Research Article

A Free Search Krill Herd Algorithm for Functions Optimization

Liangliang Li, Yongquan Zhou, and Jian Xie

College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China

Correspondence should be addressed to Yongquan Zhou; yongquanzhou@126.com

Received 5 April 2014; Revised 13 May 2014; Accepted 21 May 2014; Published 19 June 2014

Academic Editor: Yang Xu

Copyright © 2014 Liangliang Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To simulate the freedom and uncertain individual behavior of krill herd, this paper introduces the opposition based learning (OBL) strategy and free search operator into krill herd optimization algorithm (KH) and proposes a novel opposition-based free search krill herd optimization algorithm (FSKH). In FSKH, each krill individual can search according to its own perception and scope of activities. The free search strategy highly encourages the individuals to escape from being trapped in local optimal solution. So the diversity and exploration ability of krill population are improved. And FSKH can achieve a better balance between local search and global search. The experiment results of fourteen benchmark functions indicate that the proposed algorithm can be effective and feasible in both low-dimensional and high-dimensional cases. And the convergence speed and precision of FSKH are higher. Compared to PSO, DE, KH, HS, FS, and BA algorithms, the proposed algorithm shows a better optimization performance and robustness.

1. Introduction

As many optimization problems cannot be solved by the traditional mathematical programming methods, the metaheuristic algorithms have been widely used to obtain global optimum solutions. And the aim of developing modern metaheuristic algorithms is to increase the accessibility of the global optimum. Inspired by nature, many successful algorithms are proposed, for example, Genetic Algorithm (GA) [1], Particle Swarm Optimization (PSO) [2, 3], Ant Colony Optimization (ACO) [4], Differential Evolution (DE) [5], Harmony Search (HS) [6], Artificial Bee Colony Optimization (ABC) [7], Firefly Algorithm (FA) [8], Artificial Fish Swarm Algorithm (AFSA) [9], Cuckoo Search (CS) [10, 11], Monkey Algorithm (MA) [12], Bat Algorithm (BA) [13], Charged System Search (CSS) [14], and Flower Pollination Algorithm (FPA) [15]. Nature-inspired algorithms can effectively solve the problems which traditional methods cannot solve and have shown excellent performance in many respects. So its application scope has been greatly expanded. In recent years, the metaheuristic algorithms mentioned above have been applied to solve the application problems. For example, Xu et al. (2010) solve the UCAV path planning problems by chaotic artificial bee colony approach [16]. Hasançebi et al. (2013) applied the bat algorithm in structural

optimization problems [17]. Askarzadeh (2013) developed a discrete harmony search algorithm for size optimization of wind-photovoltaic hybrid energy system [18]. Basu and Chowdhury (2013) used cuckoo search algorithm in economic dispatch [19].

Based on the simulation of the herding behavior of krill individuals, Gandomi and Alavi proposed the krill herd algorithm (KH) in 2012 [20]. And KH algorithm is a novel biologically inspired algorithm to solve the optimization problems. In KH, the time-dependent position of the krill individuals is formulated by three main factors: (1) motion induced by the presence of other individuals; (2) foraging motion; (3) physical diffusion. Only time interval (C_t) should be fine-tuned in the KH algorithm which is a remarkable advantage in comparison with other nature-inspired algorithms. Therefore, it can be efficient for many optimization and engineering problems. To improve the krill herd algorithm, Wang and Guo (2013) proposed a hybrid krill herd algorithm with differential evolution for global numerical optimization [21]. The introduced HDE operator let the krill perform local search within the defined region. And the optimization performance of the DEKH was better than the KH. Then, in order to accelerate convergence speed, thus making the approach more feasible for a wider range of real-world engineering applications while keeping

the desirable characteristics of the original KH, an effective Lévy-Flight KH (LKH) method was proposed by Wang et al. in 2013 [22]. And Wang also proposed a new improved metaheuristic simulated annealing-based krill herd (SKH) method for global optimization tasks [23]. The KH algorithm has been applied to solve some application problems. In 2014, a discrete Krill Herd Algorithm was proposed for graph based network route optimization by Sur [24]. KH algorithm was further validated against various engineering optimization problems by Gandomi et al. [25]. Inspired from the animals' behavior, free search (FS) [26] is firstly proposed by Penev and Littlefair. In FS, each animal has original peculiarities called sense and mobility. And each animal can operate either with small precise steps for local search or with large steps for global exploration. Moreover, the individual decides how to search personally.

In order to overcome the limited performance of standard KH on complex problems, a novel free search krill herd algorithm is proposed in this paper. The free search strategy has been introduced into the standard KH to avoid all krill individuals getting trapped into the local optima. The proposed algorithm can greatly enrich the diversity of krill population and improve the calculation accuracy, which leads to a good optimization performance. What is more, the new method can enhance the quality of solutions without losing the robustness.

The proposed FSKH algorithm is different from standard KH in two aspects. Firstly, in FSKH, the population of individuals is initialized using opposition based learning (OBL) strategy [27]. By using OBL strategy, the proposed algorithm can make a more uniform distribution of the krill populations. What is more, we can obtain fitter starting candidate solutions even when there is no knowledge about the solutions.

And secondly, the krill can do freedom and uncertain action using free search strategy. In standard KH, krill is influenced by its "neighbors" and the optimal krill, and the sensing distance of each krill is fixed. But in nature, even for the same krill, its sensitivity and range of activities will also change in different environment and different period. The proposed algorithm can simulate this freedom, uncertain individual behavior of the krill. The free search strategy allows nonzero probability for access to any location of the search space and highly encourages the individuals to escape from trapping in local optimal solution.

The remainder of this paper is organized as follows. In the Section 2, the standard krill herd algorithm and free search strategy are described, respectively. In Section 3, the concept of opposition based learning (OBL) strategy is briefly explained. And the proposed algorithm (FSKH) is described in detail. The simulation experiments of the proposed algorithm are presented in Section 4, compared to PSO, DE, KH, HS, FS and BA algorithms. Finally, some remarks and conclusions are provided in Section 5.

2. Preliminary

2.1. Krill Herd Algorithm. Krill herd (KH) is a novel metaheuristic swarm intelligence optimization method for solving

optimization problems, which is based on the simulation of the herding behavior of krill individuals. The time-dependent position of an individual krill in two-dimensional surface is determined by the following three main actions:

- (1) movement induced by other krill individuals;
- (2) foraging activity;
- (3) physical diffusion.

KH algorithm used the Lagrangian model as follows:

$$\frac{dX_i}{dt} = N_i + F_i + D_i, \quad (1)$$

where N_i is the motion induced by other krill individuals; F_i is the foraging motion; and D_i is the physical diffusion of the i th krill individuals.

2.1.1. Motion Induced by Other Krill Individuals. For a krill individual, the motion induced by other krill individuals can be determined as follows:

$$N_i^{\text{new}} = N^{\text{max}} \alpha_i + \omega_n N_i^{\text{old}}, \quad (2)$$

$$\alpha_i = \alpha_i^{\text{local}} + \alpha_i^{\text{target}},$$

where N^{max} is the maximum induced speed, ω_n is the inertia weight of the motion induced in $[0, 1]$, N_i^{old} is the last motion induced, α_i^{local} is the local effect provided by the neighbors, and α_i^{target} is the target direction effect provided by the best krill individual. And the effect of the neighbors can be defined as

$$\alpha_i^{\text{local}} = \sum_{j=1}^{NN} \widehat{K}_{ij} \widehat{X}_{ij},$$

$$\widehat{X}_{ij} = \frac{X_j - X_i}{\|X_j - X_i\| + \varepsilon}, \quad (3)$$

$$\widehat{K}_{ij} = \frac{K_j - K_i}{K^{\text{worst}} - K^{\text{best}}},$$

where K_i is the fitness value of the i th krill individual. K^{best} and K^{worst} are the best and worst fitness values of the krill individuals so far. K_j is the fitness of j th ($j = 1, 2, \dots, NN$) neighbor. And NN is the number of the neighbors. X represents the related positions.

The sensing distance for each krill individual is determined as follows:

$$d_{s,i} = \frac{1}{5N} \sum_{j=1}^N \|X_i - X_j\|, \quad (4)$$

where N is the number of the krill individuals.

The effect of the individual krill with the best fitness on the i th individual krill is taken into account using

$$\partial_i^{\text{target}} = C^{\text{best}} \widehat{K}_{i,\text{best}} \widehat{X}_{i,\text{best}}, \quad (5)$$

where the value of C^{best} is defined as

$$C^{\text{best}} = 2 \left(\text{rand} + \frac{I}{I_{\text{max}}} \right). \quad (6)$$

2.1.2. Foraging Motion. The foraging motion is formulated in terms of two main effective parameters. The first is the food location and the second one is the previous experience about the food location. This motion can be expressed for the i th krill individual as follows:

$$\begin{aligned} F_i &= V_f \beta_i + \omega_f F_i^{\text{old}}, \\ \beta_i &= \beta_i^{\text{food}} + \beta_i^{\text{best}}, \end{aligned} \quad (7)$$

where V_f is the foraging speed, ω_f is the inertia weight of the foraging motion between 0 and 1, and F_i^{old} is the last foraging motion. β_i^{food} is the attraction of the food and β_i^{best} is the effect of the best fitness of the i th krill so far. In our paper, we set $V_f = 0.02$.

In KH, the virtual center of food concentration is approximately calculated according to the fitness distribution of the krill individuals, which is inspired from “center of mass.” The center of food for each iteration is formulated as follows:

$$X^{\text{food}} = \frac{\sum_{i=1}^N (1/K_i) X_i}{\sum_{i=1}^N (1/K_i)}. \quad (8)$$

Therefore, the food attraction for the i th krill individual can be determined as follows:

$$\beta_i^{\text{food}} = C^{\text{food}} \widehat{K}_{i,\text{food}} \widehat{X}_{i,\text{food}}, \quad (9)$$

where C^{food} is the food coefficient defined as follows:

$$C^{\text{food}} = 2 \left(1 - \frac{I}{I_{\text{max}}} \right). \quad (10)$$

The effect of the best fitness of the i th krill individual is also handled using the following equation:

$$\beta_i^{\text{best}} = \widehat{K}_{i,\text{best}} \widehat{X}_{i,\text{best}}, \quad (11)$$

where $\widehat{K}_{i,\text{best}}$ is the best previously visited position of the i th krill individual.

2.1.3. Physical Diffusion. The random diffusion of the krill individuals can be considered to be a random process in essence. This motion can be described in terms of a maximum diffusion speed and a random directional vector. It can be indicated as follows:

$$D_i = D^{\text{max}} \delta, \quad (12)$$

where D^{max} is the maximum diffusion speed, and δ is the random directional vector, and its arrays are random values in $[-1, 1]$. The better the position is, the less random the motion is. The effects of the motion induced by other krill individuals and foraging motion gradually decrease with increasing the time (iterations). Thus, another term equation (12) into equation (13). This term linearly decreases the random speed with the time and performs on the basis of a geometrical annealing schedule:

$$D_i = D^{\text{max}} \left(1 - \frac{I}{I_{\text{max}}} \right) \delta. \quad (13)$$

2.1.4. Crossover. The crossover is controlled by a crossover probability Cr , and the m th component of X_i (i.e., $x_{i,m}$) is defined as follows:

$$x_{i,m} = \begin{cases} x_{r,m} & \text{rand}_{i,m} < \text{Cr} \\ x_{i,m} & \text{else,} \end{cases} \quad (14)$$

$$\text{Cr} = 0.2 \widehat{K}_{i,\text{best}},$$

where $r \in \{1, 2, \dots, i-1, i+1, \dots, N\}$.

2.1.5. Mutation. The mutation is controlled by a mutation probability Mu , and the adaptive mutation scheme is determined as follows:

$$x_{i,m} = \begin{cases} x_{g\text{best},m} + \mu (x_{p,m} - x_{q,m}) & \text{rand}_{i,m} < \text{Mu} \\ x_{i,m} & \text{else,} \end{cases} \quad (15)$$

$$\text{Mu} = 0.05 \widehat{K}_{i,\text{best}},$$

where $p, q \in \{1, 2, \dots, i-1, i+1, \dots, K\}$, $\mu \in [0, 1]$.

2.1.6. Main Procedure of the Krill Herd Algorithm. In general, the defined motions frequently change the position of a krill individual toward the best fitness. The foraging motion and the motion induced by other krill individuals contain two global and two local strategies. These are working in parallel which make KH a powerful algorithm. Using different effective parameters of the motion during the time, the position vector of a krill individual during the interval t to $t + \Delta t$ is given by the following equation:

$$X_i(t + \Delta t) = X_i(t) + \Delta t \frac{dX_i}{dt}. \quad (16)$$

It should be noted that Δt is one of the most important constants and should be carefully set according to the optimization problem. This is because this parameter works as a scale factor of the speed vector. Δt can be simply obtained from the following formula:

$$\Delta t = C_t \sum_{j=1}^{\text{NV}} (\text{UB}_j - \text{LB}_j), \quad (17)$$

where NV is the total number of variables and LB_j , UB_j are lower and upper bounds of the j th variables ($j = 1, 2, \dots, \text{NV}$), respectively. C_t is a constant number between $[0, 2]$ (Algorithm 1).

2.2. Free Search Operator. Free search (FS) [26] models the behavior of animals and operates on a set of solutions called population. In the algorithm each animal has original peculiarities called sense and mobility. The sense is an ability of the animal for orientation within the search space. The animal uses its sense for selection of location for the next step. Different animals can have different sensibilities. It also varies during the optimization process, and one animal can have different sensibilities before different walks. The animal can select, for start of the exploration walk, any location

FIGURE 1: Flowchart of the free search krill algorithm.

FIGURE 2: Convergence curves for F_1 ($D = 30$). FIGURE 3: Convergence curves for F_2 ($D = 30$).

FIGURE 4: Convergence curves for F_3 ($D = 30$).

marked with pheromone, which fits its sense. During the exploration walk the animals step within the neighbor space. The neighbor space also varies for the different animals. Therefore, the probability for access to any location of the search space is nonzero.

During the exploration, each krill achieves some favor (an objective function solution) and distributes a pheromone in amount proportional to the amount of the found favor (the quality of the solution). The pheromone is fully replaced with a new one after each walk.

Particularly, the animals in the algorithm are mobile. Each animal can operate either with small precise steps for local search or with large steps for global exploration. And each animal decides how to search (with small or with large steps) by itself. Explicit restrictions do not exist. The previous experience can be taken into account, but it is not compulsory.

2.2.1. The Structure of the FS Operator. The structure of the algorithm consists of three major events: initialization, exploration, and termination.

(1) *Initialization.* In this paper, the initialization strategy is $x_{0ji} = x_{i\min} + (x_{i\max} - x_{i\min}) * \text{random}_{tji}(0, 1)$, where $\text{random}(0, 1)$ is a random value in $[0, 1]$ and $x_{i\min}$, $x_{i\max}$ are the space borders.

(2) *Exploration.* The exploration walk generates coordinates of a new location x_{tji} as

$$x_{tji} = x_{0ji} - \Delta x_{tji} + 2 * \Delta x_{tji} * \text{random}_{tji}(0, 1). \quad (18)$$

The modification strategy is

$$\Delta x_{tji} = R_j * (x_{i\max} - x_{i\min}) * \text{random}_{tji}(0, 1), \quad (19)$$

where T is step limit per walk and $t = 1, \dots, T$ is current step.

FIGURE 5: Convergence curves for F_4 ($D = 30$).

The individual behavior, during the walk, is modeled and described as $f_{tj} = f(x_{tji})$, $f_j = \max(f_{tj})$, where f_j is the only location marked with pheromone from an animal after the walk.

The pheromone generation is

$$P_j = \frac{f_j}{\max(f_j)}. \quad (20)$$

The sensibility generation is

$$S_j = S_{\min} + \Delta S_j, \quad (21)$$

$$\Delta S_j = (S_{\max} - S_{\min}) * \text{random}_j(0, 1),$$

where $S_{\min}$ and $S_{\max}$ are minimal and maximal possible values of the sensibility. $P_{\min}$ and $P_{\max}$ are minimal and maximal possible values of the pheromone trials. And $P_{\max} = S_{\max}$, $P_{\min} = S_{\min}$.

Selection and decision making for a start location x'_{0ji} for an exploration walk is

$$x'_{0ji} = \begin{cases} x'_{0ji}, & (P_k < S_j) \\ x_{ji}, & (P_k \geq S_j), \end{cases} \quad (22)$$

where $j = 1, \dots, NK$, $k = 1, \dots, NK$, and k is the marked locations number.

(3) *Termination.* In this paper, the criterion for termination is $\text{iter} > \text{iterMax}$, where iterMax is the maximum number of iterations.

The steps are imitation of the ability for motion and action. The steps can be large or small and can vary. In the search process, the neighbor space is a tool for tuning rough and precise searches. So, search radius R_j is a parameter related to individual j search space; the value of R_j decides the optimization quality during the search process.

FIGURE 6: Convergence curves for F_5 ($D = 30$). FIGURE 7: Convergence curves for F_6 ($D = 30$).

There are two methods to set the value of search radius. The first one is that R_j is constant. If the value is higher, the individual search space is wider, search time is longer, and the convergence precision is lower. If the value is lower, the individual search space is smaller and the convergence precision is higher.

The second method is that changing neighbor space R_j adaptively. And during the search process, R_j is decreasing gradually. The rule is as follows:

$$R_j(t) = \rho * R_j(t-1), \quad (23)$$

where t is the exploration generation and $0 \leq \rho \leq 1$ is the radius contract coefficient, which is an important parameter. In this paper, we adopt the first approach.

FIGURE 8: Convergence curves for F_7 ($D = 30$). FIGURE 9: Convergence curves for F_8 ($D = 30$).

2.2.2. Procedure of FS Operator. The detailed process of FS operator is described as in Algorithm 2.

3. Free Search krill Herd Algorithm

In KH algorithm, krill is influenced by its “neighbors” and the optimal krill, and the sensing distance of each krill is fixed. But in nature, the action of each krill is free and uncertain. In order to simulate this freedom, uncertain individual behavior of the krill, this paper introduces the free search strategy into

FIGURE 10: Convergence curves for F_9 ($D = 30$). FIGURE 11: Convergence curves for F_{10} ($D = 30$).

the krill herd algorithm and proposes a novel free search krill herd algorithm (FSKH).

3.1. Opposition-Based Population Initialization. Population initialization has an important impact on the optimization results and global convergence; this paper introduces the initialization method of opposition based learning (OBL) strategy [27] to generate initial krill populations (Algorithm 3).

By utilizing OBL we can obtain fitter starting candidate solutions even when there is no knowledge about the solutions. This initialization method can make a more uniform

FIGURE 12: Convergence curves for F_{11} ($D = 2$). FIGURE 13: Convergence curves for F_{12} ($D = 2$).

distribution of the krill populations. Therefore, it is good for the method to get better optimization results. And by utilizing free search strategy, each krill individual in FSKH can decide how to search by itself (Algorithm 4). The strategy allows nonzero probability to approach to any location of the search space and highly encourages the individuals to escape from trapping in local optimal solution. During the search process, each krill takes exploration walks according to different search radius.

In general, three main actions in standard KH algorithm can guide the krill individuals to search the promising

TABLE 1: Benchmark functions.

Category	Number	Name	Benchmark function	Scope	$f_{\min}$
I	F_1	Sphere	$f(x) = \sum_{i=1}^n x_i^2$	$[-5.12, 5.12]$	0
	F_2	Step	$f(x) = \sum_{i=1}^{n-1} ([x_i + 0.5])^2$	$[-100, 100]$	0
	F_3	Rosenbrock	$f(x) = \sum_{i=1}^{n-1} [(x_i - 1)^2 + 100(x_i^2 - x_{i+1})^2]$	$[-2.048, 2.048]$	0
	F_4	Quartic	$f(x) = \sum_{i=1}^n x_i^4 + \text{random}[0, 1]$	$[-1.28, 1.28]$	0
	F_5	Rastrigin	$f(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	0
	F_6	Ackley	$f(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i\right)\right) + 20 + e$	$[-32.768, 32.768]$	0
	F_7	Quadric	$f(x) = \sum_{j=1}^n \left(\sum_{i=1}^j x_i\right)^2$	$[-100, 100]$	0
	F_8	Griewank	$f(x) = \frac{1}{4000} \sum_{i=1}^n (x_i^2) - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	0
	F_9	Alpine	$f(x) = \sum_{i=1}^n x_i \sin(x_i) + 0.1 x_i $	$[-10, 10]$	0
	F_{10}	Zakharov	$f(x) = \sum_{i=1}^n x_i^2 + \left(\sum_{i=1}^n 0.5 i x_i\right)^2 + \left(\sum_{i=1}^n 0.5 i x_i\right)^4$	$[-5, 10]$	0
II	F_{11}	Schaffer's F6	$f(x) = \frac{\sin^2 \sqrt{(x_1^2 + x_2^2)} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2} - 0.5$	$[-100, 100]$	-1
	F_{12}	Drop wave	$f(x) = -\frac{1 + \cos\left(12\sqrt{x_1^2 + x_2^2}\right)}{(1/2)(x_1^2 + x_2^2) + 2}$	$[-5.12, 5.12]$	-1
	F_{13}	Six-hump camel back	$f(x_1, x_2) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$	$[-3, 3]$	-1.0316
	F_{14}	Easom	$f(x) = -(-1)^n \left(\prod_{i=1}^n \cos^2(x_i)\right) \exp\left(-\sum_{i=1}^n (x_i - \pi)\right)^2$	$[-2\pi, 2\pi]$	-1

solution space. But it is easy for the standard KH algorithm to be trapped into local optima, and the performance in high-dimensional cases is unsatisfied. In the FSKH algorithm, the individual can search the promising area with small or large steps. So, the krill individuals can move step by step through multidimensional search space. In nature, the activity range of krill individuals is different. R_{ji} can adjust the activity range of the individual, and there is no explicit restrictions.

Using the free search strategy, the krill individual can search any region of the search space. Each krill individual can search according to their perception and the scope of activities and can not only search around the global optimum, but also search around local optimum. When using larger step, it takes global search which can strengthen the weak global search ability of KH. Therefore, the proposed algorithm has better population diversity and convergence speed and can enhance the global searching ability of the algorithm. To achieve a better balance between local search and global search, FSKH algorithm includes both "exploration" process of FS and "exploitation" process of KH. When increasing the sensitivity, the krill individual will approach the whole population's current best value (i.e., local search). While reducing the sensitivity, the krill individual can search around other neighborhood (i.e., global search) (Figure 1).

4. Simulation Experiments

4.1. Simulation Platform. All the algorithms compared in this section are implemented in Matlab R2012a (7.14). And experiments are performed on a PC with a 3.01 GHz, AMD Athlon (tm) II X4 640 Processor, 3 GB of RAM, and Windows XP operating system. In the tests, population size is $NP = 50$. The experiment results are obtained in 50 trials.

4.2. Benchmark Functions. In order to verify the effectiveness of the proposed algorithm, we select 14 standard benchmark functions [28–30] to detect the searching capability of the proposed algorithm. The proposed algorithm in this paper (i.e., FSKH) is compared with PSO, DE, KH, HS, FS, and BA.

4.3. Parameter Setting. Generally, the choice of parameters requires some experimenting. In this paper, after a lot of experimental comparison, the parameters of the algorithm are set as follows.

In KH and FSKH, the maximum induced speed $N^{\max} = 0.01$, the foraging speed $V_f = 0.02$, and the maximum

FIGURE 14: Convergence curves for F_{13} ($D = 2$). FIGURE 15: Convergence curves for F_{14} ($D = 2$).

diffusion speed $D^{\max} = 0.005$. In FSKH and FS, the search radius

$$R_j = \begin{cases} 1, & j = 1, \dots, \frac{NP}{3} \\ 0.5, & j = \frac{NP}{3} + 1, \dots, \frac{2NP}{3} \\ 0.1, & j = \frac{2NP}{3} + 1, \dots, NP. \end{cases} \quad (24)$$

But in the FSKH, the search step is $T = 5$. In FS, the search step is $T = 50$.

FIGURE 16: ANOVA tests of the global minimum for F_1 ($D = 30$). FIGURE 17: ANOVA tests of the global minimum for F_2 ($D = 30$). FIGURE 18: ANOVA tests of the global minimum for F_3 ($D = 30$).

FIGURE 19: ANOVA tests of the global minimum for F_4 ($D = 30$). FIGURE 22: ANOVA tests of the global minimum for F_7 ($D = 30$). FIGURE 20: ANOVA tests of the global minimum for F_5 ($D = 30$). FIGURE 23: ANOVA tests of the global minimum for F_8 ($D = 30$). FIGURE 21: ANOVA tests of the global minimum for F_6 ($D = 30$). FIGURE 24: ANOVA tests of the global minimum for F_9 ($D = 30$).

Begin

Initialization. Set the generation counter $iter = 1$; initialize the population P of NP krill individuals randomly and each krill corresponds to a potential solution to the given problem; set the foraging speed V_f , maximum diffusion speed $D^{\max}$, and maximum induced speed $N^{\max}$;

Fitness evaluation. Evaluate each krill individual according to its position.

While the termination criterion is not satisfied or $iter < iterMax$ **do**

Sort the population krill from best to worst.

for $i = 1 : NP$ (all krill) **do**

Perform the following motion calculation.

Motion induced by the presence of other individuals

Foraging motion

Physical diffusion

Implement the genetic operators.

Update the krill individual position in the search space.

Evaluate each krill individual according to its new position.

end for

Sort the population krill from best to worst and find the current best.

$iter = iter + 1$.

End While

Post-processing the results and visualization.

End

ALGORITHM 1: Krill herd algorithm.

FIGURE 25: ANOVA tests of the global minimum for F_{10} ($D = 30$).

In BA, the parameters are generally set as follows: pulse frequency range is $Q_i \in [0, 2]$, the maximum loudness is $A^0 = 0.5$, maximum pulse emission is $r^0 = 0.5$, attenuation coefficient of loudness is $\alpha = 0.95$, and increasing coefficient of pulse emission is $\gamma = 0.05$. In PSO, we use linear decreasing inertia weight $\omega_{\max} = 0.9$, $\omega_{\min} = 0.4$, and learning factor is $C_1 = C_2 = 1.4962$. In HS, the harmony consideration rate is $HMCR = 0.9$, the minimum pitch adjusting rate is $PAR_{\min} = 0.4$, the maximum pitch adjusting rate is $PAR_{\max} = 0.9$, the minimum bandwidth is $bw_{\min} = 0.0001$, and the maximum bandwidth is $bw_{\max} = 1$. In DE, the crossover constant is $P_{cr} = 0.5$.

FIGURE 26: ANOVA tests of the global minimum for F_{11} ($D = 2$).

4.4. Comparison of Experiment Results. The best, mean, worst, and Std. represent the optimal fitness value, mean fitness value, worst fitness value, and standard deviation, respectively. Bold and italicized results mean that FSKH is better, while the * results means that other algorithm is better.

For the low-dimensional case, the maximum number of iterations of each algorithm is $iterMax = 500$. As seen from Tables 2 and 3, FSKH provides better results than other algorithms except F_3 , F_{13} , and F_{14} . What is more, FSKH can find the theoretical optimum solutions for nine benchmark functions (F_1 , F_2 , F_5 , $F_7 \sim F_{12}$) and has a very strong robustness. For other algorithms, only PSO can find the theoretical optimum solution for three functions

TABLE 2: Simulation results for $F_1 \sim F_9$ in low dimension.

Benchmark function	Result	PSO	DE	KH II	Method	FS	FS	BA
$F_1 (D = 30)$	Best	$1.8151E - 02$	$1.2075E - 01$	$1.1148E - 04$	0	$1.2072E + 01$	$4.5486E - 02$	$9.4991E - 04$
	Mean	$4.5954E - 01$	$2.4943E - 01$	$1.6110E - 03$	6.1254E - 316	$3.4612E + 01$	$5.9487E - 02$	$1.4434E - 03$
	Worst	$1.6052E + 00$	$3.7078E - 01$	$6.4270E - 03$	1.1599E - 314	$5.1108E + 00$	$7.0424E - 02$	$1.7796E - 03$
	Std.	$3.2409E - 01$	$6.1704E - 02$	$1.6590E - 03$	0	$1.7649E + 01$	$6.7912E - 03$	$1.8339E - 04$
	Rank	6	5	3	1	7	4	2
$F_2 (D = 30)$	Best	$7.9000E + 01$	$2.1000E + 01$	0	0	$4.0710E + 03$	$1.6000E + 01$	$3.2450E + 04$
	Mean	$2.4706E + 02$	$8.8740E + 01$	$2.8200E + 00$	0	$7.8597E + 03$	$2.3780E + 01$	$5.9598E + 04$
	Worst	$6.2900E + 02$	$2.7900E + 02$	$8.4000E + 01$	0	$1.8688E + 03$	$3.0000E + 01$	$7.2507E + 04$
	Std.	$1.2618E + 02$	$5.3506E + 01$	$1.1845E + 01$	0	$7.1600E + 03$	$2.9017E + 00$	$6.9512E + 03$
	Rank	5	4	3	1	7	2	6
$F_3 (D = 30)$	Best	$2.8953E + 01$	$3.0199E + 01$	$2.6378E + 01$	$2.8703E + 01$	$3.1922E + 02$	$2.2637E + 01$	1.0961E + 01
	Mean	$6.2541E + 01$	$3.6796E + 01$	$2.7980E + 01$	$2.8912E + 01$	$5.8759E + 02$	2.6906E + 01	$3.0216E + 01$
	Worst	$4.5307E + 02$	$4.9489E + 01$	$2.9430E + 01$	2.8991E + 01	$1.0549E + 03$	$2.9389E + 01$	$8.3654E + 01$
	Std.	$6.0248E + 01$	$3.7560E + 00$	$8.1537E - 01$	6.1287E - 02	$1.5737E + 02$	$1.4196E + 00$	$1.3356E + 01$
	Rank	5	6	3	4	7	2	1
$F_4 (D = 30)$	Best	$3.0162E - 02$	$8.6779E - 03$	$1.1579E - 02$	2.2393E - 05	$2.7531E - 01$	$6.3723E - 01$	$8.6847E - 02$
	Mean	$8.0366E - 02$	$2.5195E - 02$	$4.4238E - 02$	7.6115E - 04	$7.3426E - 01$	$9.7518E - 01$	$1.6032E - 01$
	Worst	$1.7465E - 01$	$4.5602E - 02$	$9.5740E - 02$	2.1624E - 03	$1.1753E + 00$	$1.2684E + 00$	$2.4878E - 01$
	Std.	$3.3563E - 02$	$7.4389E - 03$	$2.0059E - 02$	5.0720E - 04	$2.1038E - 01$	$1.3336E - 01$	$3.4310E - 02$
	Rank	4	2	3	1	7	6	5
$F_5 (D = 30)$	Best	$4.3907E + 01$	$1.2231E + 02$	$7.2421E + 00$	0	$8.8691E + 01$	$1.3828E + 02$	$1.4257E + 02$
	Mean	$8.8471E + 01$	$1.6497E + 02$	$2.1243E + 01$	0	$1.3383E + 02$	$1.9791E + 02$	$2.0262E + 02$
	Worst	$1.4702E + 02$	$1.8931E + 02$	$4.0592E + 01$	0	$1.7953E + 02$	$2.4035E + 02$	$2.8786E + 02$
	Std.	$2.3543E + 01$	$1.4949E + 01$	$6.7613E + 00$	0	$1.5616E + 01$	$2.2151E + 01$	$2.8986E + 01$
	Rank	6	3	2	1	4	5	7
$F_6 (D = 30)$	Best	$3.4508E + 00$	$3.1954E + 00$	$8.0527E - 02$	8.8818E - 16	$1.3509E + 01$	$1.9172E + 01$	$1.8667E + 01$
	Mean	$5.8518E + 00$	$4.2090E + 00$	$3.1470E + 00$	8.8818E - 16	$1.4935E + 01$	$1.9908E + 01$	$1.9148E + 01$
	Worst	$8.4440E + 00$	$5.2716E + 00$	$6.8219E + 00$	8.8818E - 16	$1.6955E + 01$	$2.0247E + 01$	$1.9503E + 01$
	Std.	$1.2143E + 00$	$5.3016E - 01$	$1.1229E + 00$	0	$8.6116E - 01$	$2.2907E - 01$	$1.9351E - 01$
	Rank	7	4	6	1	5	3	2
$F_7 (D = 30)$	Best	$3.5384E + 03$	$5.6171E + 01$	$7.0436E + 03$	0	$1.3848E + 05$	$1.4414E + 04$	$1.8780E + 04$
	Mean	$2.1959E + 04$	$1.4336E + 02$	$3.2982E + 04$	4.3250E - 180	$4.6076E + 05$	$2.3871E + 04$	$1.7660E + 05$
	Worst	$1.7426E + 05$	$4.3311E + 02$	$1.7281E + 05$	2.1625E - 178	$1.3967E + 06$	$3.6638E + 04$	$7.5561E + 05$
	Std.	$2.7411E + 04$	$7.5588E + 01$	$2.7450E + 04$	0	$2.4220E + 05$	$5.4373E + 03$	$1.6793E + 05$
	Rank	4	2	5	1	7	3	6
$F_8 (D = 30)$	Best	$1.0973E + 00$	$1.1225E + 00$	$2.0611E - 02$	0	$5.0342E + 01$	$1.1446E + 00$	$3.7618E + 02$
	Mean	$2.0579E + 00$	$1.6510E + 00$	$1.2319E - 01$	0	$7.3791E + 01$	$1.2110E + 00$	$5.3973E + 00$
	Worst	$3.9297E + 00$	$2.5709E + 00$	$5.1307E - 01$	0	$1.2387E + 02$	$1.2616E + 00$	$6.61453177 + 02$
	Std.	$5.5487E - 01$	$3.5593E - 01$	$8.0788E - 02$	0	$1.4831E + 01$	$2.4008E - 02$	$6.4097E + 01$
	Rank	5	4	3	1	6	2	7
$F_9 (D = 30)$	Best	0	$1.9429E - 16$	$1.0908E - 09$	0	$5.3806E - 05$	$2.8243E - 08$	$1.2522E - 09$
	Mean	$9.1221E - 41$	$1.7080E - 13$	$6.3405E - 08$	3.7231E - 168	$3.6484E - 02$	$1.4745E - 06$	$1.2981E - 06$
	Worst	$1.9689E - 39$	$1.1459E - 12$	$2.5349E - 07$	8.9590E - 167	$3.9778E - 01$	$1.3508E - 05$	$2.5807E - 05$
	Std.	$3.4654E - 40$	$2.5917E - 13$	$6.0990E - 08$	0	$6.7720E - 02$	$2.4162E - 06$	$3.7830E - 06$
	Rank	2	3	4	1	7	5	6
$F_{10} (D = 30)$	Best	$4.7818E - 01$	$2.7822E - 01$	$4.0704E - 02$	0	$6.2320E + 02$	$1.6179E - 01$	$3.2702E - 03$
	Mean	$3.5156E + 00$	$9.6176E - 01$	$1.5103E + 00$	8.6510E - 318	$8.5918E + 02$	$2.7912E - 01$	$5.2075E - 03$
	Worst	$3.1501E + 01$	$3.1197E + 00$	$9.2295E + 00$	3.3739E - 316	$1.2293E + 03$	$3.8473E - 01$	$7.4336E - 03$
	Std.	$4.3846E + 00$	$5.1776E - 01$	$1.9643E + 00$	0	$1.0523E + 03$	$4.0208E - 02$	$1.0260E - 03$
	Rank	6	4	5	1	7	3	2

Note: In this paper, both KH and KH II represent the KH with crossover operator.

TABLE 3: Simulation results for $F_{10} \sim F_{13}$.

Benchmark function	Result	Method						
		PSO	DE	KH II	FSKH	HS	FS	BA
$F_{11} (D = 2)$	Best	-1	-9.999984E-01	-9.999997E-01	-1	-9.880465E-01	-9.982745E-01	-9.627759E-01
	Mean	-9.961136E-01	-9.996264E-01	-9.999894E-01	-1	-8.445035E-01	-9.921680E-01	-7.407210E-01
	Worst	-9.902840E-01	-9.941573E-01	-9.999449E-01	-1	-5.830062E-01	-9.902681E-01	-5.482243E-01
	Std.	4.808128E-03	8.981117E-04	1.209463E-05	0	9.752777E-02	2.869464E-03	1.250904E-01
	Rank	2	4	3	1	6	5	7
$F_{12} (D = 2)$	Best	-1	-1	-1	-1	-9.756697E-01	-9.999931E-01	-9.999997E-01
	Mean	-9.961747E-01	-9.999996E-01	-9.821486E-01	-1	-8.443158E-01	-9.983692E-01	-7.368404E-01
	Worst	-9.362453E-01	-9.999938E-01	-9.362453E-01	-1	-5.856036E-01	-9.948806E-01	-2.888120E-01
	Std.	1.529461E-02	1.113912E-06	2.891639E-02	0	9.704192E-02	1.421345E-03	1.667622E-01
	Rank	3	2	4	1	7	6	5
$F_{13} (D = 2)$	Best	-1.031628E+00	-1.031628E+00	-1.031628E+00	-1.031628E+00	-1.031587E+00	-1.031628E+00	-1.031628E+00
	Mean	-1.031628E+00	-1.031628E+00	-1.031628E+00	-1.031510E+00	-9.719758E-01	-1.031628E+00	-9.989818E-01
	Worst	-1.031628E+00	-1.031628E+00	-1.031628E+00	-1.031071E+00	-6.378180E-01	-1.031627E+00	-2.154637E-01
	Std.	6.728967E-16	8.535156E-16	2.176352E-09	1.420660E-04	7.340586E-02	1.353951E-07	1.615586E-01
	Rank	1	2	3	5	7	4	6
$F_{14} (D = 2)$	Best	-1	-1	-1	-1	-9.938758E-01	-9.999999E-01	-1
	Mean	-1	-1	-9.999999E-01	-9.998070E-01	-5.496989E-01	-9.997912E-01	-6.000308E-01
	Worst	-1	-1	-9.999999E-01	-9.995538E-01	-1.236216E-05	-9.984373E-01	-8.743099E-12
	Std.	0	0	2.026218E-09	1.291856E-03	4.099494E-01	2.968278E-04	4.948335E-01
	Rank	1	1	3	4	7	6	5

TABLE 4: Simulation results for $F_1 \sim F_9$ in high dimension.

Benchmark function	Result	PSO	DE	KH II	Method FSKH	HS	FS	BA
$F_1 (D = 1000)$	Best	$1.7539E + 02$	$1.1046E + 02$	$2.2325E + 02$	0	$5.5839E + 03$	$1.9398E + 03$	$4.6377E + 00$
	Mean	$4.2055E + 02$	$1.2990E + 02$	$2.3419E + 02$	4.6082E – 316	$5.8593E + 03$	$2.1829E + 03$	$5.1666E + 00$
	Worst	$8.7083E + 02$	$1.4698E + 02$	$2.4475E + 02$	4.4691E – 315	$6.0746E + 03$	$2.2895E + 03$	$5.7013E + 00$
	Std.	$1.5335E + 02$	$8.4319E + 00$	$8.0663E + 00$	0	$9.3024E + 01$	$7.1875E + 01$	$2.4083E – 01$
	Rank	4	3	5	1	7	6	2
$F_2 (D = 200)$	Best	$1.7420E + 03$	$5.8630E + 03$	$3.8140E + 03$	0	$2.0879E + 05$	$3.6160E + 03$	$5.1312E + 05$
	Mean	$5.6810E + 03$	$7.6012E + 03$	$6.4365E + 03$	0	$2.4108E + 05$	$4.2361E + 03$	$5.5451E + 05$
	Worst	$1.2432E + 04$	$9.4600E + 03$	$9.4830E + 03$	0	$1.3406E + 04$	$4.8140E + 03$	$6.0191E + 05$
	Std.	$2.4260E + 03$	$7.9567E + 02$	$1.2702E + 03$	0	$2.5040E + 05$	$2.8971E + 02$	$1.8030E + 04$
	Rank	2	5	4	1	6	3	7
$F_3 (D = 100)$	Best	$2.2607E + 02$	$1.8958E + 02$	$1.0216E + 02$	9.8070E + 01	$4.8892E + 03$	$1.1008E + 02$	$9.9091E + 01$
	Mean	$1.2251E + 03$	$2.3327E + 02$	$1.1403E + 02$	9.8701E + 01	$6.6060E + 03$	$1.1743E + 02$	$1.0747E + 02$
	Worst	$4.3896E + 03$	$2.8370E + 02$	$1.3766E + 02$	9.8942E + 01	$8.4203E + 03$	$1.2626E + 02$	$1.6127E + 02$
	Std.	$1.1701E + 03$	$2.0992E + 01$	$9.4142E + 00$	2.1724E – 01	$9.0378E + 02$	$2.8202E + 00$	$2.0822E + 01$
	Rank	6	5	3	1	7	4	2
$F_4 (D = 1000)$	Best	$2.9336E + 00$	$2.1517E – 01$	$1.3101E + 00$	6.3543E – 06	$2.6746E + 02$	$7.7482E + 01$	$6.9964E + 00$
	Mean	$2.7344E + 01$	$3.0597E – 01$	$1.5941E + 00$	4.4048E – 04	$2.9365E + 02$	$8.4210E + 01$	$8.4533E + 00$
	Worst	$6.4757E + 01$	$4.0618E – 01$	$2.0862E + 00$	9.8040E – 04	$3.1329E + 02$	$8.9597E + 01$	$9.7956E + 00$
	Std.	$1.3793E + 01$	$4.7306E – 02$	$2.5818E – 01$	3.0041E – 04	$9.2166E + 00$	$3.2374E + 00$	$5.4824E – 01$
	Rank	4	2	3	1	7	6	5
$F_5 (D = 500)$	Best	$2.7456E + 03$	$4.0397E + 03$	$2.0311E + 03$	0	$6.1828E + 03$	$6.3281E + 03$	$3.8981E + 03$
	Mean	$3.9835E + 03$	$4.2471E + 03$	$2.3252E + 03$	0	$6.5147E + 03$	$6.5574E + 03$	$4.2476E + 03$
	Worst	$4.6952E + 03$	$4.4312E + 03$	$2.5811E + 03$	0	$6.7919E + 03$	$6.7360E + 03$	$4.5534E + 03$
	Std.	$4.2126E + 02$	$8.3027E + 01$	$1.2453E + 02$	0	$1.2874E + 02$	$9.1246E + 01$	$1.3686E + 02$
	Rank	3	5	2	1	6	7	4
$F_6 (D = 300)$	Best	$6.3274E + 00$	$8.0299E + 00$	$7.0415E + 00$	8.8818E – 16	$1.9882E + 01$	$2.0898E + 01$	$1.9386E + 01$
	Mean	$8.2684E + 00$	$8.7293E + 00$	$8.6721E + 00$	8.8818E – 16	$2.0099E + 01$	$2.1003E + 01$	$1.9495E + 01$
	Worst	$1.0209E + 01$	$9.4457E + 00$	$1.0111E + 01$	8.8818E – 16	$2.0345E + 01$	$2.1056E + 01$	$1.9589E + 01$
	Std.	$8.7857E – 01$	$3.0761E – 01$	$6.4227E – 01$	0	$7.7479E – 02$	$3.3659E – 02$	$3.5615E – 02$
	Rank	2	4	3	1	6	7	5
$F_7 (D = 100)$	Best	$2.1725E + 05$	$2.7428E + 03$	$4.3171E + 05$	3.8339E – 319	$9.4228E + 06$	$7.1039E + 05$	$7.9254E + 05$
	Mean	$1.5219E + 06$	$5.7789E + 03$	$2.8233E + 06$	2.2442E – 280	$3.0414E + 07$	$1.3329E + 06$	$4.5801E + 06$
	Worst	$6.8641E + 06$	$1.2878E + 04$	$7.0375E + 06$	1.1221E – 278	$7.1585E + 07$	$1.8894E + 06$	$1.6408E + 07$
	Std.	$1.3202E + 06$	$2.5486E + 03$	$1.6109E + 06$	0	$1.3820E + 07$	$2.6259E + 05$	$3.7611E + 06$
	Rank	3	2	4	1	7	5	6
$F_8 (D = 500)$	Best	$4.7748E + 01$	$1.6294E + 02$	$2.1827E + 02$	0	$7.7044E + 03$	$1.0245E + 03$	$1.1878E + 04$
	Mean	$1.5434E + 02$	$2.0567E + 02$	$2.7742E + 02$	0	$8.3385E + 03$	$1.2015E + 03$	$1.3091E + 04$
	Worst	$3.1407E + 02$	$2.4796E + 02$	$3.3449E + 02$	0	$8.9522E + 03$	$1.4065E + 03$	$1.3825E + 04$
	Std.	$5.9349E + 01$	$1.7027E + 01$	$2.6215E + 01$	0	$2.7756E + 02$	$7.9418E + 01$	$3.3184E + 02$
	Rank	2	3	4	1	6	5	7
$F_9 (D = 200)$	Best	0	0	$3.4191E – 10$	0	$8.9839E – 05$	$7.9967E – 09$	$2.4396E – 10$
	Mean	$7.9602E – 78$	$9.7855E – 19$	$5.6362E – 05$	6.1471E – 167	$1.0189E – 02$	$4.9600E – 07$	$3.5023E – 07$
	Worst	$2.6973E – 76$	$1.3878E – 17$	$1.7294E – 03$	2.3564E – 165	$1.1317E – 01$	$2.5864E – 06$	$6.8099E – 06$
	Std.	$3.8349E – 77$	$2.9411E – 18$	$2.5904E – 04$	0	$1.9049E – 02$	$5.7052E – 07$	$1.1021E – 06$
	Rank	2	3	5	1	7	6	4
$F_{10} (D = 300)$	Best	$1.2447E + 02$	$1.2043E + 02$	$5.5973E + 02$	0	$9.0756E + 03$	$2.9821E + 02$	$7.3081E + 03$
	Mean	$3.9115E + 02$	$1.5201E + 02$	$8.4831E + 02$	1.3901E – 317	$1.0803E + 05$	$3.6677E + 02$	$8.3861E + 03$
	Worst	$8.1348E + 02$	$1.9321E + 02$	$1.1894E + 03$	6.4647E – 316	$1.9276E + 06$	$4.7007E + 02$	$9.4975E + 03$
	Std.	$1.8186E + 02$	$1.6427E + 01$	$1.6003E + 02$	0	$3.6921E + 05$	$3.4810E + 01$	$4.8766E + 02$
	Rank	3	2	5	1	7	4	6

Begin

Initialize $x_{i\min}, x_{i\max}, NK, NP, G, x_{0ji}, T, R_{ji}$;
 Take initial walks $f_{tj}(x_{0ji} + \Delta x_t)$;
 Generate the initial pheromone P_k ;
 Distribute the initial pheromone $P_k \rightarrow x_{kp}$;
 Learn the initial achievements $P_k \rightarrow x_{jkP}$;

While ($t \leq T$)

Generate sensibility S_j ;
 Select start locations for a walk $x'_{0j} = x_k(S_j, P_k)$;
 Take exploration walks $f_{tj}(x_{0ji} + \Delta x_t)$;
 Generate a pheromone P_k ;
 Distribute the pheromone $P_k \rightarrow x_{kp}$;
 Learn the achievements $P_k \rightarrow x_{jkP}$;

End While

End

ALGORITHM 2: Free search operator.

FIGURE 29: ANOVA tests of the global minimum for F_{14} ($D = 2$). FIGURE 27: ANOVA tests of the global minimum for F_{12} ($D = 2$). FIGURE 28: ANOVA tests of the global minimum for F_{13} ($D = 2$). FIGURE 30: Convergence curves for F_1 ($D = 1000$).

(F_9, F_{11}, F_{12}), DE can find the theoretical optimum solution for one function (F_{14}), and KH II can find the theoretical optimum solution for one function (F_2). The number of finding the optimal solution for FSKH is more than that of the other six algorithms. For F_4 and F_6 FSKH has a higher precision of optimization. The accuracy of FSKH can be higher than that of other algorithms for 2 and 14 orders of magnitude, respectively, at least. For F_3, F_{13} , and F_{14} , we can find that there is at least one algorithm that can perform better than FSKH. But in general, even for these three functions, the performance of FSKH is highly competitive with other algorithms. For all functions, the standard deviations of FSKH are very small which indicates that FSKH is very robust and efficient.

FIGURE 31: Convergence curves for F_2 ($D = 200$). FIGURE 32: Convergence curves for F_3 ($D = 100$).

For benchmark functions in Table 1, Figures 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, and 15 are the convergence curves, Figures 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, and 29 are the ANOVA tests of the global minimum. Figures 2, 3, 4, 5, 6, 7, 8, 9, and 11 show that the convergence speed of FSKH is quicker than other algorithms. FSKH is the best algorithm for most functions. Moreover, as seen from the ANOVA tests of the global minimum, we can find that FSKH is the most robust method. For the fourteen functions, other algorithms compared (i.e., PSO, DE, KH, HS, FS, and BA)

FIGURE 33: Convergence curves for F_4 ($D = 1000$). FIGURE 34: Convergence curves for F_5 ($D = 500$).

can only be robust for a few functions, but cannot be robust for all functions. Therefore, FSKH is an effective and feasible method for optimization problems in low-dimensional case.

In order to validate the optimization ability of algorithms in high-dimensional case, this paper set F_3 and F_7 to 100-dimension, F_2 and F_9 to 200-dimension, F_6 and F_{10} to 300-dimension, F_5 and F_8 to 500-dimension, F_1 and F_4 to 1000-dimension. The maximum number of iterations of each algorithm is adjusted to $iterMax = 1000$, and other parameters are the same.

The comparison results for high-dimensional case are shown in Table 4. As seen from the results, the optimization

```

For  $i = 1$  to NP do
  For  $j = 1$  to NK do
     $X_{i,j} = X_{\min,j} + (X_{\max,j} - X_{\min,j}) * \text{rand};$ 
  End For
End For
  /* opposition-based population initialization */
  For  $i = 1$  to NP do
    For  $j = 1$  to NK do
       $X'_{i,j} = X_{\min,j} + (X_{\max,j} - X_{i,j});$ 
    End For
  End For
  Select the NP fittest individuals from  $\{P_0 \cup OP_0\}$  as initial population.

```

ALGORITHM 3: Opposition-based population initialization.

Begin

Initialization. Set the generation counter $iter = 1$; set the foraging speed V_f , maximum diffusion speed $D^{\max}$, and the maximum induced speed $N^{\max}$;

Generate uniformly distributed random population P_0 ;

Opposition-based population initialization

For $i = 1$ to NP **do**

For $j = 1$ to NK **do**

$X'_{i,j} = X_{\min,j} + (X_{\max,j} - X_{i,j});$

End For

End For

Select the NP fittest individuals from $\{P_0 \cup OP_0\}$ as initial population.

Fitness evaluation. Evaluate each krill individual according to its position.

While the termination criterion is not satisfied or $iter < iterMax$ Generation **do**

 Sort the population/krill from best to worst.

For $i = 1 : NP$ (all krill) **do**

 Perform the following motion calculation.

 Motion induced by other individuals

 Foraging motion

 Free search operation

While ($t \leq T$)

 Take exploration walks

 Modification strategy

End While

 Update the krill individual position in the search space.

 Evaluate each krill individual according to its new position.

End For

 Sort the population/krill from best to worst and find the current best.

$iter = iter + 1$.

End While

 Post-processing the results and visualization.

End

ALGORITHM 4: Free search krill herd algorithm.

performance of FSKH is the best. Although the dimension of the functions is very high, the proposed algorithm (FSKH) can also find optimum solutions for six benchmark functions ($F_1, F_2, F_5, F_8 \sim F_{10}$). Some algorithms can show a good performance in low-dimensional case, but in the high-dimensional case it cannot get a good result for most benchmark functions. The optimization ability of FSKH does not show a significant decline with increasing the dimension.

Figures 30, 31, 32, 33, 34, 35, 36, 37, 38, and 39 are the convergence curves for high-dimensional case and Figures

40, 41, 42, 43, 44, 45, 46, 47, 48, and 49 are the ANOVA tests of the global minimum for high-dimensional case. As seen from the ANOVA tests of the global minimum, we can find that FSKH is still the most robust method, and superiority is even more obvious. Therefore, in high-dimensional case, FSKH can also be an effective and feasible method for optimization problems. For metaheuristic algorithm, it is important to tune its related parameters. The proposed algorithm is not a very complex metaheuristic algorithm. Only time interval (C_t) and search radius (R_{ji}) should be fine-tuned in FSKH.

FIGURE 35: Convergence curves for F_6 ($D = 300$). FIGURE 37: Convergence curves for F_8 ($D = 500$). FIGURE 36: Convergence curves for F_7 ($D = 100$). FIGURE 38: Convergence curves for F_9 ($D = 200$).

It is a remarkable advantage of the proposed algorithm compared with other algorithms. But in order to get high accuracy solutions, the time complexity of the proposed algorithm is a little high. How to reduce the time complexity of the proposed algorithm by some strategies is our main work in the future.

5. Conclusions

In order to overcome the shortcomings of krill herd algorithm (e.g., poor population diversity, low precision of

optimization, and poor optimization performance in high dimensional case). This paper introduces the free search strategy into krill herd algorithm and proposes a novel free search krill herd algorithm (FSKH). The main improvement is that the krill individual can search freely and the diversity of krill population is enriched. The proposed algorithm (FSKH) achieves a better balance between the global search and local search. Experiment simulation and comparison results with other algorithms show that the optimization precision, convergence speed, and robustness of FSKH are all better

FIGURE 39: Convergence curves for F_{10} ($D = 300$). FIGURE 40: ANOVA tests of the global minimum for F_1 ($D = 1000$).

than other algorithms for most benchmark functions. So FSKH is a more feasible and effective way for optimization problems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Science Foundation of China under Grant no. 61165015, the Key Project of Guangxi

FIGURE 41: ANOVA tests of the global minimum for F_2 ($D = 200$). FIGURE 42: ANOVA tests of the global minimum for F_3 ($D = 100$). FIGURE 43: ANOVA tests of the global minimum for F_4 ($D = 1000$).

FIGURE 44: ANOVA tests of the global minimum for F_5 ($D = 500$). FIGURE 47: ANOVA tests of the global minimum for F_8 ($D = 500$). FIGURE 45: ANOVA tests of the global minimum for F_6 ($D = 300$). FIGURE 48: ANOVA tests of the global minimum for F_9 ($D = 200$). FIGURE 46: ANOVA tests of the global minimum for F_7 ($D = 100$). FIGURE 49: ANOVA tests of the global minimum for F_{10} ($D = 300$).

Science Foundation under Grant no. 2012GXNSFDA053028, the Key Project of Guangxi High School Science Foundation under Grant no. 20121ZD008, and the Funded by Open Research Fund Program of Key Lab of Intelligent Perception and Image Understanding of Ministry of Education of China under Grant no. IPIU01201100.

References

- [1] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 4, pp. 656–667, 1994.
- [2] J. Kennedy, R. Eberhart, and Y. Shi, *Swarm Intelligence*, Morgan Kaufman, San Francisco, Calif, USA, 2001.
- [3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.
- [4] M. Dorigo, V. Maniezzo, and A. Coloni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.
- [5] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.
- [6] 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.
- [7] 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.
- [8] X. S. Yang, "Firefly algorithms for multimodal optimization," in *System for Automated Geoscientific Analyses*, O. Watanabe and T. Zeugmann, Eds., vol. 5792, pp. 169–178, LNCS, 2009.
- [9] X.-L. Li, Z.-J. Shao, and J.-X. Qian, "Optimizing method based on autonomous animats: fish-swarm Algorithm," *Xitong Gongcheng Lilun yu Shijian/System Engineering Theory and Practice*, vol. 22, no. 11, p. 32, 2002.
- [10] 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.
- [11] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NABIC '09)*, pp. 210–214, December 2009.
- [12] R. Q. Zhao and W. S. Tang, "Monkey algorithm for global numerical optimization," *Journal of Uncertain Systems*, vol. 2, no. 3, pp. 164–175, 2008.
- [13] X. S. Yang, "A new meta-heuristic bat-inspired algorithm," in *Proceedings of the International Workshop on Nature Inspired Cooperative Strategies for Optimization (NICSO '10)*, pp. 65–74.
- [14] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: charged system search," *Acta Mechanica*, vol. 213, pp. 267–289, 2010.
- [15] X.-S. Yang, "Flower pollination algorithm for global optimization," *Lecture Notes in Computer Science*, vol. 7445, pp. 240–249, 2012.
- [16] C. Xu, H. Duan, and F. Liu, "Chaotic artificial bee colony approach to Uninhabited Combat Air Vehicle (UCAV) path planning," *Aerospace Science and Technology*, vol. 14, no. 8, pp. 535–541, 2010.
- [17] O. Hasançebi, T. Teke, and O. Pekcan, "A bat-inspired algorithm for structural optimization," *Computers & Structures*, vol. 128, pp. 77–90, 2013.
- [18] A. Askarzadeh, "Developing a discrete harmony search algorithm for size optimization of wind-photovoltaic hybrid energy system," *Solar Energy*, vol. 98, pp. 190–195, 2013.
- [19] M. Basu and A. Chowdhury, "Cuckoo search algorithm for economic dispatch," *Energy*, vol. 60, pp. 99–108, 2013.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] G.-G. Wang, L. Guo, A. H. Gandomi, A. H. Alavi, and H. Duan, "Simulated annealing-based krill herd algorithm for global Optimization," *Abstract and Applied Analysis*, vol. 2013, Article ID 213853, 11 pages, 2013.
- [24] C. Sur, "Discrete krill herd algorithm—a bio-inspired metaheuristics for graph based network route optimization," in *Distributed Computing and Internet Technology*, R. Natarajan, Ed., vol. 8337 of *Lecture Notes in Computer Science*, pp. 152–163, 2014.
- [25] A. H. Gandomi, A. H. Alavi, and S. Talatahari, "Structural Optimization Using Krill Herd Algorithm," in *Swarm Intelligence and Bio-Inspired Computation Theory and Applications*, X.-S. Yang, Z. Cui, R. Xiao et al., Eds., pp. 335–349, Elsevier, 2013.
- [26] K. Penev and G. Littlefair, "Free search—a comparative analysis," *Information Sciences*, vol. 172, no. 1–2, pp. 173–193, 2005.
- [27] R. S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, 2008.
- [28] K. Tang, X. Yao, P. N. Suganthan et al., *Benchmark Functions For the CEC'2008 Special Session and Competition on Large Scale Global Optimization*, University of Science and Technology of China, Hefei, China, 2007.
- [29] A. H. Gandomi, X.-S. Yang, S. Talatahari, and S. Deb, "Coupled eagle strategy and differential evolution for unconstrained and constrained global optimization," *Computers and Mathematics with Applications*, vol. 63, no. 1, pp. 191–200, 2012.
- [30] A. H. Gandomi, "Benchmark problems in structural optimization," in *Computational Optimization, Methods and Algorithms*, S. Koziel and X. S. Yang, Eds., Study in Computational Intelligence, SCI 356, pp. 259–281, Springer, 2011.