A Novel Metaheuristic Algorithm inspired by Rhino Herd Behavior

Gai-Ge Wang[1] Xiao-Zhi Gao[2] Kai Zenger[2] Leandro dos S. Coelho[3]

[1]School of Computer Science and Technology, Jiangsu Normal University, China, gaigewang@163.com
[2]Department of Electrical Engineering and Automation, Aalto University, Finland,
xiao.z.gao@gmail.com, kai.zenger@aalto.fi
[3]Industrial and Systems Engineering Graduate Program, University of Parana, Brazil,
leandro.coelho@pucpr.br

Abstract

In this paper, inspired by the herding behavior of rhinos, a new kind of swarm-based metaheuristic search method, namely Rhino Herd (RH), is proposed for solving global continuous optimization problems. In various studies of rhinos in nature, the synoptic model is used to describe rhino's space use and estimate its probability of occurrence within a given domain. The number of rhinos increases year by year, and this increment can be forecasted by several population size updating models. Synoptic model and a population size updating model are formalized and generalized to a general-purpose metaheuristic optimization algorithm. In RH, null model without introducing any influences is generated as the initial herding. This is followed by rhino modification via synoptic model. After that, the population size is updated by a certain population size updating model, and newly-generated rhinos are randomly initialized within the given conditions. RH is benchmarked by fifteen test problems in comparison with biogeography-based optimization (BBO) and stud genetic algorithm (SGA). The results clearly show the superiority of RH in searching for the better function values on most benchmark problems over BBO and SGA.

Keywords: rhino herd, synoptic model, population size updating model, benchmark functions, swarm intelligence

1 Introduction

The current real-world optimization problems are increasingly more and more complex and they are hard to be solved by the traditional mathematical methods. On the other hand, human beings are always learning the rule of nature, and improve the ability to handle the complicated problems. By learning the collective behavior of systems, swarm intelligence (SI) (Cui and Gao, 2012) is studied.

Since they are put forward, SI-based algorithms are becoming more and more popular in several engineering applications because of their promising performances when addressing different kinds of real-world optimization problems, such as test-sheet composition (Duan et al., 2012), target threat assessment (Wang et al., 2012a), parameter estimation (Li and Yin, 2014), feature selection (Li and Yin, 2013a), path planning (Wang et al., 2016a; Wang et al., 2012b), wind generator design (Gao et al., 2012a,b), nonlinear system modeling (Gandomi and Alavi, 2011), scheduling (Li and Yin, 2013b), neural network training (Mirjalili et al., 2014a) and knapsack problem (Zou et al., 2011; Feng et al., 2017; Feng et al., 2014). Although SI algorithms involve a great number of methods, particle swarm optimization (PSO) (Kennedy and Eberhart, 1995; Mirjalili et al., *2014b;* Wang et al., *2016b;* Wang et al., *2014c;* Zhao et al., *2012,* Zhao, *2010;* Mirjalili et al., *2013)* and ant colony optimization (ACO) (Dorigo et al., *1996)* are two of the most representative and widely used ones so far. They are inspired by the social behavior of bird when searching for food and remembering paths via pheromone. Recently, inspired by swarm behavior of different animals, serials of SI algorithms have been developed and proposed, such as artificial bee colony (ABC) (Karaboga and Basturk, 2007), elephant herding optimization (EHO) (Wang et al., 2015a;Wang et al., *2016b)* chicken swarm optimization (CSO) (Meng et al., 2014) bird swarm algorithm (BSO) (Meng et al., 2015) cuckoo search (CS) (Yang and Deb, 2009; Li et al., 2013; Wang et al., 2016c; Wang et al., *2016d;*Wang et al., *2016e ;*Li and Yin, *2015)* bat algorithm (BA) (Yang, 2010; Mirjalili et al., 2013; Zhang and Wang, 2012; Wang et al., 2015b), firefly algorithm (FA) (Gandomi et al.,2011; Yang, 2010; Wang et al., *2014d;* Guo et al., *2013)* ant lion optimizer (ALO) (Mirjalili, 2015), chaotic swarming of particles (CSP) (Kaveh et al., 2014) monarch butterfly optimization (MBO) (Wang et al., 2015c;Wang et al., *2016e;*Wang et al., *2016f;*Ghetas et al., *2016)* krill herd (KH) (Gandomi and Alavi, 2012; Wang et al., 2013; Gandomi et al., *2013;* Wang et al., *2014d;*Wang et al., *2014e;* Wang et al., *2014f;*Wang et al., *2016f;*Guo et al., *2014;* Wang et al., *2016g;*Li et al., *2015)* multi-verse optimizer (MVO) (Mirjalili et al., 2016) dragonfly algorithm (DA) (Mirjalili, 2016), and grey wolf optimizer (GWO) (Mirjalili et al.,.2014; Saremi et al., *2014)* These algorithms have been successfully used to address an array of real-world problems.

Except for SI algorithms, inspired by the evolutionary rule of nature, evolutionary algorithms (EAs) are proposed. Among different kinds of EAs, the following algorithms are some of the most representative paradigms, which are genetic algorithm (GA) (Goldberg, 1998), stud genetic algorithm (SGA) (Khatib and Fleming, 1998), differential evolution (DE) (Storn and Price, 1997; Zou et al.,2013; Li and Yin, 2016) earthworm optimization algorithm (EWA) (Wang et al.,2015e) biogeography-based optimization (BBO) (Simon, 2008;Li and Yin, 2012; Saremi et al.,2014; Li and Yin, 2012) and animal migration optimization (AMO) (Li et al.,.2014).

Rhinos are one of the largest mammals in the world. The studies about rhinos have been done in various aspects, involving rhino's space use and the increment of population size. For rhino's space use, synoptic model (Horne et al., *2008)* is one of the most representative paradigms that is used to estimate its probability of occurrence in a given domain associated with a fixed spatial area (i.e. home range), the spatial distribution of resources, and the occurrence of other animals (Horne et al., 2008) which are called herding density variables (HDVs). With the increment of rhino number, the resources represented by HDVs and owned by each rhino individual are becoming less and less. That is, the fewer recourses, the worse they feel. In our current work, we use rhino comfort index (RCI) to represent this feeling. In other words, RCI is used to measure the goodness of a feasible solution. A good solution is analogous to a rhino with a high RCI, and a poor solution is similar to a rhino with a low RCI.

In this paper, synoptic models and population size updating models are formalized and generalized to a general-purpose metaheuristic algorithm. Accordingly, a new kind of swarm-based algorithm, called Rhino Herd (RH), is proposed for coping with global optimization tasks. Null model in synoptic model is a special kind of model without any influences from others. In RH, null model is considered as the initial herding or the herding before updating. This is followed by rhino modification via synoptic model. Finally, the population size is updated by some population size updating model, and newly-generated rhinos are randomly initialized within the given conditions. The RH is benchmarked by fifteen test optimization problems by comparing it with BBO, and SGA. The results clearly show the superiority of RH in searching for the better function values on most benchmarks over BBO and SGA.

The rest of paper is structured as follows. Section 2 reviews the herding behavior of rhinos in nature, involving synoptic model and population size updating model. Subsequently, Section 3 discusses how the herding behavior of rhinos can be used to formulate a general-purpose metaheuristic algorithm. To fully investigate the performance of RH algorithm, several simulation results comparing the optimal RH algorithm with other optimization methods on fifteen benchmark functions, are presented in Section 4. Finally, Section 5 draws some concluding remarks.

2 Herding behavior of rhinos

Rhinos are one of the biggest mammals in the world, and their weight can reach one ton or more. Rhinos are herbivorous, and they mainly live on in leafy materials. The current rhinos only involve five extant species, and some of the rhinos have two horns, while others have a single horn. Several researchers have done many studies of rhinos from various aspects. Synoptic model of space use and population size updating model are two of the most representative paradigms.

2.1 A synoptic model of space use

To describe rhino space use, a multivariate model, called synoptic model, is proposed to estimate a rhino's probability of occurrence associated with various HDVs, such as a fixed spatial area (i.e., home range), the spatial distribution of resources, and the occurrence of other animals (Horne et al., 2008). In synoptic model, $s(x)$ represents the probability density of finding the rhino at location x during the period of study. For each location, k environmental variables (HDVs) are used as covariates to model a rhino's utilization distribution (Horne et al.,2008). In this model, a null model of space use $f_0(x)$ is applied to describe a rhino's utilization distribution without effects from environmental covariates, which is expressed in the form of an exponential power model (Horne et al.,2008), shown in (1).

$$f_0(x) = \frac{2}{c^2 \pi a^2 \Gamma(c)} \exp\left[-\left(\frac{\| x-\mu \|}{a} \right)^{2/c} \right] \qquad (1)$$

where Γ is the gamma function, μ is the center of the distribution, a>0 is the scale parameter, c>0 is the shape parameter, and $\|x-\mu\|$ is the distance between x and μ (Horne et al., 2008).

Subsequently, a spatially explicit environmental covariate $H(x)$ is added to the null model $f_0(x)$, where $H(x)$ is defined as a function for describing the environmental covariate. The function $H(x)$ has various forms according to its environmental variables to be described. After introducing one covariate, the synoptic model can be expressed as

$$s(x) = \frac{f_0(x)+\beta H(x) f_0(x)}{\int_x \left[f_0(x)+\beta H(x) f_0(x) \right]} \qquad (2)$$

where β is an estimated selection parameter controlling the magnitude of the effect.

Similarly, after introducing k covariates, the synoptic model of space use can be expressed as

$$s(x)=\frac{f_0(x)\prod_{i=1}^{k}\left(1+\beta_i H_i(x)\right)}{\int_x\left[f_0(x)\prod_{i=1}^{k}\left(1+\beta_i H_i(x)\right)\right]} \tag{3}$$

The denominator of (3) is hard to handle, as it cannot be analytically intractable for most combinations of initial models and environmental covariates (Horne et al., 2008). As an alternative, the landscape can be divided into l discrete grid cells, and the denominator of (3) is therefore calculated as follows:

$$A\sum_{j=1}^{l}\left[f_0(x_j)\prod_{i=1}^{k}\left(1+\beta_i H_i(x_j)\right)\right]\approx\int_x\left[f_0(x)\prod_{i=1}^{k}\left(1+\beta_i H_i(x)\right)\right] \tag{4}$$

where A is the area of each grid cell.

Accordingly, (3) can be approximated as

$$s(x)=\frac{f_0(x)\prod_{i=1}^{k}\left(1+\beta_i H_i(x)\right)}{A\sum_{j=1}^{l}\left[f_0(x_j)\prod_{i=1}^{k}\left(1+\beta_i H_i(x_j)\right)\right]} \tag{5}$$

More information about synoptic model of space use can be found in (Horne et al., 2008).

2.2 Population size updating model

The original data from two sites in South Africa is collected to model the rhino population and predict the rhino number next year (Cromsigt et al., 2002). Population density is thus determined. The predicted rhino number, $n(t)$, and the real population number from the original data, $p(t)$, have the following relationship:

$$p(t)=n(t)+\varepsilon(t) \tag{6}$$

where $\varepsilon(t)$ is an error term between $p(t)$ and $n(t)$.

3 Rhino herd (RH) algorithm

Here, the herding behavior of rhinos described in Section 2, involving synoptic model of space use and population size updating model, is formed to handle optimization problems.

3.1 Synoptic model

In this section, how to use the synoptic model of space use to optimize is given. As aforementioned, the synoptic model is to estimate a rhino's probability of occurrence within a given domain. Here, an updated synoptic model is used to determine the direction of the search for the next iteration. For the jth HDV of rhino i, this updated model can be given as

$$S(X_{i,j})=\frac{f_0(X_i)\left(1+\alpha_i H_i(X_{i,j})\right)}{A\sum_{k=1}^{n}\left[f_0(X_{i,k})\left(1+\beta_k H_k(X_{i,k})\right)\right]} \tag{7}$$

where α_i and β_k are the estimated selection parameter controlling the magnitude of the effect from $H(X)$; n is the population size. Null model $f_0(X)$ and A are defined as above. $H_i(X)$ and $H_k(X)$ are defined as functions for describing the related variables that have influence on rhino i. In our current work, for the sake of simplicity, we set $H_i(X_{i,j})=X_{i,j}$, $H_k(X_{i,k})=X_{i,k}$. That is, $H(X)=X$.

After all the HDVs in rhino i are calculated, the rhino i is updated as

$$X_{i,new}=\begin{cases}X_i+S(X_i)\otimes X_i, & rand>0.5\\ X_i\text{-}S(X_i)\otimes X_i, & rand\le 0.5\end{cases} \tag{8}$$

where $X_{i,new}$ is the updated rhino, symbol "$\otimes$ represents pairwise product, and $rand$ is a random number drawn from a stochastic distribution.

In order to increase the diversity of the population in the later search, a random term is added to above equation. Therefore, the updated expression can be given as

$$X_{i,new}=\begin{cases}X_i+rand\times S(X_i)\otimes X_i, & rand>0.5\\ X_i\text{-}rand\times S(X_i)\otimes X_i, & rand\le 0.5\end{cases} \tag{9}$$

It should be noted that for the center μ in (1), for the jth HDV in μ, it can be calculated as

$$\mu_j=\frac{1}{n}\times\sum_{i=1}^{n}X_{i,j} \tag{10}$$

3.2 Population size updating model

The rhino population number varies each year, and the number is generally becoming larger and larger in nature. This trend can be modeled as above. In this paper, the exponential model is used to update population size, which is given as

$$n_{t+1}=n_t+r\times n_t \tag{11}$$

where r is a constant specific growth rate; n_t and n_{t+1} are population size at generation t and t+1, respectively.

3.3 RH algorithm

Rhino modification operator is a critical operator in RH algorithm, which can loosely be given below.

Algorithm 1 Rhino modification

Begin

Calculate RCI_0 for null model (1).

for i=1 to n (all the rhinos in the herding) **do**

Calculate synoptic model S.

Update the rhino i according to S;

end for i

End

Population size updating is another important operator that updates rhino population size based on a certain rule. The exponential model is provided to calculate the modified population size n' in our current work (Section 3.2).

Algorithm 2 Population size updating

Begin

Calculate n' as per the population size updating model.

for i= n+1 to n' (all the newly-generated rhinos) **do**

Initialize X_i with a randomly generated HDV^m.

end for i

End

According to the analyses above, the schematic framework of RH algorithm can be described as follows.

Algorithm 3 Rhino Herd (RH) Algorithm

Begin

Step 1: Parameters initialization. Firstly, the problem-dependent solutions are mapped to HDVs and rhinos. In addition, an elitism parameter, and the parameters used in null model and population size updating models (see Section 4.1) are initialized.

Step 2: Generate a group of rhinos at random X_0^n Each rhino represents a feasible solution to the problem of interest.

Step 3: Map each rhino to RCI. Each rhino in initial herding is mapped to RCI that can measure the goodness of the rhino.

Step 4: Calculate RCI$_0$. The RCI$_0$ of null model at each generation is calculated. Here, the herding before being modified can be considered as null model at each generation, followed by modifying each rhino based on this null model.

Step 5: Rhino modification. For each rhino in the herding, it is modified by synoptic model (Algorithm 1).

Step 6: Map each rhino to RCI. Each rhino in newly-generated herding is mapped to RCI.

Step 7: Population size updating model. Update population size by using updating model. Rrandomly initialize the newly-generated rhinos and calculate its corresponding RCI for each rhino (see Algorithm 2).

Step 8: Stop or not. Go to **Step 4,** if the termination criterion is not satisfied; terminate the optimization process, if the predefined termination criterion is reached.

End

In Algorithm 3, a rhino comfort index RCI: $X \to R$ is a measure of goodness of the solution that is represented by the rhino. It should be mentioned that, for most population-based metaheuristic algorithms, RCI is called fitness, and its value is the fitness value. A rhino comfort index of null model RCI$_0$: $N \to R$ is a measure of goodness of the solution that is represented by the X_0. Here, RCI$_0$ is a special RCI that is different with other RCI. Null model can be formulated in different forms, and there are various ways of calculating RCI$_0$.

4 Simulation results

In this section, the RH is verified by benchmark evaluation in comparison with two methods (BBO (Simon, 2008), and SGA (Khatib and Fleming, 1998) on fifteen test problems (Table 1).

In order to obtain fair results, all the implementations are conducted under the same conditions shown in (Wang et al., 2014a).

The same parameters for RH are set as follows: the area of each grid cell A=1; the constant specific growth rate r=0.04; the scale parameter a=2831; the shape parameter c=0.53; for rhino i, its estimated selection parameter a_i and β_i are set to be its RCI and $1/\mathrm{RCI}$, respectively. The numbers of generations and initial population size are set to 50 and 50, respectively. In other methods, their parameter settings can be found in (Wang et al., 2014a,b) The dimension is twenty.

Table 1. Benchmark functions.

No.	*Name*	*No.*	*Name*
F01	Ackley	F09	Schwefel 2.26
F02	Alpine	F10	Schwefel 1.2
F03	Griewank	F11	Schwefel 2.22
F04	Holzman 2 function	F12	Schwefel 2.21
F05	Levy 8	F13	Step
F06	Pathological function	F14	Sum function
F07	Perm	F15	Zakharov
F08	Powell		

Metaheuristic algorithms are always based on certain stochastic distribution. Therefore, 50 independent runs are implemented (Table 2). In the following experiments, the best solution is highlighted in **bold**.

From Table 2, RH method has demonstrated its best performance on F01, F05, F08, and F12-F14. At the same time, RH is able to find the best solutions with the smallest Std (standard deviation) on F02, F07, and F11. BBO has shown its best performance on F09-F10. Both of them perform significantly better than SGA. This indicates that RH has a powerful search ability, and can find the fittest solution on most benchmarks.

Table 2. Fitness values obtained by three methods.

Test	BBO	RH	SGA
F01	6.19±0.85	**3.84±0.18**	7.80±1.02
F02	**1.68**±0.85	2.07±**0.60**	3.55±1.70
F03	9.98±4.32	10.59±19.24	**7.91±2.97**
F04	398.30±669.60	7.01E4±5.88E4	**159.80±109.70**
F05	2.64±1.38	**2.40±0.42**	2.52±1.54
F06	5.22±0.55	2.77±0.49	4.98±0.56
F07	6.79E51±2.19E51	6.01E51±**1.33E36**	6.01E51±**1.33E36**
F08	167.00±97.03	**8.82±5.21**	100.10±55.83
F09	**926.40±258.80**	4.52E3±483.70	1.04E3±268.20
F10	**1.16E4±4.00E3**	3.74E4±1.05E4	1.62E4±4.89E3
F11	**3.74**±1.75	4.16±**0.86**	11.94±3.34
F12	48.18±9.23	**1.84±0.26**	43.18±13.20
F13	3.30±1.23	**1.12±0.33**	5.26±1.54
F14	89.80±26.06	**6.88±2.87**	121.20±52.95
F15	**137.70**±43.15	217.50±174.50	224.40±65.28

Moreover, the convergent processes of three most representative algorithms on the most representative benchmarks can be given as follows (Figures 1 and 2).

Figure 1 shows the convergent history of F01-F02, F05, and F08. For F1 case, it can be easily observed that RH11, BBO and SGA rank the first, the second, and the third, respectively. For F02 and F05 cases, although all the three algorithms converge to the similar final solutions, RH has the fastest convergent speed, and it can find the best solution within ten generations. For F08 case, RH has a stable convergence speed, and it can find the final best solution after BBO and SGA have been trapped into the premature status.

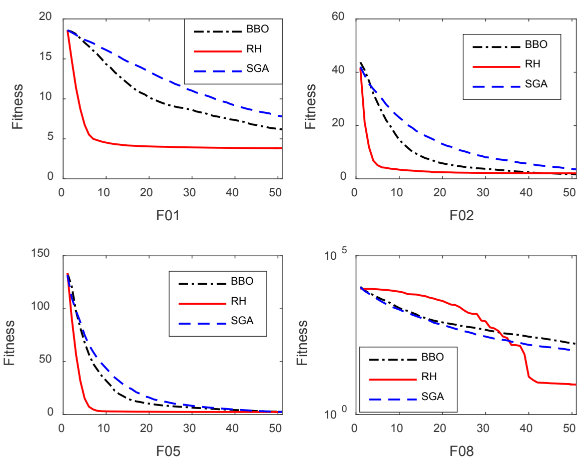

Figure 1. Convergent curves of the benchmarks F01, F02, F05, and F08.

Figure 2 shows the convergent history of benchmarks F11-F14. For F11 and F13 cases, although all the three algorithms can produce similar final solutions, RH has the fastest convergence speed, and it can find the best solution within ten generations. For F12 case, it is clearly visible that RH has a much better solution than BBO and SGA, which have similar optimization performances. For F14 case, RH can eventually find the optimal solution.

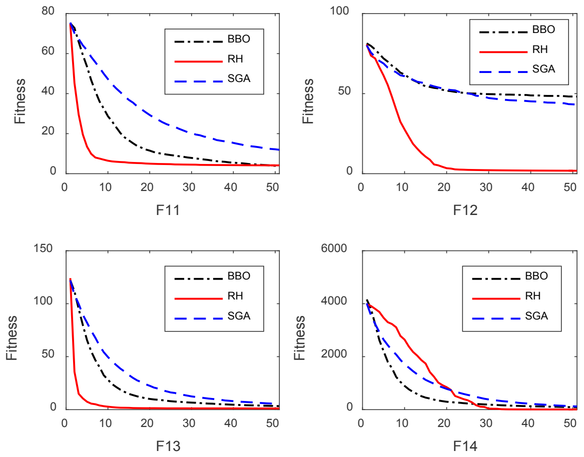

Figure 2. Convergent curves of the benchmarks F11-F14.

5 Discussions and conclusions

We have shown how rhino herding, the research of synoptic model and population size updating model, can be used to develop a novel algorithm for optimization. This new family of algorithms is called RH, which has been benchmarked by fifteen test problems. The results have shown RH's competitive performance in comparison with other two state-of-the-art algorithms. Unfortunately, we cannot conclude RH algorithm is universally better than other two algorithms, or *vice versa*, as per the no free lunch theorem. However, the good performance of RH algorithm in comparison with two other algorithms on fifteen benchmarks demonstrates that it is well capable of addressing practical problems successfully.

In our current work, the influence of some herding density variables is exerted on the null model. Other factors, such as the ratio of male and female, the sun, and landscape will be included in the improved synoptic model as herding density variables.

Another bottleneck of many algorithms is computational requirements. How to reduce computation efforts is highly worthy of in-depth study for an algorithm.

Acknowledgements

This work was supported by Jiangsu Province Science Foundation for Youths (No. BK20150239) and National Natural Science Foundation of China (No. 61503165).

References

J. P. Cromsigt, J. Hearne, I. M. Heitkönig, and H. H. Prins.

Using models in the management of black rhino populations. *Ecological modelling,* 149(1):203-211, 2002.

Z. Cui and X. Gao. Theory and applications of swarm intelligence. *Neural Computing & Applications,* 21(2):205-206, 2012.

M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* 26(1):29-41, 1996.

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,* 2012:1-22, 2012.

Y. Feng, G.-G. Wang, S. Deb, M. Lu, and X. Zhao. Solving 0-1 knapsack problem by a novel binary monarch butterfly optimization. *Neural Computing and Applications*, 28(7):1619-1634, 2017.

Y.-H. Feng, G.-G. Wang, Q. Feng, and X.-J. Zhao, An effective hybrid cuckoo search algorithm with improved shuffled frog leaping algorithm for 0-1 Knapsack problems. *Computational Intelligence and Neuroscience,* 2014(2014):1-17, 2014.

A. H. Gandomi and A. H. Alavi. Multi-stage genetic programming: a new strategy to nonlinear system modeling. *Information Sciences,* 181(23):5227-5239, 2011.

A. H. Gandomi, X.-S. Yang, and A. H. Alavi. Mixed variable structural optimization using firefly algorithm. *Computers & Structures,* 89(23-24):2325-2336, 2011.

A. H. Gandomi and A. H. Alavi. Krill herd: a new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation,* 17(12): 4831-4845, 2012.

A. H. Gandomi, S. Talatahari, F. Tadbiri, and A. H. Alavi. Krill herd algorithm for optimum design of truss structures. *International Journal of Bio-Inspired Computation,* 5(5):281-288, 2013.

X. Z. Gao, X. Wang, S. J. Ovaska, and K. Zenger, A hybrid optimization method of harmony search and opposition-based learning. *Engineering Optimization,* 44(8):895-914, 2012a.

X.-Z. Gao, X. Wang, T. Jokinen, S. J. Ovaska, A. Arkkio, and K. Zenger. A hybrid optimization method for wind generator design. *International Journal of Innovative Computing, Information and Control,* 8(6):4347-4373, 2012b.

M. Ghetas, H.-Y. Chan, and P. Sumari, Harmony-based monarch butterfly optimization algorithm. In *Proceedings of the 2015 IEEE International Conference on Control System, Computing and Engineering* (ICCSCE), pages 156-161, November 2015.

D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine learning*. Addison-Wesley, New York, 1998.

L. Guo, G.-G. Wang, H. Wang, and D. Wang. An effective hybrid firefly algorithm with harmony search for global numerical optimization. *The Scientific World Journal,* 2013(2013):1-10, 2013.

L. Guo, G.-G. Wang, A. H. Gandomi, A. H. Alavi, and H. Duan. A new improved krill herd algorithm for global numerical optimization. *Neurocomputing,* 138: 392-402, 2014.

J. S. Horne, E. O. Garton, and J. L. Rachlow. A synoptic model of animal space use: Simultaneous estimation of home range, habitat selection, and inter/intra-specific relationships. *Ecological Modelling,* 214(2-4):338-348, 2008.

D. Karaboga and B. Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization,* 39(3):459-471, 2007.

A. Kaveh, R. Sheikholeslami, S. Talatahari, and M. Keshvari-Ilkhichi. Chaotic swarming of particles: a new method for size optimization of truss structures. *Advances in Engineering Software,* 67:136-147, 2014.

J. Kennedy and R. Eberhart, Particle swarm optimization. In *Proceeding of the IEEE International Conference on Neural Networks*, Perth, Australia, pages 1942-1948, 1995.

W. Khatib and P. Fleming. The stud GA: A mini revolution?, *Parallel Problem Solving from Nature - PPSN V*. Lecture Notes in Computer Science. A. Eiben, T. Bäck, M. Schoenauer and H.-P. Schwefel, eds., 683-691, London, UK: Springer Berlin Heidelberg, 1998.

X. Li, and M. Yin. Parameter estimation for chaotic systems by hybrid differential evolution algorithm and artificial bee colony algorithm. *Nonlinear Dynamics,* 77(1-2): 61-71, 2014.

X. Li, and M. Yin. Multiobjective binary biogeography based optimization for feature selection using gene expression data. *IEEE Transactions on NanoBioscience,* 12(4): 343-353, 2013a.

X. Li, and M. Yin. An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure. *Advances in Engineering Software,* 55: 10-31, 2013b.

X. Li, J. Wang, and M. Yin. Enhancing the performance of cuckoo search algorithm using orthogonal learning method. *Neural Computing and Applications,* 24(6): 1233-1247, 2013.

X. Li, and M. Yin. Modified cuckoo search algorithm with self adaptive parameter method. *Information Sciences,* 298: 80-97, 2015.

Z.-Y. Li, J.-H. Yi, and G.-G. Wang. A new swarm intelligence approach for clustering based on krill herd with elitism strategy. *Algorithms,* 8(4):951-964, 2015.

X. Li and M. Yin. Modified differential evolution with self-adaptive parameters method. *Journal of Combinatorial Optimization,* 31(2):546-576, 2016.

X. Li, and M. Yin. Multi-operator based biogeography based optimization with mutation for global numerical optimization, *Computers & Mathematics with Applications,* 64(9):2833-2844, 2012a.

X.-T. Li, and M.-H. Yin. Parameter estimation for chaotic systems using the cuckoo search algorithm with an orthogonal learning method. *Chinese Physics B,* 21(5): 050507, 2012b.

X. Li, J. Zhang, and M. Yin. Animal migration optimization: an optimization algorithm motivated by elephant herding behavior. *Neural Computing and Applications*, 24(7-8):1867-1877, 2014.

X. Meng, Y. Liu, X. Gao, and H. Zhang. A new bio-inspired algorithm: chicken swarm optimization. *Advances in Swarm Intelligence*, Lecture Notes in Computer Science Y. Tan, Y. Shi and C. C. Coello, eds. 86-94: Springer International Publishing, 2014.

X.-B. Meng, X. Z. Gao, L. Lu, Y. Liu, and H. Zhang. A new bio-inspired optimisation algorithm: bird swarm algorithm. *Journal of Experimental & Theoretical Artificial Intelligence*, 1-15, 2015.

S. Mirjalili, S. M. Mirjalili, and A. Lewis. Let a biogeography-based optimizer train your Multi-Layer Perceptron. *Information Sciences,* 269: 188-209, 2014a.

S. Mirjalili, G.-G. Wang, and L. d. S. Coelho. Binary optimization using hybrid particle swarm optimization and gravitational search algorithm. *Neural Computing and Applications,* 25(6):1423-1435, 2014b.

S. Mirjalili and A. Lewis. S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization. *Swarm and Evolutionary Computation,* 9:1-14, 2013.

S. Mirjalili. The ant lion optimizer. *Advances in Engineering Software,* 83:80-98, 2015.

S. Mirjalili, S. M. Mirjalili, and X.-S. Yang. Binary bat algorithm. *Neural Computing and Applications,* 25(3-4): 663-681, 2013.

S. Mirjalili, S. M. Mirjalili, and A. Hatamlou. Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Computing and Applications,* 27(2): 495-513, 2016.

S. Mirjalili. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications,* 27(4):1053-1073, 2016.

S. Mirjalili, S. M. Mirjalili, and A. Lewis. Grey wolf optimizer. *Advances in Engineering Software,* 69:46-61, 2014c.

S. Saremi, S. Mirjalili, and A. Lewis. Biogeography-based optimisation with chaos. *Neural Computing and Applications,* 25(5):1077-1097, 2014.

S. Saremi, S. Z. Mirjalili, and S. M. Mirjalili. Evolutionary population dynamics and grey wolf optimizer. *Neural Computing and Applications,* 26(5):1257-1263, 2014.

D. Simon. Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation,* 12(6):702-713, 2008.

R. Storn and K. Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization,* 11(4):341-359, 1997.

G.-G. Wang, L. Guo, H. Duan, L. Liu, and H. Wang. The model and algorithm for the target threat assessment based on Elman_AdaBoost strong predictor. *Acta Electronica Sinica,* 40(5):901-906, 2012a.

G.-G. Wang, H. E. Chu, and S. Mirjalili. Three-dimensional path planning for UCAV using an improved bat algorithm. *Aerospace Science and Technology,* 49:231-238, 2016a.

G. Wang, L. Guo, H. Duan, L. Liu, H. Wang, and J. Wang. A hybrid meta-heuristic DE/CS algorithm for UCAV path planning. *Journal of Information and Computational Science,* 9(16):4811-4818, 2012b.

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,* 24(3-4):853-871, 2014a.

G.-G. Wang, L. Guo, A. H. Gandomi, G.-S. Hao, and H. Wang. Chaotic krill herd algorithm. *Information Sciences,* 274:17-34, 2014b.

G.-G. Wang, A. H. Gandomi, A. H. Alavi, and S. Deb. A hybrid method based on krill herd and quantum-behaved particle swarm optimization. *Neural Computing and Applications,* 27(4):989-1006, 2016a.

G.-G. Wang, A. H. Gandomi, X.-S. Yang, and A. H. Alavi. A novel improved accelerated particle swarm optimization algorithm for global numerical optimization. *Engineering Computations,* 31(7):1198-1220, 2014c.

G.-G. Wang, S. Deb, X.-Z. Gao, and L. d. S. Coelho. A new metaheuristic optimization algorithm motivated by elephant herding behavior. *International Journal of Bio-Inspired Computation*, 8(6):394, 2016b.

G.-G. Wang, A. H. Gandomi, X. Zhao, and H. C. E. Chu. Hybridizing harmony search algorithm with cuckoo search for global numerical optimization. *Soft Computing*, 20(1):273-285, 2016c.

G.-G. Wang, S. Deb, A. H. Gandomi, Z. Zhang, and A. H. Alavi. Chaotic cuckoo search, *Soft Computing*, 20(9):3349-3362, 2016d.

G.-G. Wang, A. H. Gandomi, X.-S. Yang, and A. H. Alavi. A new hybrid method based on krill herd and cuckoo search for global optimization tasks. *International Journal of Bio-Inspired Computation*, 8(5):2016e.

G.-G. Wang, S. Deb, and L. d. S. Coelho. Elephant herding optimization. In *2015 3rd International Symposium on Computational and Business Intelligence (ISCBI 2015),* Bali, Indonesia: 1-5, 2015a.

G.-G. Wang, B. Chang, and Z. Zhang. A multi-swarm bat algorithm for global optimization. In *2015 IEEE Congress on Evolutionary Computation (CEC 2015)*, Sendai, Japan: 480-485, 2015b.

G.-G. Wang, L. Guo, H. Duan, and H. Wang. A new improved firefly algorithm for global numerical optimization. *Journal of Computational and Theoretical Nanoscience,* 11(2):477-485, 2014d.

G.-G. Wang, S. Deb, and Z. Cui. Monarch butterfly optimization. *Neural Computing and Applications*, 2015c.

G.-G. Wang, G.-S. Hao, S. Cheng, and Q. Qin. A discrete monarch butterfly optimization for Chinese TSP problem. *Advances in Swarm Intelligence*, 2016e.

G.-G. Wang, S. Deb, X. Zhao, and Z. Cui. A new monarch butterfly optimization with an improved crossover operator. *Operational Research: An International Journal*, 2016f.

G.-G. Wang, A. H. Gandomi, and A. H. Alavi. A chaotic particle-swarm krill herd algorithm for global numerical optimization. *Kybernetes,* 42(6):962-978, 2013.

G.-G. Wang, A. H. Gandomi, A. H. Alavi, and G.-S. Hao. Hybrid krill herd algorithm with differential evolution for global numerical optimization. *Neural Computing and Applications,*25(2):297-308, 2014e.

G.-G. Wang, A. H. Gandomi, and A. H. Alavi. An effective krill herd algorithm with migration operator in biogeography-based optimization. *Applied Mathematical Modelling,* 38(9-10):2454-2462, 2014f.

G-G Wang, A. H. Gandomi, and A. H. Alavi. Stud krill herd algorithm. *Neurocomputing,* 128: 363-370, 2014g.

G.-G. Wang, A. H. Gandomi, A. H. Alavi, and S. Deb. A multi-stage krill herd algorithm for global numerical optimization. *International Journal on Artificial Intelligence Tools,* 25(2):1550030, 2016g.

G.-G. Wang, S. Deb, A. H. Gandomi, and A. H. Alavi. Opposition-based krill herd algorithm with Cauchy mutation and position clamping. *Neurocomputing,* 177: 147-157, 2016h.

G.-G. Wang, S. Deb, and L. d. S. Coelho. Earthworm optimization algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. *International Journal of Bio-Inspired Computation*, 2015d.

X. S. Yang. *Nature-inspired metaheuristic algorithms*, 2nd ed., Luniver Press, Frome, 2010.

X. S. Yang, Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation,*2(2):78-84, 2010.

X. S. Yang and S. Deb, Cuckoo search via Lévy flights. In *Proceeding of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009),* Coimbatore, India:210-214, 2009.

J.-W. Zhang and G.-G. Wang. Image matching using a bat algorithm with mutation. *Applied Mechanics and Materials,* 203(1):88-93, 2012.

X. Zhao, B. Song, P. Huang, Z. Wen, J. Weng, and Y. Fan. An improved discrete immune optimization algorithm based on PSO for QoS-driven web service composition, *Applied Soft Computing,* 12(8):2208-2216, 2012.

X. Zhao. A perturbed particle swarm algorithm for numerical optimization. *Applied Soft Computing,* 10(1):119-124, 2010.

D. Zou, L. Gao, S. Li, and J. Wu. Solving 0-1 knapsack problem by a novel global harmony search algorithm. *Applied Soft Computing,* 11(2):1556-1564, 2011.