many benchmark functions. If the user has no prior knowledge about the landscape of the problem, or if he/she is a new user of NAA, use the settings listed in Table V could be a good start. Meanwhile, the detailed parameter sensitivity analysis of NAA on a wider range of benchmark functions and real-applications are still needed to be investigated in future.

VI. NAMING PHILOSOPHY

In sections I and II, we introduced some biological background of NAA. In this Section, we summarize following 5 features of the aggregation behaviors of many group-living species to explain our naming philosophy of NAA.

(a) **Autonomous**. The aggregation behaviors of the group-living animals are completely autonomous and without any centralized control schemes;

(b) **Limited Cognitive Abilities**. The individuals of the swarm make decisions based on their limited local knowledge;

(c) **Social Attractions**. The individuals will exchange information with each other to find and share the resources;

(d) **Competition**. The crowding effect of the limited resources influences the behaviors of the swarm members;

(c) **Dynamic Stability**. The aggregation of the group-living animals will evolve to a dynamic stable status rather than an absolutely static state.

For NAA, we choose the word 'natural' based on the realization that although the mathematical migration model of NAA inherits from the cockroach aggregation experiments, the inspiration of NAA is actually from the aggregation phenomenon of the group-living animals, which are widely existed in the nature. All above 5 characteristics are reflected in the evolution mechanism of NAA. In NAA, the individuals are autonomously organized into multiple sub-populations and make their search decisions based on their local knowledge; the social attractions and competitions are modelled by the stochastic migration model; and in NAA, every individual has a certain probability to do the dynamic migration during the whole evolution process.

VII. Conclusions

In this paper, a new metaheuristic optimization algorithm over the continuous space is proposed, which is called the natural aggregation optimization. NAA mimics the self-organized aggregation behaviors of the group-living animals. The algorithm divides the whole population into multiple sub-populations, and iteratively search for the global/near-global optimum in the problem space. In each generation, the individuals make decisions on entering/leaving a sub-population based on the quality and crowding degree of the sub-population. Correspondingly, the individuals will do the located and generalized searches in the problem space. The experiment results show that the proposed algorithm has stronger global search capability than DE and PSO in both of the low-dimensional and high-dimensional continuous spaces.

Currently, the authors are working on comparing NAA and more other heuristic algorithms on a wider range of benchmark functions. The authors are looking forward to seeing the applications of NAA on different real-world problems in the future.

References

[1] R. Haupt and S. Haupt, *Practical Genetic Algorithms*, Second Edition, New Jersey: John Wiley & Sons, 2004.

[2] H. Bremermann, "Optimization through evolution and recombination," in *Self-Organizing Systems*, M.C. Yovits, G.T. Jacobi, and G.D. Goldstine, Eds. Washington, DC: Spartan, pp. 93-106, 1962.

[3] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671-680, 1983.

[4] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Network*, pp. 1942-1948, Nov. 1995.

[5] R. Storn and K. Price, "Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341-359, 1997.

[6] K. Han and J. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 6, pp. 580-593, 2002.

[7] J. Krause and G. Ruxton, *Living in Groups*, Oxford University Press, 2002.

[8] S. Boinski and P.A. Garber, *On the Move: How and Why Animals Travel in Groups*, University of Chicago Press, 2000.

[9] J. Ame, J. Halloy, C. Rivault, C. Detrain, and J. Deneubourg, "Collegial decision making based on social amplification leads to optimal group formation," *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, vol. 103, no. 15, pp. 5835-5840, 2006.

[10] J. Vesterstrom and R. Thomson, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark functions," in *Proc. Congress on Evolutionary Computation*, 2004.

[11] Good parameters for particle swarm optimization, [Online]. Available: http://www.cof.orst.edu/cof/teach/fe640/Class_Materials/Particle%20Swarm/PSO%20parameters.pdf

[12] M. Jiang, Y. Luo, and S. Yang, "Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm," *Information Processing Letters*, vol. 102, pp. 8-16, 2007.

[13] J. Liang, A. Qin, P. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimization for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281-295, Jun. 2006.

[14] R. Salomon, "Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions," *BioSystems*, vol. 39, pp. 263-278, 1996.

Acknowledgment

The authors gratefully acknowledge Dr. Yang Chen from Microsoft Research Asia for his valuable assistance to do the proofreading of this paper. The authors also would like to thank the anonymous reviewers for their valuable comments on the improvement of this paper.

Fengji Luo received the B.S. and M.S. degrees in software engineering and computer science from the Chongqing University, China, in 2006 and 2009, respectively. He received the Ph.D. degree in electrical engineering from the University of Newcastle, Australia, in 2013. Currently, he is the postdoctoral research associate of the Faculty of Engineering, The University of Sydney, Australia. He also held academic positions with the Hong Kong Polytechnic University, Hong Kong, and the Centre for Intelligent Electricity Networks, The University of Newcastle, Australia. He was awarded by the Pro Vice Chancellor's 2015 Award for Research Excellence, The University of Newcastle, Australia. His research interests include the evolutionary computation, computational intelligence, and smart grid applications.

Junhua Zhao received his Ph.D. degree from the University of Queensland, Australia. Currently he is the associate professor of the Chinese University of Hong Kong (Shen Zhen), China. Before that, he was the senior lecturer of the Centre for Intelligent Electricity Networks, The University of Newcastle, Australia. His research interests include power system analysis and computation, smart grid, cyber physical system, electricity market, computational intelligence and its applications in smart grids.

Zhao Yang Dong obtained his Ph.D. degree from the University of Sydney, Australia in 1999. He is now professor and Head of Schol of Electrical and Informationb Engineering, the University of Sydney, Australia. His immediate role was Ausgrid Chair and Director of the Centre for Intelligent Electricity Networks (CIEN), The University of Newcastle, Australia. He also held academic and industrial positions with the Hong Kong Polytechnic University, Hong Kong, and Transend Networks, Tasmania, Australia. His research interests include power system operation & planning, smart grid, renewable energy systems, computational intelligence, and evolutionary computation. He is an editor of IEEE TRANSACTIONS ON SMART GRID, IEEE TRANSACTIONS OF SUSTAINABLE ENERGY, IEEE PES Letters and IET Renewable Power Generation.