Research Article

Simulated Annealing-Based Krill Herd Algorithm for Global Optimization

Gai-Ge Wang,[1,2] Lihong Guo,[1] Amir Hossein Gandomi,[3] Amir Hossein Alavi,[4] and Hong Duan[5]

[1] *Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China*

[2] *Graduate School of Chinese Academy of Sciences, Beijing 100039, China*

[3] *Department of Civil Engineering, University of Akron, Akron, OH 44325-3905, USA*

[4] *Department of Civil and Environmental Engineering, Engineering Building, Michigan State University, East Lansing, MI 48824, USA*

[5] *School of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China*

Correspondence should be addressed to Lihong Guo; guolh@ciomp.ac.cn

Recently, Gandomi and Alavi proposed a novel swarm intelligent method, called krill herd (KH), for global optimization. To enhance the performance of the KH method, in this paper, a new improved meta-heuristic simulated annealing-based krill herd (SKH) method is proposed for optimization tasks. A new krill selecting (KS) operator is used to refine krill behavior when updating krill's position so as to enhance its reliability and robustness dealing with optimization problems. The introduced KS operator involves greedy strategy and accepting few not-so-good solutions with a low probability originally used in simulated annealing (SA). In addition, a kind of elitism scheme is used to save the best individuals in the population in the process of the krill updating. The merits of these improvements are verified by fourteen standard benchmarking functions and experimental results show that, in most cases, the performance of this improved meta-heuristic SKH method is superior to, or at least highly competitive with, the standard KH and other optimization methods.

1. Introduction

In management science, mathematics, and economics, the process of optimization is the selection of the best solution from some set of feasible alternatives. More generally, optimization consists of finding the optimal values of some objective function within a given domain. In general, a great many optimization techniques have been developed and applied to solve optimization problems [1]. A general classification way for these optimization techniques is considering the nature of these techniques, and these optimization techniques can be categorized into two main groups: deterministic methods and modern intelligent algorithms. Deterministic methods using gradient such as hill climbing follow a rigorous step and will repeat the process of optimization if the iterations start with the same initial starting point. Eventually, they will reach the same set of solutions. On the other hand, modern intelligent algorithms without adopting gradient always have some randomness, and the process of optimization cannot be repeatable even with the same initial value. However, generally, the final solutions, though slightly different, will arrive at the same optimal values within a given accuracy [2]. The growth of stochastic optimization methods as a blessing from the mathematical and computing theorem has opened up a new facet to complete the optimization of a function. Recently, nature-inspired metaheuristic methods perform efficiently and effectively in solving modern nonlinear numerical global optimization problems. To some extent, all metaheuristic methods make an attempt at making balance between diversification (global search) and intensification (local search) [2, 3].

Inspired by nature, these strong metaheuristic methods have ever been applied to solve a variety of complicated problems, such as task-resource assignment [4], constrained

optimization [5], test-sheet composition [6], and water, geo-technical, and transport engineering [7, 8]. All the problems that need to find extreme value could be solved by these optimization methods. These types of metaheuristic methods use a population of solutions and always obtain optimal or suboptimal solutions. During the 1960s and 1970s, computer researchers investigated the possibility of formulating evolution as an optimization method and this eventually generated a subset of gradient-free methods, namely genetic algorithms (GAs) [2, 9]. In the last twenty years, a great number of classical optimization techniques have been developed on function optimization, such as bat algorithm (BA) [10], bacterial foraging optimization [11], biogeography-based optimization (BBO) [12–14], modified Lagrangian method [15], artificial plant optimization algorithm (APOA) [16], artificial physics optimization [17], differential evolution (DE) [18, 19], genetic programming [20], particle swarm optimization (PSO) [21–25], cuckoo search (CS) [26], and, more recently, the krill herd (KH) method [27] that is inspired by the herding behavior of krill individuals in nature [2].

Firstly presented by Gandomi and Alavi in 2012, based on the simulation of the herding behavior of krill individuals, KH algorithm is a novel metaheuristic search approach for optimizing possibly nondifferentiable and nonlinear functions in continuous space [2, 27]. In KH, the objective function for the krill motion is manly influenced by the minimum distances of each krill individual from food and from highest density of the herd. The position of the krill consists of three main motions: (i) foraging motion, (ii) movement led by other individuals, and (iii) random physical diffusion. KH algorithm does not need derivative information, because it uses a stochastic search instead of a gradient search used in deterministic methods. Furthermore, comparing with other metaheuristic approaches, this new method requires few control parameters, in effect, only a single parameter Δt (time interval), to adjust, which makes KH implement simply and easily, and is well fitted for parallel computation [2].

KH is an efficient algorithm in exploration but at times it may trap into some local best values so that it cannot search globally well [2]. For KH, the search only relies on random walks, so there is no guarantee for a fast convergence [2]. To improve KH for solving complicated optimization problems, a few techniques have been introduced into the standard KH method [2, 28], which increase the diversity of population and greatly enhance the performance of the basic KH method.

On the other hand, inspired by the annealing process, simulated annealing (SA) algorithm [29] is a probabilistic meta-heuristic search method. Differently form most intelligent algorithm, SA is a trajectory-based optimization technique [30].

Firstly presented here, a novel metaheuristic SKH method based on KH and SA is originally proposed in this paper, with the aim of accelerating convergence speed, thus making the approach more feasible for a wider range of practical applications without losing the attractive merits of the canonical KH approach. In SKH, firstly, we use a standard KH method to select a good candidate solution set. And then, a new krill selecting (KS) operator is introduced into the basic KH

method. The introduced KS operator involves greedy strategy and accepting a not-so-good solution with a low probability originally used in simulated annealing (SA) [3]. This greedy strategy is applied to decide whether a good candidate solution is accepted so as to improve its efficiency and reliability for solving global numerical optimization problem. Furthermore, to improve the exploration of KH and evade premature convergence, the concept of acceptance probabilities in KS operator is introduced into the method through accepting a not-so-good solution with a low probability in place of previous solution in order to enhance the adversity of the population. Fourteen standard benchmark functions, which have ever been applied to verify optimization algorithms in continuous optimization problems, are used to evaluate our proposed method. Experimental results show that the SKH performs more efficiently and accurately than basic KH, ABC, BA, CS, DE, ES, GA, HS, KH, PBIL, PSO, and SA.

The structure of this paper is organized as follows. Section 2 gives a brief description of basic KH and SA algorithms. Our proposed SKH method is described in detail in Section 3. Subsequently, comparing with ABC, BA, CS, DE, ES, GA, HS, KH, PBIL, PSO, and SA, the merits of our method are verified by 14 benchmark functions in Section 4. Finally, Section 5 provides the conclusion and proposals for future work.

2. Preliminary

At first, in this section we will provide a background on the krill herd and simulated annealing algorithm in brief.

2.1. Krill Herd Algorithm. Krill herd (KH) [2, 27] is a novel metaheuristic search method for optimization tasks, which mimics the herding of the krill swarms in response to specific biological and environmental processes. The position of an individual krill in search space is mainly influenced by three motions described as follows [2]:

(i) foraging action,

(ii) movement influenced by other krill individuals,

(iii) physical diffusion.

In KH method, the following Lagrangian model in a d-dimensional decision space is used as shown in the following:

$$\frac{dX_i}{dt} = F_i + N_i + D_i, \tag{1}$$

where F_i, N_i, and D_i are the foraging motion, the motion induced by other krill individuals, and the random physical diffusion of the krill i [27].

The foraging motion includes two components that are the current food location and the previous experience about the food location. For the krill i, this motion can be formulated as follows [2]:

$$F_i = V_f \beta_i + \omega_f F_i^{\text{old}}, \tag{2}$$

where

$$\beta_i = \beta_i^{\text{food}} + \beta_i^{\text{best}} \tag{3}$$

and V_f is the foraging speed, ω_f is the inertia weight of the foraging motion in $(0, 1)$, and F_i^{old} is the last foraging motion. β_i^{food} is the food attractive and β_i^{best} is the effect of the best fitness of the ith krill so far [2, 27].

In movement induced by other krill, the direction of motion induced, α_i, is approximately calculated by the following three factors: local effect (a local swarm density), target effect (the target swarm density), and repulsive effect (a repulsive swarm density). For a krill, this movement can be expressed as follows [2, 27]:

$$N_i^{\text{new}} = N^{\text{max}}\alpha_i + \omega_n N_i^{\text{old}} \tag{4}$$

and N^{max} is the maximum induced speed, ω_n is the inertia weight of the motion induced in $[0, 1]$, and N_i^{old} is the last motion induced [2].

For the krill individuals, the physical diffusion can be looked on as a random process. This motion can be decided by a maximum diffusion speed and a random directional vector. It can be provided as follows [2]:

$$D_i = D^{\text{max}}\delta, \tag{5}$$

where D^{max} is the maximum diffusion speed, and δ is the random directional vector whose values are random numbers in $[-1, 1]$ [2].

Based on the three previously mentioned movements, using different parameters of the motion during the time, the position of the ith krill during the interval t to $t + \Delta t$ is expressed by [2]:

$$X_i(t + \Delta t) = X_i(t) + \Delta t \frac{dX_i}{dt}. \tag{6}$$

Note that Δt is one of the most significant constants and should be fine-tuned in terms of the given real-world optimization problem [2]. This is because this parameter can be considered as a scale factor of the speed vector. More details about the three actions and KH approach can be found in [2, 27].

2.2. Simulated Annealing. Simulated annealing (SA) algorithm is a stochastic search technique that originated from statistical mechanics. The SA method is inspired by the annealing process of metals. In the annealing process, a metal is heated to a high temperature and gradually cooled to a low temperature that can crystallize. As the heating process lets the atoms travel randomly, if the cooling is done slowly enough, so the atoms have enough time to adjust themselves so as to reach a minimum energy state. This analogy can be applied in function optimization with the state of metal corresponding to the possible and the minimum energy state being the final best solution [30].

The SA method repeats a neighbor generation procedure and follows search paths that minimize the objective function value. When exploring search space, the SA provides the possibility of accepting worse generating solutions in a special manner in order to avoid trapping into local minima. More precisely, in each generation, for a current solution X whose

value is $f(X)$, a neighbor X' is chosen from the neighborhood of X denoted by $N(X)$. For each step, the objective difference $\Delta = f(X') - f(X)$. X' could be accepted with a probability calculated by [30]

$$P_s = \exp\left(-\frac{\Delta}{T}\right). \tag{7}$$

And then, this acceptance probability is compared to a random number $r \in (0, 1)$ and X' is accepted whenever $p > r$. T is temperature controlled by a cooling scheme [30].

The SA method includes specific factors: a neighbor generation move, objective function calculation, a method for assigning the initial temperature, a procedure to update the temperature, and a cooling scheme including stopping criteria [30].

3. Our Approach: SKH

For the original KH approach, as the search only relies on random walks, a rapid convergence cannot always be guaranteed. To improve its performance, genetic reproduction mechanisms have added to the KH approach [27]. It has been demonstrated that, comparing other approaches, KH II (KH with crossover operator) performed the best. In effect, KH makes full use of the three motions in the population and it is shown that the KH performs well in both convergence speed and final accuracy on unimodal problems and most simple multimodal problems. However, once in a while, in a rough region of the fitness landscape, KH cannot always succeed in finding better solutions on some complex problems. In our present study, in order to further improve the performance of KH, a modified greedy strategy and mutation scheme, called krill selecting (KS) operator, is introduced into the KH method to design a novel simulated annealing-based krill herd (SKH) algorithm. The introduced KS operator is inspired from the classical simulated annealing algorithm. That is to say, in our work, the physical property of metal is added to the krill to produce a type of super krill that is able to perform the KS operator. The difference between SKH and KH is that the KS operator is applied to only accept the basic KH generating new better solution for each krill instead of accepting all the krill updating adopted in KH. This is rather greedy. The standard KH is very efficient and powerful, but the solutions have slight changes as the optima are approaching in the later run phase of the search. Therefore, to evade premature convergence and further improve the exploration ability of the KH, KS operator also accepts few not-so-good krill with a low acceptance probability p as new solution. This probability p is also called transition probability. This acceptance probability technique can increase diversity of the population in an effort to avoid premature convergence and explore a large promising region in the prior run phase to search the whole space extensively. The main step of KS operator adopted in SKH method is given in Algorithm 1.

In Algorithm 1, to begin with, the temperature is updated according to

$$T = \alpha * T. \tag{8}$$

```
Begin
    T = α * T;
    Δf = f(X'_i) − f(X_i);
    % Accept if improved
    If (−Δf > f_n) then do
        X_{i+1} = X'_i;
    end if
    % Accept with a low probability if not improved
    If (Δf <= f_n & exp(−Δf/(k * T)) > r) then do
        X_{i+1} = X'_i;
    end if
End.
```

ALGORITHM 1: Krill selecting (KS) operator.

Here, T is the temperature for controlling the acceptance probability p. And then, the change of the objective function value Δf is computed by (9)

$$\Delta f = f\left(X_i'\right) - f\left(X_i\right). \tag{9}$$

Here, X_i' is new generating krill for krill i by three motions in basic KH. Whether or not we accept a change, we usually use a constant number f_n as a threshold. If $-\Delta f > f_n$, the newly generating krill X_i' is accepted as its latest position for krill i. Otherwise, when (10) is true, the newly generating krill X_i' is also accepted as its latest position for krill i

$$p = \exp\left(\frac{-\Delta f}{k * T}\right) > r. \tag{10}$$

Here, r is random number drawn from uniform distribution in $(0, 1)$. k is Boltzmann's constant. For simplicity without losing generality, we use $k = 1$ in our present study.

In SKH, the critical operator is the KS operator that comes from SA algorithm, which is similar to the LLF operator used in LKH [2]. The core idea of the proposed KS operator is based on two considerations. Firstly, good solutions can make the method converge faster. Secondly, the KS operator can significantly improve the exploration of the new search space.

In SKH, at first, standard KH method with high convergence speed is used to shrink the search region to a more promising area. And then, KS operator with great greedy ability is applied to only accept better solutions to improve the quality of the whole population. In this way, SKH method can explore the new search space with KH and extract optimal population information by KS operator. In addition, transition probability p in KS operator is applied to accept few nonimproved krill with a low acceptance probability p in an effort to increase diversity of the population and evade premature convergence.

In addition, except krill selecting operator, another vital improvement is the addition of elitism strategy into the SKH approach. Undoubtedly, both KH and SA have some basic elitism. However, it can be further enhanced. As with other optimization approaches, we introduce some sort of elitism with the aim of holding some optimal krill in the population. Here, a more intensive elitism on the optimal krill is applied, which can stop the optimal krill from being spoiled by three motions and krill selecting operator. In SKH, at first, the *KEEP* optimal krill are memorized in a vector *KEEPKRILL*. In the end, the *KEEP* worst krill are replaced by the *KEEP* stored optimal krill. This elitism strategy always has a guarantee that the whole population cannot decline to the population with worse fitness. Note that, in SKH, an elitism strategy is applied to keep some excellent krill that have the optimal fitness, so even if three motions and krill selecting operator corrupts its corresponding krill, we have memorized it and can recover to its previous good status if needed.

By integrating previously mentioned krill selecting operator and intensive elitism strategy into basic KH approach, the SKH has been designed that can be presented in Algorithm 2. Here, NP is the size of the parent population P.

4. Simulation Experiments

In this section, the effectiveness of our proposed SKH method is tested to global numerical optimization through a number of experiments implemented in standard benchmark functions.

To allow an unbiased comparison of time requirements, all the experiments were conducted in the same hardware and software environments with [2].

Well-defined problem sets are fitted for verifying the performance of optimization approaches proposed in this paper. Based on mathematical expressions [31], benchmark functions can be considered as objective functions to implement such tests. In our present work, fourteen different benchmark functions are used to evaluate our proposed metaheuristic SKH method. The formulation of these benchmark functions and their properties can be found in [12, 32]. It is worth pointing out that, in [32], Yao et al. have ever used 23 benchmarks to test optimization approaches. However, for the other low-dimensional benchmark functions (d = 2, 4, and 6), all the methods perform slightly differently with each other [33], because these low-dimensional benchmarks are so simple that they cannot distinguish the performance difference among different approaches. Thus, in our study, only fourteen high-dimensional benchmarks are applied to verify our proposed SKH method [2].

```
Begin
    Step 1: Initialization. Set the generation counter t, the population P of NP krill, the foraging
        speed V_f, the diffusion speed D^max, and the maximum speed N^max,
        temperature T_0, Boltzmann constant k, cooling factor α, and an acceptance threshold number f_n,
        and elitism parameter KEEP.
    Step 2: Fitness calculation. Calculate the fitness for each krill based on their initial position.
    Step 3: While t < MaxGeneration do
            Sort all the krill according to their fitness.
            Store the KEEP best krill.
            for i = 1 : NP (all krill) do
                Implement three motion as described in Section 2.1.
                Update position for krill i by krill selecting operator in Algorithm 1.
                Calculate the fitness for each krill based on its new position X_{i+1}.
            end for i
            Substitute the KEEP best krill for the KEEP worst krill.
            Sort all the krill according to their fitness and find the current best.
            t = t + 1;
    Step 4: end while
    Step 5: Output the best solution.
End.
```

ALGORITHM 2: Simulated annealing-based krill herd algorithm.

4.1. General Performance of SKH. In this section, the performance of SKH approach on global numeric optimization problem with eleven optimization methods was compared so as to look at the merits of SKH. The eleven optimization methods are ABC (artificial bee colony) [34], BA (bat algorithm) [10], CS (cuckoo search) [35], DE (differential evolution) [18], ES (evolutionary strategy) [36], GA (genetic algorithm) [9], HS (harmony search) [1, 37], KH [27], PBIL (probability-based incremental learning) [38], PSO (particle swarm optimization) [21, 39, 40], and SA [29]. More information about these comparative methods can be found in [2]. Besides, we must point out that, in [27], comparing all the algorithms, the experimental results show that the performance of KH II was the best which proves the superiority of the KH approach. Consequently, in our present study, we use KH II as basic KH method.

In the following experiments, the same parameters for KH, SA and SKH are adopted, which are the foraging speed $V_f = 0.02$, the maximum diffusion speed $D^{max} = 0.005$, the maximum induced speed $N^{max} = 0.01$, initial temperature $T_0 = 1.0$, maximum number of accept $Accept_{max} = 15$, Boltzmann constant $k = 1$, cooling factor $\alpha = 0.95$, and an acceptance threshold number $f_n = 0.01$ (only for SKH). For other methods used here, their parameters are selected as [2, 12, 41].

We set population size $NP = 50$ and maximum generation Maxgen = 50 for each approach. We ran 100 Monte Carlo simulations of each approach on each benchmark function to get typical performances [1]. The results of the experiments are recorded in Tables 1 and 2. Table 1 illustrates the average minima found by each approach, averaged over 100 Monte Carlo runs. Table 2 illustrates the absolute best minima found by each approach over 100 Monte Carlo runs. That is to say, Table 1 represents the average performance of each approach, while Table 2 represents the best performance

of each approach. The best value obtained for each test problem is shown in bold. Note that the normalizations in the tables are based on different scales, so values are not comparative between the two tables. For each function used in our work, their dimension is 20.

From Table 1, we see that, on average, SKH is the most effective at finding objective function minimum on eleven of the fourteen benchmarks (F01–F06, F08, F10, and F12–F14). ABC, CS, and GA are the second most effective, performing the best on the benchmark, F07, F11, and F09 when multiple runs are made, respectively. Table 2 shows that SKH performs the best on thirteen of the fourteen benchmarks which are F01–F08 and F10–F14. GA is the second most effective, performing the best on the benchmark F09 when multiple runs are made.

Moreover, the running time of the twelve optimization approaches was slightly different. We collected the average CPU time of the optimization methods as applied to the 14 benchmarks considered in our work. The results are marked in Table 1. PBIL was the quickest optimization method. SKH was the ninth fastest of the twelve approaches. However, it should be noted that in most real-world applications, it is the fitness function evaluation that is by far the most time-consuming part of an optimization approach.

In addition, to prove the superiority of the proposed method in the future, convergence graphs of ABC, BA, CS, DE, ES, GA, HS, KH, PBIL, PSO, SA, and SKH are also provided in this section. However, limited by the length of paper, here only some most typical benchmarks are illustrated in Figures 1–7 which mean the process of optimization. The function values shown in Figures 1–7 are the average objective function minimum obtained from 100 Monte Carlo simulations, which are the true objective function solution, not normalized. By the way, note that the global optima of the benchmarks (F04, F05, F08, F10, and F14) are illustrated

Table 1: Mean normalized optimization results in fourteen benchmark functions.

	ABC	BA	CS	DE	ES	GA	HS	KH	PBIL	PSO	SA	SKH
F01	5.86	8.43	4.66	5.28	8.17	7.17	8.35	1.84	8.45	7.02	7.16	**1.00**
F02	5.29	28.57	4.34	7.44	20.25	7.26	17.66	6.70	17.41	15.67	1.66	**1.00**
F03	42.23	224.13	17.11	22.27	102.89	43.18	206.61	6.29	229.90	79.38	1.57	**1.00**
F04	1.2E6	5.8E7	3.3E3	1.9E5	2.8E7	3.8E5	4.2E7	1.3E4	6.0E7	4.1E6	12.54	**1.00**
F05	3.0E7	8.3E8	1.2E6	1.0E7	4.2E8	1.5E7	6.4E8	1.1E6	8.2E8	9.6E7	188.67	**1.00**
F06	3.6E6	6.3E7	4.5E5	1.3E6	4.5E7	3.5E6	4.2E7	4.2E5	5.3E7	1.0E7	5.4E3	**1.00**
F07	**1.00**	2.75	1.14	1.63	2.54	1.70	2.41	1.01	2.61	1.94	1.05	1.01
F08	16.46	99.06	6.23	15.71	132.91	28.27	88.88	7.30	104.89	34.65	2.75	**1.00**
F09	1.76	3.94	1.80	2.23	2.76	**1.00**	3.38	2.13	3.51	3.36	1.63	2.01
F10	416.63	970.18	101.00	514.39	586.60	441.37	537.77	267.70	555.24	387.49	71.95	**1.00**
F11	1.06	646.05	**1.00**	1.20	4.55	2.15	3.57	1.72	3.63	2.95	1.88	4.35
F12	26.50	29.62	11.15	21.16	26.82	21.31	26.94	4.23	28.03	22.22	12.75	**1.00**
F13	2.7E3	1.4E4	991.45	1.3E3	1.5E4	5.0E3	1.3E4	364.49	1.5E4	5.4E3	38.11	**1.00**
F14	225.79	1.2E3	87.39	116.81	794.09	233.34	1.2E3	28.01	1.3E3	459.13	3.56	**1.00**
Time	2.39	1.11	2.58	1.98	2.05	2.40	2.83	4.73	**1.00**	2.42	1.88	2.54
Total	1	0	1	0	0	1	0	0	0	0	0	11

*The values are normalized so that the minimum in each row is 1.00. These are not the absolute minima found by each algorithm, but the average minima found by each algorithm.

Table 2: Best normalized optimization results in fourteen benchmark functions.

	ABC	BA	CS	DE	ES	GA	HS	KH	PBIL	PSO	SA	SKH
F01	14.43	23.66	10.43	13.84	23.34	15.83	24.81	3.03	24.37	19.35	8.84	**1.00**
F02	12.93	78.18	9.36	19.93	64.37	7.42	46.93	15.76	56.66	46.26	1.86	**1.00**
F03	10.25	88.97	8.48	13.29	47.57	8.71	128.96	2.82	157.15	53.02	1.33	**1.00**
F04	214.81	2.6E7	137.18	2.9E4	9.1E7	183.20	2.8E8	366.68	6.3E8	3.9E6	223.10	**1.00**
F05	5.4E6	5.6E8	4.5E4	8.5E6	7.9E8	1.5E5	1.1E9	7.4E5	2.0E9	9.2E7	48.84	**1.00**
F06	7.5E6	5.3E8	3.3E6	1.1E7	9.1E8	8.8E6	5.0E8	3.7E6	1.1E9	9.7E7	2.2E3	**1.00**
F07	10.97	29.54	11.22	19.21	31.10	18.09	27.12	8.72	34.18	25.25	9.89	**1.00**
F08	8.22	37.08	3.50	6.76	75.99	11.20	51.34	3.84	67.57	20.31	1.06	**1.00**
F09	4.06	7.72	3.13	5.25	6.54	**1.00**	8.36	4.83	8.72	6.69	2.25	3.22
F10	870.11	1.6E3	166.38	1.1E3	1.1E3	563.34	1.2E3	387.57	994.72	675.38	30.48	**1.00**
F11	3.01	13.91	3.37	4.21	15.66	6.44	15.39	6.65	15.01	7.46	1.82	**1.00**
F12	69.59	65.27	22.47	46.15	75.13	48.37	65.04	5.69	65.77	55.79	21.52	**1.00**
F13	6.3E3	3.9E4	2.6E3	5.1E3	6.9E4	1.4E4	5.5E4	796.88	6.1E4	1.3E4	108.69	**1.00**
F14	696.00	4.3E3	281.00	444.33	3.4E3	359.67	5.0E3	67.67	5.9E3	2.1E3	10.67	**1.00**
Total	0	0	0	0	0	1	0	0	0	0	0	13

*The values are normalized so that the minimum in each row is 1.00. These are the absolute best minima found by each algorithm.

in the form of the semilogarithmic convergence plots. We use KH short for KH II in the legend of Figures 1–7 and next texts.

Figure 1 shows the results obtained for the twelve methods when the F01 Ackley function is applied. From Figure 1, clearly, we can draw the conclusion that SKH is significantly superior to all the other algorithms during the process of optimization. For other algorithms, although slower, KH eventually finds the global minimum close to SKH, while ABC, BA, CS, DE, ES, GA, HS, PBIL, PSO and SA fail to search the global minimum within the limited iterations. Here, all the algorithms show the almost same starting point; however SKH outperforms them with fast and stable convergence rate.

Figure 2 shows the results for F04 Penalty #1 function. From Figure 2, although slower later, PSO shows a fastest convergence rate initially among twelve methods; however, it is outperformed by SKH after 6 generations. Furthermore, SKH outperforms all other methods during the whole progress of optimization in this multimodal benchmark function. Eventually, SA performs the second best at finding the global minimum. CS and KH perform the third and the fourth best at finding the global minimum with a relatively slow and stable convergence rate.

Figure 3 shows the performance achieved for F05 Penalty #2 function. For this multimodal function, similar to the F04 Penalty #1 function as shown in Figure 2, SKH is significantly
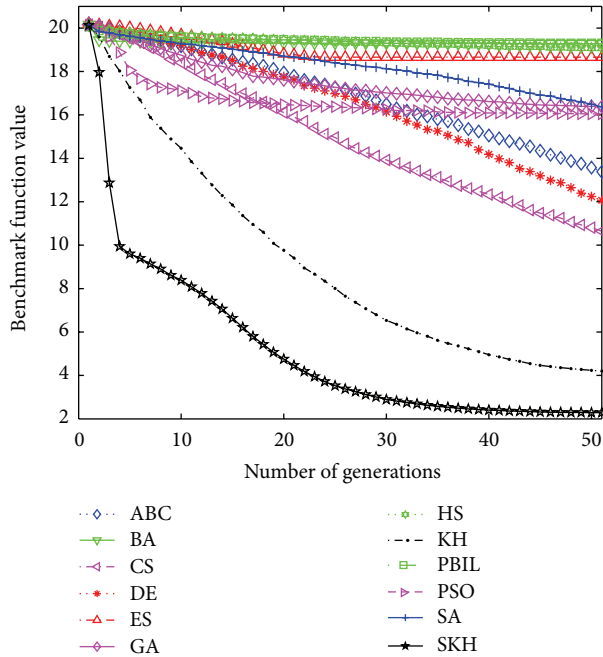
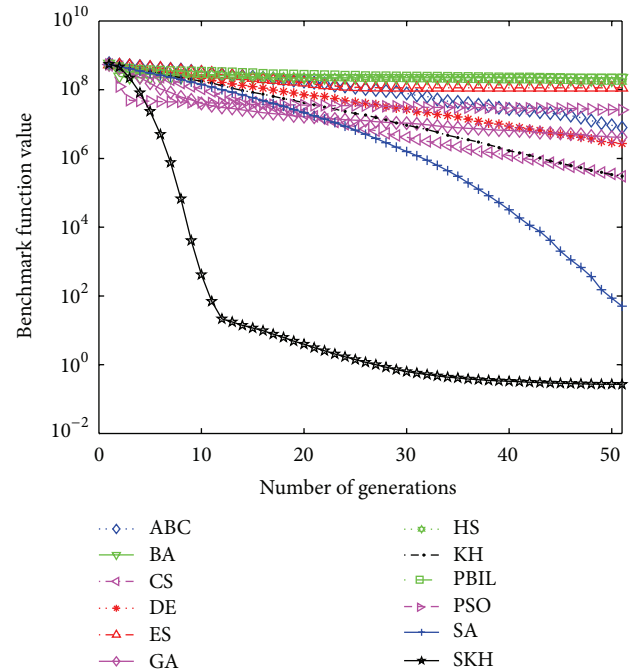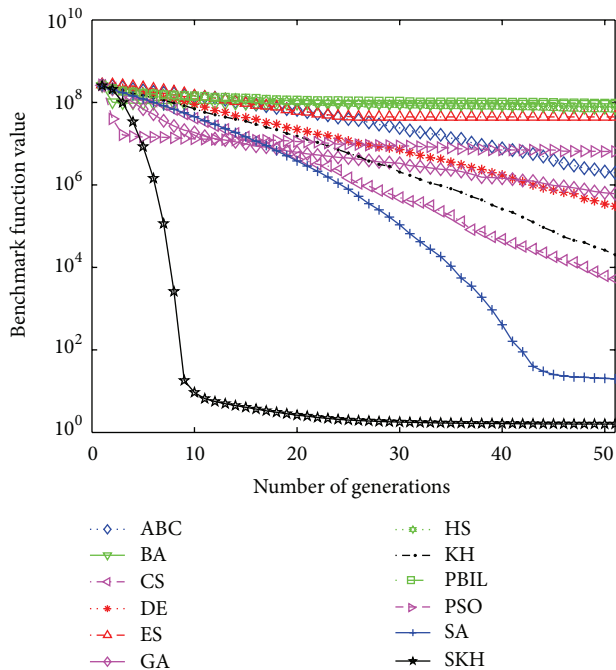FIGURE 1: Comparison of the performance of the different methods for the F01 Ackley function.

FIGURE 2: Comparison of the performance of the different methods for the F04.

FIGURE 3: Comparison of the performance of the different methods for the F05 Penalty.
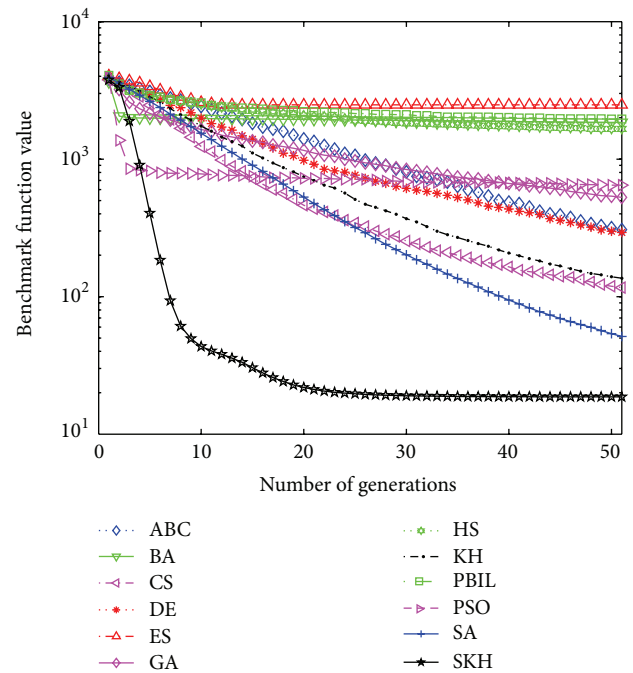

FIGURE 4: Comparison of the performance of the different methods for the F08 Rosenbrock function.

superior to all the other algorithms during the process of optimization. SA performs the second best at finding the global minimum. For other algorithms, the figure shows that there is little difference between the performance of CS and KH. However, carefully studying Table 1 and Figure 3, we can conclude that KH performs slightly better than CS in this multimodal function.

Figure 4 shows the results for F08 Rosenbrock function. From Figure 4, very clearly, SKH has the fastest convergence rate at finding the global minimum and significantly outperforms all other approaches. Looking carefully at Figure 4, SA is only inferior to SKH and performs the second best in this unimodal function. In addition, CS and KH perform very
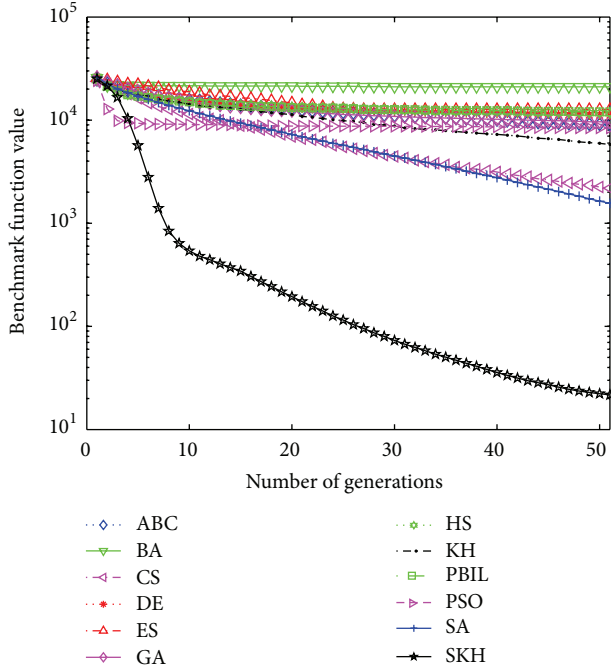
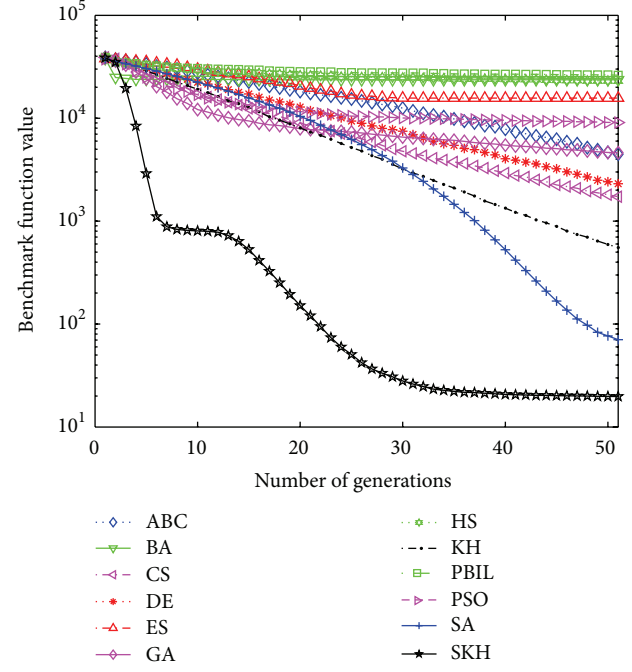FIGURE 5: Comparison of the performance of the different methods for the F10.

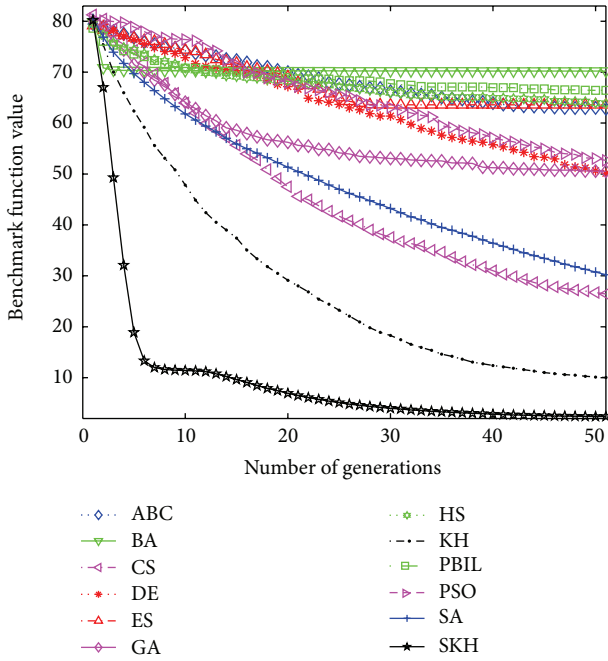FIGURE 7: Comparison of the performance of the different methods for the F14 Step function.

FIGURE 6: Comparison of the performance of the different methods for the F12 Schwefel.

well and have ranks of 3 and 4, respectively. Furthermore, PSO has a fast convergence initially towards the known minimum; however, it is outperformed by SKH after 7 generations. ABC, BA, DE, ES, GA, HS, PBIL, and PSO do not manage to succeed in this benchmark function within

maximum number of generations, showing a wide range of obtained results.

Figure 5 shows the results for F10 Schwefel 1.2 function. From Figure 5, we can see that SKH performs far better than other algorithms during the optimization process in this relative simple unimodal benchmark function. PSO shows a faster convergence rate initially than SKH; however, it is outperformed by SKH after 5 generations. CS and KH perform very well and rank 3 and 4, respectively.

Figure 6 shows the results for F12 Schwefel 2.21 function. Very clearly, SKH has the fastest convergence rate at finding the global minimum and significantly outperforms all other approaches. For other algorithms, KH, CS, and SA perform very well and have ranks of 2, 3 and 4, respectively. Particularly, KH is only inferior to SKH and eventually converges to the value that is very close to CSKH.

Figure 7 shows the results for F14 Step function. Apparently, SKH shows the fastest convergence rate at finding the global minimum and significantly outperforms all other approaches. Here, all the algorithms show the almost same starting point; however SKH outperforms the other algorithms with fast and stable convergence rate. For other algorithms, SA performs the second best among 12 methods. Furthermore, ABC, BA, CS, DE, ES, GA, HS, KH, PBIL, and PSO do not manage to succeed in this benchmark function within maximum number of generations, showing a wide range of obtained results.

From previous analyses about Figures 1–7, we can arrive at a conclusion that our proposed metaheuristic SKH method significantly outperforms the other eleven approaches. In general, SA is only inferior to SKH and performs the second best among twelve methods. CS and KH perform the third

best only inferior to the SA and SKH. Furthermore, the illustration of benchmarks F04, F05, F08, and F10 shows that PSO has a faster convergence rate initially, while later it converges slower and slower to the true objective function value.

4.2. Discussion. For all the benchmark functions considered here, it has been demonstrated that the proposed SKH method is superior to, or at least highly competitive with, the standard KH and other eleven great state-of-the-art population-based methods. The advantages of SKH involve implementing simply and easily and have few parameters to regulate. The work carried out here proves the SKH to be effective, robust, and efficient over a variety of benchmark functions.

Benchmark evaluation is a good way for testing the effectiveness of the metaheuristic methods, but it also has some limitations [2]. First, we made little effort to carefully adjust the optimization methods used in this paper. In most cases, different adjusting parameters in the optimization methods might generate great differences in their performance. Second, real-world optimization problems may not have much of a relationship with benchmark functions. Third, benchmark evaluation may reach significantly different conclusions if the grading criteria or problem setup change. In our work, we securitized the average and best results achieved with some population size and after some number of iterations. However, we might arrive at different conclusions if we change the population size, or consider how many population sizes it needs to reach a certain function value or if we modify the iteration. Regardless of these caveats, the experimental results obtained here are promising for SKH and show that this novel method might be able to find a niche among the plethora of optimization approaches [2].

It is worth pointing out that time requirement is a limitation to the implementation of most optimization methods. If an approach does not has a fast converge, it will be infeasible, since it would take much time to get an optimal or suboptimal solution. SKH seems not to require an impractical amount of CPU time; of the twelve optimization approaches discussed in this paper, SKH was the ninth fastest. How to speed up the SKH's convergence is still worthy of further study.

In our study, 14 benchmark functions are applied to prove the merits of our method; we will verify our proposed method on more optimization problems, such as the high-dimensional ($d \geq 20$) CEC 2010 test suit [42] and the real-world engineering problems. Moreover, we will compare SKH with other optimization methods. In addition, we only look at the unconstrained function optimization here. Our future work includes adding more optimization techniques into SKH for constrained optimization problems, such as constrained real-parameter optimization CEC 2010 test suit [43].

5. Conclusion and Future Work

Due to the limited performance of KH on complex problems [2], KS operator has been introduced to the standard KH to develop a novel improved metaheuristic optimization method based on KH and SA, called simulated annealing-based krill herd (SKH) algorithm, for optimization problem. In SKH, firstly, we use a standard KH algorithm to select a good candidate solution set. And then, a new krill selecting (KS) operator is introduced into the basic KH method. The introduced KS operator includes greedy strategy and accepting a not-so-good solution with a small probability. This greedy strategy is applied to accept a good candidate solution so as to improve its efficiency and reliability for solving global numerical optimization problem. Furthermore, the KS operator not only accepts changes that improve the objective function, but also keeps some changes in not-so-good solutions with a low probability that are not ideal. This can enhance the adversity of the population, improve the exploration of KH, and evade premature convergence.

Furthermore, we can see that this new method can speed up the global convergence rate without losing the strong robustness of the basic KH. From the analyses of the experimental results, we observe that the SKH greatly improves the reliability of the global optimality and they also improve the quality of the solutions on unimodal and most multimodal problems. In addition, the SKH is simple and implements easily.

In the field of numerical optimization, there are numerous issues that deserve further scrutiny. Our future work will emphasize the two issues. On the one hand, we would employ our proposed SKH method to solve real-world engineering optimization problems, and, obviously, SKH can be a great method for real-world engineering optimization problems. On the other hand, we would develop more new metaheuristic methods to solve optimization problems more accurately and efficiently [2].

Acknowledgments

References

[1] G. Wang and L. Guo, "A novel hybrid bat algorithm with harmony search for global numerical optimization," *Journal of Applied Mathematics*, vol. 2013, Article ID 696491, 21 pages, 2013.

[2] G. Wang, L. Guo, A. H. Gandomi et al., "Lévy-flight krill herd algorithm," *Mathematical Problems in Engineering*, vol. 2013, Article ID 682073, 14 pages, 2013.

[3] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, Frome, UK, 2nd edition, 2010.

[4] R.-M. Chen and C.-M. Wang, "Project scheduling heuristics-based standard PSO for task-resource assignment in heterogeneous grid," *Abstract and Applied Analysis*, vol. 2011, Article ID 589862, 20 pages, 2011.

[5] W. Y. Zhang, S. Xu, and S. J. Li, "Necessary conditions for weak sharp minima in cone-constrained optimization problems,"

Abstract and Applied Analysis, vol. 2012, Article ID 909520, 11 pages, 2012.

[6] H. Duan, W. Zhao, G. Wang, and X. Feng, "Test-sheet composition using analytic hierarchy process and hybrid metaheuristic algorithm TS/BBO," *Mathematical Problems in Engineering*, vol. 2012, Article ID 712752, 22 pages, 2012.

[7] X. S. Yang, A. H. Gandomi, S. Talatahari, and A. H. Alavi, *Metaheuristics in Water, Geotechnical and Transport Engineering*, Elsevier, Waltham, Mass, USA, 2013.

[8] A. H. Gandomi, X. S. Yang, S. Talatahari, and A. H. Alavi, *Metaheuristic Applications in Structures and Infrastructures*, Elsevier, Waltham, Mass, USA, 2013.

[9] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, New York, NY, USA, 1998.

[10] X. S. Yang and A. H. Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations*, vol. 29, no. 5, pp. 464–483, 2012.

[11] H. Chen, Y. Zhu, and K. Hu, "Adaptive bacterial foraging optimization," *Abstract and Applied Analysis*, vol. 2011, Article ID 108269, 27 pages, 2011.

[12] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[13] T. S. J. Laseetha and R. Sukanesh, "Investigations on the synthesis of uniform linear antenna array using biogeography-based optimisation techniques," *International Journal of Bio-Inspired Computation*, vol. 4, no. 2, pp. 119–130, 2012.

[14] M. R. Lohokare, S. Devi, S. S. Pattnaik, B. K. Panigrahi, and J. G. Joshi, "Modified biogeography-based optimisation (MBBO)," *International Journal of Bio-Inspired Computation*, vol. 3, no. 4, pp. 252–266, 2011.

[15] A. Hamdi and A. A. Mukheimer, "Modified Lagrangian methods for separable optimization problems," *Abstract and Applied Analysis*, vol. 2012, Article ID 471854, 20 pages, 2012.

[16] X. Cai, S. Fan, and Y. Tan, "Light responsive curve selection for photosynthesis operator of APOA," *International Journal of Bio-Inspired Computation*, vol. 4, no. 6, pp. 373–379, 2012.

[17] L. Xie, J. Zeng, and R. A. Formato, "Selection strategies for gravitational constant G in artificial physics optimisation based on analysis of convergence properties," *International Journal of Bio-Inspired Computation*, vol. 4, no. 6, pp. 380–391, 2012.

[18] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[19] Y. Gao and J. Liu, "Multiobjective differential evolution algorithm with multiple trial vectors," *Abstract and Applied Analysis*, vol. 2012, Article ID 172041, 12 pages, 2012.

[20] A. H. Gandomi and A. H. Alavi, "Multi-stage genetic programming: a new strategy to nonlinear system modeling," *Information Sciences*, vol. 181, no. 23, pp. 5227–5239, 2011.

[21] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, December 1995.

[22] S. Gholizadeh and F. Fattahi, "Design optimization of tall steel buildings by a modified particle swarm algorithm," *The Structural Design of Tall and Special Buildings*, 2012.

[23] S. Talatahari, M. Kheirollahi, C. Farahmandpour, and A. H. Gandomi, "A multi-stage particle swarm for optimum design of truss structures," *Neural Computing and Applications*, 2012.

[24] C. Yang and D. Simon, "A new particle swarm optimization technique," in *Proceedings of the 18th International Conference on Systems Engineering (ICSEng '05)*, pp. 164–169, August 2005.

[25] A. I. Selvakumar and K. Thanushkodi, "A new particle swarm optimization solution to nonconvex economic dispatch problems," *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 42–51, 2007.

[26] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013.

[27] A. H. Gandomi and A. H. Alavi, "Krill herd: a new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831–4845, 2012.

[28] G. Wang, L. Guo, H. Wang, H. Duan, L. Liu, and J. Li, "Incorporating mutation scheme into krill herd algorithm for global numerical optimization," *Neural Computing and Applications*, 2012.

[29] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[30] S. M. Chen, A. Sarosh, and Y. F. Dong, "Simulated annealing based artificial bee colony algorithm for global numerical optimization," *Applied Mathematics and Computation*, vol. 219, no. 8, pp. 3575–3589, 2012.

[31] M. A. Tawhid, "Solution of nonsmooth generalized complementarity problems," *Journal of the Operations Research Society of Japan*, vol. 54, no. 1, pp. 12–24, 2011.

[32] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.

[33] X. Li, J. Wang, J. Zhou, and M. Yin, "A perturb biogeography based optimization with mutation for global numerical optimization," *Applied Mathematics and Computation*, vol. 218, no. 2, pp. 598–609, 2011.

[34] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[35] X. S. Yang and S. Deb, "Engineering optimisation by cuckoo search," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 1, no. 4, pp. 330–343, 2010.

[36] H.-G. Beyer, *The Theory of Evolution Strategies*, Springer, Berlin, Germany, 2001.

[37] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[38] B. Shumeet, "Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning," Tech. Rep. CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, Pa, USA, 1994.

[39] Z. Cui, F. Gao, Z. Cui, and J. Qu, "A second nearest-neighbor embedded atom method interatomic potential for Li–Si alloys," *Journal of Power Sources*, vol. 207, pp. 150–159, 2012.

[40] Z. Cui, F. Gao, Z. Cui, and J. Qu, "Developing a second nearest-neighbor modified embedded atom method interatomic potential for lithium," *Modelling and Simulation in Materials Science and Engineering*, vol. 20, no. 1, Article ID 015014, 2011.

[41] G. Wang and L. Guo, "Hybridizing harmony search with biogeography based optimization for global numerical optimization," *Journal of Computational and Theoretical Nanoscience*. In press.

[42] K. Tang, X. Li, P. N. Suganthan, Z. Yang, and T. Weise, "Benchmark functions for the CEC 2010 special session and

competition on large scale global optimization," Tech. Rep., Nature Inspired Computation and Applications Laboratory, USTC, Hefei, China, 2010.

[43] R. Mallipeddi and P. Suganthan, "Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization," Tech. Rep., Nanyang Technological University, Singapore, 2010.