JOIV

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

Common Benchmark Functions for Metaheuristic Evaluation: A Review

Kashif Hussain[#], Mohd Najib Mohd Salleh[#], Shi Cheng[*], Rashid Naseem[**]

[#]Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Batu Pahat, Malaysia
[*] School of Computer Science, Shaanxi Normal University, Xi'an, China
[**]Department of Computer Science, City University of Science and Information, Technology, Peshawar, Pakistan
E-mail: najib@uthm.edu.my

Abstract— **In literature, benchmark test functions have been used for evaluating performance of metaheuristic algorithms. Algorithms that perform well on a set of numerical optimization problems are considered as effective methods for solving real-world problems. Different researchers choose different set of functions with varying configurations, as there exists no standard or universally agreed test-bed. This makes hard for researchers to select functions that can truly gauge the robustness of a metaheuristic algorithm which is being proposed. This review paper is an attempt to provide researchers with commonly used experimental settings, including selection of test functions with different modalities, dimensions, the number of experimental runs, and evaluation criteria. Hence, the proposed list of functions, based on existing literature, can be handily employed as an effective test-bed for evaluating either a new or modified variant of any existing metaheuristic algorithm. For embedding more complexity in the problems, these functions can be shifted or rotated for enhanced robustness.**

Keywords— **benchmark test functions, numerical optimization, metaheuristic algorithms, optimization.**

I. INTRODUCTION

Strong relationship between metaheuristic algorithms and numerical test problems or test functions is found in existing literature. Generally, the introduction of any new metaheuristic algorithm is often accompanied with a set of benchmark test functions that are used in order to prove efficiency of the algorithm [1]. In fact, there exists no standard list of test functions. Different researchers choose different set of benchmark test functions, which is not backed by any justifiable reason for the choice made in this regard. Moreover, the number of functions in a test-bed also varies significantly from two functions to two dozen functions. The range of test functions includes different features; such as, unimodal and multimodal, regular and irregular, separable and non-separable, etc. These functions are also often rotated or shifted and also used with wider range of dimensions (from 10D to 1000D) to add complexity in optimization landscape. This creates significant gap for new researchers who want to validate any modified version of existing metaheuristic algorithm or completely new algorithm on benchmark test functions. Often, the researchers have not enough background knowledge on landscape structure, difficulties, modalities, separability, non-linearity, and various other aspects of optimization problems.

Yang, in a survey paper [2] about benchmark functions, suggested to use a good set of benchmark functions for performance validation and comparison with other algorithms. However, there is no any agreed good list of functions among metaheuristic researchers. But, Yang elaborates the good set should contain functions with diverse characteristics and should be unbiased. Nevertheless, researchers choose a set of functions on which the target algorithm performs better than other selected counterparts. This raises an important concern over biasness towards algorithm's suitable set of functions. Hence, in future, an important research can be conducted to find relationship between metaheuristic algorithms and benchmark test functions in substantial existing literature.

Since, looking for a set of benchmark test functions for testing an algorithm is a tedious job, this limited survey is an attempt to suggest commonly used benchmark test functions, representing unconstrained and single-objective optimization problems, from the related literature. The subsequent section explains test functions and different characteristics. Section III reports existing work performed on metaheuristic algorithm validation using benchmark functions. Section IV provides detail of commonly used benchmark test functions. Section V concludes and recommends future potential research in the related area.

II. Benchmark Functions And Characteristics

Benchmark test functions are fundamentally optimization problems presented in the form of mathematical numerical functions. These functions are optimized with a set of best suitable parameter values that help achieve best solution, where D is the problem dimensions. The best solution is hidden in large amount of sub-optimal solutions spread all over problem landscape with different number and types of hills and valleys. Any optimization algorithm, including metaheuristic algorithms, tend to find the best solution (but not always guaranteed to have obtained) as quickly as possible. The efficiency of any metaheuristic is gauged with its global searchability and local convergence ability. The algorithms that have better global searchability are hard to trap in sub-optimal (local minima or maxima) locations. At the same time, metaheuristics with efficient convergence ability are hard to miss any best solution within the neighbourhoods.

There can be found many online resources that provide greater detail along with MATLAB and R language codes for the functions, see for example [3], [4], and Congress of Evolutionary Computing (CEC) [5], etc. According to characteristics (modality, basins, valleys, and separability, dimensionality) [2] which shape the landscape of the problem, the test functions can be grouped into following categories:

A. Modality

Modality defines the number of peaks in problem landscape. These peaks form local minima and global minima locations.

Unimodal Functions: These functions have one valley and one global minima location where the best solution resides. Such functions are considered to be easy to solve, however shifting and rotating these functions increases difficulty. Metaheuristic algorithms can be tested on these functions for evaluating local searchability.

Multimodal Functions: The functions in this category maintain more than one solutions but true global best is one. Such functions have many local minima locations but one true global minima. Hence any metaheuristic algorithm needs to travel all the landscape in order to find the true global best solution. These functions are difficult to solve and are good for testing global searchability of any algorithm.

B. Basin

A basin is a steep descent surrounded by high peaks. The multimodal functions contain more number of basins of local minima than global minima. Poor algorithms tend to be hampered in basins of local minima and fail to find the basin of global minima.

C. Valleys

A valley is an area surrounded by peaks and it is narrow in width and longer in length. Fundamentally, it is area with narrow change. The metaheuristic algorithms take longer time in these areas as local search tends to search these locations in detail. Unimodal and multimodal functions shape these valleys differently and in different frequency.

D. Separability

Separability defines the approach of optimizing variables of a function. Both unimodal and multimodal functions can be separable and non-separable as well.

Separable Functions: If the function is separable, each variable xi can be optimized independently. Functions of this category are easy to solve.

Non-Separable Functions: If the function is non-separable then all its variables have strong relationship with each other and cannot be optimized independently. Such functions are relatively difficult to solve.

E. Dimensionality

This property defines the area of search space. The larger the dimensionality, the larger is landscape, which means the larger number of sub-optimal locations. Functions with small dimensions are easily solved and most of the metaheuristic algorithms often perform best on these functions. However, for true performance evaluations, functions need to be highly dimensional.

III. Benchmark Functions Used In Literature

In literature, different sets of test functions have been employed while either introducing new metaheuristic algorithm or proposing any modified variant of existing method. Not only this, a variety of dimensional settings can be found in different literature. This implies that there exists no agreed test-bed with standard configurations, which may guide new researchers to perform experiments to validate their proposed methods. In fact, it is claimed that such diversified approaches indicate problem solving ability of an algorithm on a specific set of problems. However, most of the benchmark test functions are often similar in nature i.e., numerical optimization problems, or to be more specific, unconstrained numerical problems. It is also found that researchers have certain choice of functions which they often repeat in multiple experimental studies. This limited review surveyed published literature which performs experiments on benchmark test functions to evaluate performance of metaheuristic algorithms. Following is reported some of the work that indicates what number of test functions has been used with what configurations, so that an idea of existing approaches can be established.

Shi [6] used only 2 benchmark test functions (1 unimodal, 1 multimodal), but with three different dimensions D=10, 20, and 30, while introducing novel Brain Storm Optimization (BSO) algorithm. According to the researcher, the small test-bed was used as preliminary study to test the effectiveness of the proposed algorithm. In this study, the experimental results (mean, best, worst, and standard deviation) were averaged over 50 independent runs. In another experimental study, Karaboga and Gorkemli [7] tested the proposed Artificial Bee Colony (ABC) variant quick-ABC (qABC) on two unimodal and two multimodal test functions with 10 dimensions. The experiments were executed 30 times with random initialization. The results in the form of mean and standard deviations of objective functions values found over 30 runs were reported. In another ABC related research, Karaboga and Akay [8] compared the performances of ABC, Harmony Search, and Bees Algorithm on five test functions, two unimodals and three multimodals, with dimensions

ranging from 5 to 50. The researchers chose to present mean and standard deviation of 30 individual runs as performance evaluation matrix.

In a different experimental work, He, Wu and Saunders in [9] presented a novel swarm-based metaheuristic algorithm Group Search Algorithm (GSA) when tested against 6 benchmark test functions for convergence ability. Each function was solved with 30 dimensions and experimental results were averaged over 50 independent runs. Along with GSA, other evolutionary and swarm-based metaheuristic algorithms, including Genetic Algorithm (GA), Evolutionary Programming (EP), Evolution Strategies (ES), and PSO, were also tested on the same functions for comparisons based on mean and standard deviations of best values found over designated runs. Soliman and Rassem in [10] and Tang et al. in [11] both employed 8 test functions comprising of unimodal and multimodal categories with 30 dimensions. But, the number of runs in both the experiments was different, 30 in [10] and 100 in [11]. The prior work introduced Estimation of Distribution Algorithm (EDA), while Wolf Search Algorithm (WSA) was the proposed method in later research.

Yang in [12] and Yang and Deb in [13] used same set of benchmark test functions which comprised of 2 unimodal and 8 multimodal in nature. The number of dimensions (2 to 256) and number of runs (100) were same in both the experiments. The earlier introduced Bat Algorithm (BA), while the later work proposed Cuckoo Search (CS) algorithm. Both the methods were swarm-based metaheuristic algorithms. Interestingly, in both papers, the proposed algorithms were compared with GA and PSO in terms of mean and standard deviation values. The researches [14-19] also used ten test functions in experiments while introducing different metaheuristic algorithms with dimensions ranging from 2 to 30. For comparison with other algorithms, most of these works reported results in the form of mean and standard deviation values of best fitness values collected in 30 independent experiments.

Another novel swarm-based metaheuristic algorithm Chicken Swarm Optimization (CSO) was introduced by Meng et al. [20] and it was evaluated on 12 test functions which made up 8 unimodal and 4 multimodal functions. The functions were solved with 20 dimensions and statistics of results were mean, standard deviation, best, and worst of best fitness values found over independent experiments executed 100 times. Same test functions were also solved by PSO, differential evolution (DE), and BA for comparison with the proposed CSO, as the proposed algorithm outperformed the counterparts. Zhang et al. [21] also performed experiments on almost the similar number of functions as in previous work while proposing a hybrid Firefly Algorithm (HFA). This research used 13 functions including 6 unimodal and 7 multimodal functions with 30 dimensions and equal number of experimental runs. As usual in such type of research, this work also presented mean, standard deviation, best, and worst fitness values found over designated runs. In comparison with PSO, DE and FA, the proposed HFA produced promising results. The equal number of test functions were employed by Zheng et al. [22] but this time unimodal functions were 7 and 6 were the multimodal functions. This research proposed swarm-based

metaheuristic algorithm so-called Ecogeography-Based Optimization (EBO) and compared with DE and Biogeography-Based Optimization (BBO) algorithms on these 13 functions with variety of dimensions: 10, 30, and 50. The mean and standard deviation of 60 runs were reported as statistics.

The test-bed of Gandomi and Alavi [23] was majorly formulated by multimodal test functions (15) as compared to unimodal functions (5). These 20 benchmark functions were employed to evaluate the performance of the proposed novel bio-inspired Krill Herd (KH) metaheuristic algorithm. For robust statistical results, the experiments were conducted on test functions with 2, 4, 6, and 20 dimensions, and executed 50 times. A wider range of counterpart metaheuristic algorithms were compared, which included PSO and DE variants as well as BBO, ACO, ES, and ACO. Cui at el. [24] and Xiang et al. [25] tested the proposed ABC variants on more than 20 test functions with dimensions 30, 50, and 100. These researches used 9 and 11 unimodal respectively, and 13 and 15 multimodal functions respectively. The algorithms in both the experiments were executed around 30 times and mean and standard deviation of fitness values were reported for comparison.

Vortex Search Algorithm (VSA) was proposed and evaluated on adequately large test-bed of 50 benchmark functions by Doğan and Ölmez [26]. Multimodal functions were more than double in number 34 to 16 unimodal functions. The number of dimensions ranged from 2 to 30 and the experiments were run 30 independent times. As commonly found norm, this research also compared results in the form of mean, standard deviation, and best fitness values found over the specified runs.

Based on limited survey performed for this research, as a summary, it can be established that minimum 2 test functions were used and at maximum 50 functions were employed. However, it is noteworthy to mention that sometimes, it is also found in research that around 100 test functions have also been used – though it is rear. Other than generalized functions, researchers have also shifted and rotated these function to increase complexity and evaluating metaheuristic algorithms more authentically. On average, the test-bed contained around 13 benchmark functions with 5 unimodal and 8 multimodal functions. The common test-bed size was 10 and commonly used dimensions are 10, 30, and 50. Mostly, the results were averaged on 30 independent runs with different random initializations, minimum and maximum runs found were 20 and 100, respectively. These statistics of survey summary are presented in Table I, where min., max., avg., com. are minimum, maximum, average and commonly used settings, respectively.

TABLE I
COMMONLY USED EXPERIMENTAL SETTINGS

Item	Min.	Max.	Avg.	Com.
Test-bed size	2	50	13	10
Unimodal functions	1	16	5	1,4
Multimodal functions	1	34	8	6
Dimensions	2	256	26	30, 50
Experimental Runs	20	100	49	30

220

IV. COMMONLY USED BENCHMARK FUNCTIONS

Since, it is already established in previous section that despite diversity in test-beds used by different researchers to evaluate the proposed methods, there is found a commonality in the experimental studies performed in literature. This is highly useful for new researchers to determine the type of test-bed based on guidelines provided by experimental studies performed by previous researchers. Keeping this in view, Table II presents 20 commonly used unimodal (U) and multimodal (M) benchmark functions with different dimensions of variables including minimum (Min. D), maximum (Max. D), and mostly used (Com. D) dimensional settings.

TABLE II
COMMONLY USED EXPERIMENTAL SETTINGS

Function	Nature	Max. D	Min. D	Com. D
Sphere	U	256	3	30
Rosenbrock	U	100	5	30
Quartic	U	100	10	30
Step	U	100	5	30
Schwefel 2.22	U	100	10	30
SumSquare	U	100	10	30
Elliptic	U	100	10	30
Rastrigin	M	100	2	30
Griewank	M	100	2	30
Ackley	M	128	5	30
Michalewicz	M	100	2	10
Penalized1	M	100	10	30
Schwefel	M	128	10	30
Weierstrass	M	100	10	30
Non-Continuous Rastrigin	M	100	10	30
Penalized2	M	100	10	30
Schwefel 2.26	M	100	10	30
Schaffer	M	50	2	2
Alpine	M	100	10	30
Himmelblau	M	200	30	50

The detail of the functions given above is as follow. In following functions D is referred to as problem dimensions and x_i is a problem variable, whereas x^* is optimum solution.

1) *Sphere*: Sphere is a unimodal and continuous function, which is considered as easy to solve. This function is evaluated using range between [-5.12,5.12] and its minimum solution is *0* which is located at $f(x^*)=[0,0,...,0]$. It is mathematically expressed as:

$$f(x) = \sum_{i=1}^{D} x_i^2$$

2) *Rosenbrock:* This is also a unimodal function which is also known as banana function as its global minimum solution *0* is found in the narrow valley, with optimum solution $f(x^*)=[0,0,...,0]$. The range of values of parameter values is often set to [-5, 10]. Mathematically, the function is expressed as:

$$f(x) = \sum_{i=1}^{D-1} [100(x_{i+1} + x_i^2)^2 + (x_i - 1)^2]$$

3) *Quartic:* This function is a unimodal function like Sphere with degree *4*. The best solution *0* is found at $f(x^*)=[0,0,...,0]$, whereas the search space is spread between [-1.28,1.28]. This function is expressed as:

$$f(x) = \sum_{i=1}^{D} ix_i^4$$

4) *Step:* This function represents flat surface which is often considered as difficult to solve as no proper direction towards globally optimum location is easily found. Step is a unimodal function where minimum solution *0* located at $f(x^*)=[0,0,...,0]$ within the values spread over [-100,100] range.

$$f(x) = \sum_{i=1}^{D} (x_i + 0.5)^2$$

5) *Schwefel 2.22:* This function is a unimodal with search space usually spread over [-10,10] values. The global minimum 0 is located at $f(x^*)=[0,0,...,0]$.

$$f(x) = \sum_{i=1}^{D} |x_i| + \prod_{i=1}^{D} |x_i|$$

6) *SumSquare:* This function is also known as Axis Parallel Hyper-Ellipsoid function which maintains no local optima but single global optima $f(x^*)=[0,0,...,0]$. The function is normally evaluated with continuous values within the range of [-10,10]. The function expressed as:

$$f(x) = \sum_{i=1}^{D} ix_i^2$$

7) *Elliptic:* This unimodal function with single global best solution *0* found at $f(x^*)=[0,0,...,0]$ within the search space of [-100,100]. The function expressed as:

$$f(x) = \sum_{i=1}^{D} (10^6)^{(i-1)/(D-1)} x_i^2$$

8) *Rastrigin:* This multimodal function is difficult to solve as it presents numerous local minima locations where an optimization algorithm, with poor explorative capability, has high chances of being trapped. The function's only globally best solution *0* is found at $f(x^*)=[0,0,...,0]$ within the domain of [-5.12,5.12]. The function mathematically written as:

$$f(x) = \sum_{i=1}^{D} (x_i^2 - 10 \cos(2\pi x_i) + 10)$$

9) *Griewank:* It is a multimodal function with widespread suboptimal solutions spread all over the search environment. This function has one global optimum solution *0* to be located at $f(x^*)=[0,0,...,0]$. The function is solved with range of [-600,600].

$$f(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

10) *Ackley:* This multimodal function is one of the most commonly used test function for metaheuristic algorithm evaluation. It has numerous local minima but one global optimal solution found in deep narrow basin in the middle. The best solution *0* is found at $f(x^*)=[0,0,...,0]$ in domain [-32,32].

$$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i\right)$$

11) *Michalewicz:* It is another multimodal function used to test global search ability of a metaheuristic algorithm on high dimensional functions with small number of global optimum in the search space. This function is solved with $0 \le x_i \le \pi$. The globally best solution with *D=2* is *-1.8013*, *-4.687658* with *D=5*, and *-9.66015* with *D=10*. Mathematical expression of Michalewics is as follow:

$$f(x) = -\sum_{i=1}^{D}\sin(x_i)\left[\sin\left(\frac{ix_i^2}{\pi}\right)\right]^{2m}, m = 10$$

12) *Penalized1:* Sometimes also referred to as Generalized Penalized-1 function is a difficult multimodal function. Its global optimum *0* is located at $f(x^*)=[1,1,...,1]$ in the domain of [-50,50]. Mathematically, Penalized1 function is expressed as following, where the values of parameters are often set as *k=100*, *a=10*, and *m=4*.

$$f(x) = \frac{\pi}{D}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_D - 1)^2\right\} + \sum_{i=1}^{D}u_i,$$

$$u_i = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & a \le x_i \le a \\ k(-x_i - a)^m & x_i < -a \end{cases}$$

$$y = 1 + (x_i + 1)/4$$

13) *Schwefel*: This function is one of those hard multimodal function that are difficult to solve, as it contains several local minima locations, however global minima *0* is located at $f(x^*)=[1,1,...,1]$ in the domain of [-500,500]. Mathematically, this function is expressed as:

$$f(x) = \sum_{i=1}^{D} -x_i \sin(\sqrt{|x_i|})$$

14) *Weierstrass*: It is another multimodal function which is solved in the domain of [-0.5,0.5] and optimum value *0* is located at $f(x^*)=[0,0,...,0]$. Mathematically, this function is expressed as:

$$f(x) = \sum_{i=1}^{D}\left(\sum_{k=0}^{k_{max}}\left[a^k\cos\left(2\pi b^k(x_i + 0.5)\right)\right]\right) - \sum_{k=0}^{k_{max}}[a^k\cos(2\pi b^k \times 0.5)]$$

15) *Non-Continuous Rastrigin*: This is another multimodal function for finding minimized solution *0* at $f(x^*)=[0,0,...,0]$ in domain [-5.12,5.12]. The functions is given below:

$$f(x) = \sum_{i=1}^{D}(y_i^2 - 10\cos(2\pi y_i) + 10),$$

$$y_i = \begin{cases} x_i & \text{if } |x_i| < 0.5, \\ \dfrac{round(2x_i)}{2} & \text{else } |x_i| \ge 0.5. \end{cases}$$

16) *Penalized2*: This is multimodal function for finding minimized solution *0* at $f(x^*)=[0,0,...,0]$ in domain [-5.12,5.12]. The function is given below:

$$f(x) = \sum_{i=1}^{D}u_i + 0.1\left\{\begin{array}{l} \sin^2(3\pi x_i) + (x_D - 1)^2 \\ + \sum_{i=1}^{D-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] \end{array}\right\}$$

$$u_i = \begin{cases} |x_i| < 50 \\ k(x_i - a)^m & x_i > a \\ 0 & a \le x_i \le a \\ k(-x_i - a)^m & x_i < -a \end{cases}$$

17) *Schwefel 2.26*: This variant of Schwefel is a multimodal function with best solution *0* at $f(x^*)=[0,0,...,0]$ in domain [-500,500]. The function is written as:

$$f(x) = -418.98288727243369 \times D + \sum_{i=1}^{D}\left[-x_i\sin\left(\sqrt{|x_i|}\right)\right]$$

18) *Schaffer:* Schaffer is some multimodal function where global minima is either small or close to local minima. Hence, this requires a metaheuristic algorithm to have efficient search strategy to solve this problem. The best solution *0* is found at $f(x^*)=[0,0,...,0]$ in the range of [-100,100]. The function is mathematically defined as:

$$f(x) = 0.5 + \frac{\sin^2\left(\sqrt{x_1^2 + x_2^2}\right) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$$

19) *Alpine:* It is a multimodal function where global minima *0* is found at $f(x^*)=[0,0,...,0]$ in the range of [-10,10]. The function is written as:

$$f(x) = \sum_{i=1}^{D-1}|x_i\sin(x_i) + 0.1x_i|$$

20) *Himmelblau:* This is a multimodal function. It is usually solved with continuous values in the domain of [-6,6]. The best solution *0* can be found at four locations: $f(x^*)=[3.2,2.0]$, $f(x^*)=[-2.805118,3.131312]$, $f(x^*)=[-$

3.779310,-3.283186], and $f(x^*)=[3.584428,-1.848126]$ in 2 dimensional space. The function is defined as:

$$f(x) = \frac{1}{D} \sum_{i=1}^{D} (x_i^4 - 16x_i^2 + 5x_i)$$

V. Conclusion

Benchmark numerical test functions have been widely used as performance evaluation tools in metaheuristic literature. This is also considered that metaheuristic algorithms that perform well on these functions are able to solve real-life hard optimization problems. It is found that different researchers choose different sets of functions with different experimental configurations, yet there exist common experimental settings that may guide other researchers to follow the trend and gauge robustness of any newly proposed metaheuristic algorithm. In this regard, this survey provided a list of commonly used benchmark test functions, including unimodal and multimodal, which can be readily employed and enhanced by rotating and shifting the listed functions, for extended complexity.

References

[1] R. W. Garden and A. P. Engelbrecht, "Analysis and classification of optimisation benchmark functions and benchmark suites," in Proc. IEEE CEC 2014, pp. 1641-1649.

[2] M. Jamil and X. S. Yang, "A literature survey of benchmark functions for global optimisation problems," International Journal of Mathematical Modelling and Numerical Optimisation., vol. 4, pp. 150-194, Jan. 2013.

[3] S. Surjanovic and D. Bingham. (2013) Virtual library of simulation experiments: test functions and datasets. [Online]. Available: http://www. sfu. ca/~ ssurjano/optimization. html

[4] A. Rehman. (2017) Global Optimization Test Problems. [Online]. http://www.optima.amp.i.kyotou.ac.jp/member/student/hedar/Hedar_files/TestGO.htm

[5] (2017) IEEE Congress on Evolutionary Computation (CEC). [Online].Available:http://www.ieee.org/conferences_events/conferences/conferencedetails/index.html?Conf_ID=38257

[6] Y. Shi, "Brain storm optimization algorithm," in Proc. International Conference in Swarm Intelligence 2011, pp. 303-309.

[7] D. Karaboga and B. Gorkemli, "A quick artificial bee colony-qABC-algorithm for optimization problems," in Proc. International Symposium on Innovations in Intelligent Systems and Applications (INISTA) 2012, pp. 1-5.

[8] D. Karaboga and B. Akay, "Artificial bee colony (ABC), harmony search and bees algorithms on numerical optimization," in Proc. Innovative production machines and systems virtual conference 2009.

[9] S. He, Q. H. Wu, and J. R. Saunders, "A novel group search optimizer inspired by animal behavioural ecology," in Proc. IEEE Congress on Evolutionary Computation (CEC 2009)2009, pp. 272-1278.

[10] O. S. Soliman and A. Rassem, "A bio inspired estimation of distribution algorithm for global optimization," in Proc. International Conference on Neural Information Processing 2012, pp. 645-652.

[11] R. Tang, S. Fong, X. S. Yang, and S. Deb, "Wolf search algorithm with ephemeral memory," in Proc. IEEE Seventh International Conference on Digital Information Management (ICDIM) 2012, pp. 165-172.

[12] X. S. Yang, "A new metaheuristic bat-inspired algorithm," Nature inspired cooperative strategies for optimization (NICSO 2010), 2010.

[13] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in Proc. IEEE World Congress on Nature & Biologically Inspired Computing (NaBIC) 2009, pp. 210-214.

[14] J. A. Koupaei, M. M. S. Hosseini, and F. M. Ghaini, "A new optimization algorithm based on chaotic maps and golden section search method," Engineering Applications of Artificial Intelligence, vol. 50, pp. 201-214, Apr. 2016.

[15] R. Rahmani and R. Yusof, "A new simple, fast and efficient algorithm for global optimization over continuous search-space problems: Radial Movement Optimization", Applied Mathematics and Computation, vol. 248, pp. 287-300, Dec. 2014.

[16] W. Li, L. Wang, Q. Yao, Q. Jiang, L. Yu, B. Wang, and X. Hei, "Cloud particles differential evolution algorithm: a novel optimization method for global numerical optimization," Mathematical Problems in Engineering, Dec. 2015.

[17] L. Wen, L. Gao, X. Li, and L. Zhang, "Free Pattern Search for global optimization," Applied Soft Computing, vol. 13(9), pp. 3853-3863, Sep. 2013.

[18] R. Q. Zhao and W. S. Tang, "Monkey algorithm for global numerical optimization," Journal of Uncertain Systems, vol. 2(3), pp. 165-176, 2008.

[19] M. A. Munoz, J. A. López, and E. Caicedo, "An artificial beehive algorithm for continuous optimization," International Journal of Intelligent Systems, vol. 24(11), pp. 1080-1093, Nov. 2009.

[20] X. Meng, Y. Liu, X. Gao, and H. Zhang, "A new bio-inspired algorithm: chicken swarm optimization," in Proc. International Conference in Swarm Intelligence 2014, pp. 86-94.

[21] L. Zhang, L. Liu, X. S. Yang, and Y. Dai, "A novel hybrid firefly algorithm for global optimization," PloS One, vol. 11(9), Sep. 2016.

[22] Y. J. Zheng, H. F. Ling, and J. Y. Xue, "Ecogeography-based optimization: enhancing biogeography-based optimization with ecogeographic barriers and differentiations," Computers & Operations Research, vol. 50, pp. 115-127, Oct. 2014.

[23] A. H. Gandomi and A. H. Alavi, "Krill herd: a new bio-inspired optimization algorithm," Communications in Nonlinear Science and Numerical Simulation, vol. 17(12), pp. 4831-4845, Dec. 2012.

[24] L. Cui, G. Li, Z. Zhu, Q. Lin, Z. Wen, Z. Lu, … and J. Chen, "A novel artificial bee colony algorithm with an adaptive population size for numerical function optimization," Information Sciences, 2017.

[25] W. L. Xiang, Y. Z. Li, X. L. Meng, C. M. Zhang, and M. Q. An, "A grey artificial bee colony algorithm," Applied Soft Computing, Jun. 2017.

[26] B. Doğan and T. Ömez, "A new metaheuristic for numerical function optimization: Vortex Search algorithm," Information Sciences, vol. 293, pp. 125-145, Feb. 2015.