Expert Systems With Applications 207 (2022) 117993

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Sine–Cosine-Barnacles Algorithm Optimizer with disruption operator for global optimization and automatic data clustering

Mohamed Abd Elaziz [a,b,c], Ahmed A. Ewees [d,e,*], Mohammed A.A. Al-qaness [f,*], Laith Abualigah [g,h,i], Rehab Ali Ibrahim [c]

[a] Faculty of Computer Science & Engineering, Galala University, Suze 435611, Egypt, Suze, Egypt
[b] Artificial Intelligence Research Center (AIRC), Ajman University, Ajman 346, United Arab Emirates, Ajman, United Arab Emirates
[c] Department of Mathematics, Faculty of Science, Zagazig University, Zagazig, Egypt
[d] Department of Information System, Faculty of Computing and Information Technology, University of Bisha, Saudi Arabia
[e] Department of Computer, Damietta University, Egypt
[f] State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China
[g] Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan
[h] School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang 11800, Malaysia
[i] Faculty of Information Technology, Middle East University, Amman 11831, Jordan, Amman, Jordan

ARTICLE INFO

Keywords:
Optimization techniques
Swarm-intelligence
Sine–Cosine Algorithm
Barnacles mating optimizer
Exploration and exploitation

ABSTRACT

In this paper, an improved Barnacles Mating Optimizer (BMO) is proposed to deal with optimization problems and develop a new automatic clustering approach. BMO is a well-established optimization technique inspired by the mating behavior of barnacles in real-life. The exploratory trends of BMO are influential and can maintain the right balance among exploration and exploitation. However, this population-based method can be improved further to reduce the probability of potential drawbacks for any optimization technique. As such, we revised the core searching phased of BMO based on a sine–cosine algorithm (SCA) and disruption operators (DO). The proposed method is named BMSCD, which updates the current solution by switching between the mechanisms of the BMO and SCA based on a probability calculated using the fitness value of the current solution. The experiments results on various benchmark cases for global optimizations demonstrate the improved performance of the proposed BMSCD in terms of quality of solutions, the balance of the exploration–exploitation, and convergence rates. Besides, the proposed BMSCD is evaluated by nine measures in solving different clustering problems. The results show that the BMSCD can effectively and powerfully address the tested problems and provide excellent performance compared to the state-of-the-art methods.

1. Introduction

Optimization problems have become popular and challenging in the last years due to their heterogeneous nature and many forms of robust optimization (Qu, Han, Wu, & Raza, 2020), multiobjective (Cao et al., 2019), single objective (Liu, Wu, Wu, & Wang, 2015), large scale optimization (Abualigah & Diabat, 2020; Cao et al., 2020b), many objective (Cao et al., 2020a; Cao, Wang, Zhang, Song, & Lv, 2020c), fuzzy optimization (Chen, Qiao, Xu, Feng, & Cai, 2019b), and memetic optimization (Fu, Pace, Aloi, Yang, & Fortino, 2020). Such problems were remarked in various domains varying from computer science and process optimization (Liu, Lv, Yi, & Xie, 2019), optimal control (Precup, David, Roman, Szedlak-Stinean, & Petriu, 2021; Rigatos, Siano, Selisteanu, & Precup, 2017), and industrial engineering applications to finance utilization (Abualigah, 2020b; Ewees, Elaziz, & Houssein, 2018). Several recent meta-heuristic algorithms (MAs) are proposed to address such optimization problems. Such algorithms are characterized by their ability to solve challenging problems despite being mostly stochastic and not deterministic (Xu et al., 2019a, 2019b). In real-world applications, decision-makers seek intelligent systems to play their role as assistance by finding and giving optimal sets of decisions (Xia et al., 2017). These decisions can be very vital and for the early detection of diseases, and this can change the health conditions of patients (Chen et al., 2016). These Such problems often appear both in continuous and binary problems such as various types of automated clustering, feature selection (Zhang, Fan, Wang, Zhou, & Tao, 2020a; Zhao et al., 2015), face recognition (Wang, Chen, Yan, Chen, & Fu, 2014a; Zhang, Wang,

* Corresponding authors.
E-mail addresses: abd_el_aziz_m@yahoo.com (M.A. Elaziz), ewees@du.edu.sa (A.A. Ewees), alqaness@whu.edu.cn (M.A.A. Al-qaness), aligah.2020@gmail.com (L. Abualigah), rehab100r@yahoo.com (R.A. Ibrahim).

https://doi.org/10.1016/j.eswa.2022.117993
Received 9 November 2021; Received in revised form 24 April 2022; Accepted 26 June 2022
Available online 1 July 2022

Zhou, & Ma, 2019), image processing (Zhang, Wang, Wang, Tang, & Zhao, 2020c), and neural networks (Zhang, Jiang, Wang, & Wang, 2020b). These problems get more complicated when we also need to perform pattern recognition cases (Liu et al., 2016) and medical systems (Li et al., 2018). Because in such cases, the final clinical decisions directly affect the health conditions of patients (Hu, Hong, Ma, Wang, & Chen, 2015), and this further shows the importance of efficacy and performance of the involved algorithms. Usually, any problem requires optimal decision-making to be addressed effectively by determining the smallest cost value or obtaining the highest profit ratio (Osman & Laporte, 1996). Meta-heuristic optimization algorithms are standard procedures that handle various classes of different problems (Abualigah, 2020a). They can be utilized to any problem using several search methods affected by particular factors to examine wide-area in the search regions and utilize the local regions (Shen et al., 2016; Wang & Chen, 2020). According to this reality, various intelligent search methods such as crossover operator, gaussian mechanism, levy flight distribution, and other algorithms have been applied in meta-heuristic optimization to develop their performance abilities, as mentioned by Ibrahim, Ewees, Oliva, Elaziz, and Lu (2019), Ibrahim, Oliva, Ewees, and Lu (2017), Wang et al. (2017), Xu and Chen (2014). Optimization-based methods are categorized into evolutionary algorithms (EAs), swarm algorithms (SAs). Many swarm-intelligence methods have been developed such as trajectory optimization algorithms (TAs) (Abualigah, Khader, & Hanandeh, 2019), Dwarf Mongoose Optimization Algorithm (DMOA) Agushaka, Ezugwu, and Abualigah (2022), Sine Cosine Algorithm (SCA) (Mirjalili, 2016), Group Teaching Optimization Algorithm (Zhang & Jin, 2020), Ebola Optimization Search Algorithm (EOSA) Oyelade, Ezugwu, Mohamed, and Abualigah (2022), Reptile Search Algorithm (RSA) Abualigah, Abd Elaziz, Sumari, Geem, and Gandomi (2022a), Political Optimizer (Askari, Younas, & Saeed, 2020), Tunicate Swarm Algorithm (Kaur, Awasthi, Sangal, & Dhiman, 2020), Arithmetic Optimization Algorithm (Abualigah, Diabat, Mirjalili, Abd Elaziz, & Gandomi, 2020), and Aquila Optimizer (Abualigah et al., 2021).

BMO is a novel bio-inspired optimization algorithm introduced for solving engineering optimization problems proposed in Sulaiman, Mustaffa, Saari, and Daniyal (2020). The inspiration of BMO is motivated by the mating behavior of barnacles in real-life. In this approach, Barnacles are born with the ability to float, but as they attain adulthood, they tie themselves to items in the water and develop a shell. This method mathematically simulates such changing models and behaviors to reveal an algorithm. Due to its robust searching capability and ease of implementation, BMO has been used to address various optimization problems since the proposal.

Certain aspects make optimization problems hard to be solved. In reality, the complexity of the search areas increases exponentially as the character of variables raises, which is the primary problem in this domain (Abualigah & Diabat, 2021). Despite, there is no hypothesis to confirm which method can obtain the most reliable and stable performance; certainly, modified algorithms increase the performance of the conventional methods (Sulaiman et al., 2018). BMO has bestowed superior performance for addressing various problems with complicated structures. However, two primary defects are recognized in the performance operation of the conventional BMO: lack of diversity on the candidate solutions, which makes enduring premature convergence and fast acceleration in the convergence rule. Because of these vulnerabilities, BMO demands further development as modifying or hybridizing with other search operators to evade early convergence, enhance the diversity of the solutions, and improve its achievement. Consequently, an enriched approach, disruption operator-based on BMO-SCA, is introduced and examined using various optimization problems and real-world clustering problems. The main target of using SCA is to improve the exploration ability of BMO. At the same time, the disruption operator is applied to improve the diversity of the agents during the updating stage. The primary motivation bind using a new operator in the proposed method is to avoid the risen weaknesses in the original method and make it more efficient during the optimization process.

The following are the key contributions in this paper:

- A novel approach based on disruption operator is designed utilizing BMO and SCA to equilibrium exploration and exploitation search methods.
- Different benchmark functions and benchmark clustering datasets are employed to evaluate the proposed method's performance.
- Experiments demonstrated that the suggested technique can handle the underlying optimization problems effectively and probably in most situations, resulting in improved performance.

The remaining sections of this paper are structured as follows. Section 2 reviews some main related works. Section 3 introduces the study's algorithms. The proposed approach is presented in Section 4. Section 5 contains the experiment results and the discussion for global optimization and clustering problems. The conclusion and future works are listed in Section 7.

2. Related works

Due to the novelty, performance, and robustness of BMO, many papers have been devoted to studying the features of this optimizer. Researchers utilized some enhanced versions of BMO to understand some challenging cases. For instance, to realize the photovoltaic systems, a hybrid BMO with mutation and flower pollination method was developed by Sulaiman and Mustaffa (2021). A multi-population BMO proposed by Sulaiman, Mustaffa, and Aliman (2019a.) that mixes the advantages of BMO with the well-known differential evolution mechanisms for reaching a better quality of results. Also, a new enhanced method based on using BMO to solve the proton exchange membrane fuel cell and Increasing the precision and robustness of the system (Yang, Liu, Zhang, Dai, & Razmjooy, 2020). Also, researchers developed a structure for BMO to deal with Optimal Power Flow (Sulaiman & Mustaffa, 2021). The image contrast enhancement is treated as an optimization challenge in (Ahmed, Ghosh, Bera, Schwenker, & Sarkar, 2020) and a new version of BMO is used to find the best solution. To transform an image into a solution to the optimization problem, a gray-level mapping technique is used. There are several other applications such as (Al-qaness, Ewees, Fan, Abualigah, & Abd Elaziz, 2022; Bojan-Dragos et al., 2021; Houssein, AbdElminaam, Hassan, Al-Sayed, & Nabil, 2021; Precup et al., 2020; Sulaiman et al., 2019b; Tan, Ooi, Leong, & Lin, 2014).

SCA is a swarm-based optimization algorithm introduced for tackling optimization problems (Mirjalili, 2016). The SCA is very similar to the grey wolf optimizer (GWO), and it generates the initial population randomly and guides them to search about the best agent according to the sine and cosine functions. Neggaz, Ewees, Abd Elaziz, and Mafarja (2020) developed a modified salp swarm algorithm (SSA) using SCA. The developed method is applied as a feature selection. Wan, Ma, Zhong, Hu, and Zhang (2020), a multiobjective hyperspectral FS according a modified version SCA. Pashiri, Rostami, and Mahrami (2020), an enhanced artificial neural network (ANN) based on SCA and applied it as a spam detection method. Gupta, Deep, and Engelbrecht (2020) presented a modified SCA algorithm using a memory-guided strategy and applied global optimization. Jiang, Han, Meng, and Li (2020), the tree–seed algorithm is enhanced using the operators of SCA and then applied the developed method as global optimization. Al-Qaness, Elaziz, and Ewees (2018) utilized the SCA to improve the Adaptive Neuro-Fuzzy Inference System for forecasting oil consumption. Authors in Chen et al. (2019a) developed an opposition-based SCA with a new local search for parameter detection of photovoltaic models. Attia, El Sehiemy, and Hasanien (2018) enhanced the performance of SCA and applied it to find the solve power flow problem. Lin et al. (2019)

introduced a chaotic SCA-based fuzzy K-Nearest neighbor classification method for predicting the intentions of master students. Tawhid and Savsani (2019) presented a multi-objective method according to SCA for solving engineering design problems. Also, Jouhari et al. (2019) applied SCA to solve machine scheduling problems with setup times. Mahdad and Srairi (2018) proposed a version of SCA to improve the power usage security based on the loading margin balance and errors. Zhou et al. (2020) proposed a multi-core SCA, and they revealed a deep analysis of all aspects of the performance of their method. Fan et al. (2020) proposed a boosted fruit fly optimizer known as SCA_FOA by designing the logic of SCA, and they show the results are superior to the basic SCA. Chen, Heidari, Zhao, Zhang, and Chen (2020) developed a version of SCA with the orthogonal learning, multi-swarm, and greedy selection ideas and they did a comprehensive analysis. An improved SCA algorithm based on the chaos-based intensification mechanism and Levy flight idea proposed by Huang et al. (2020). Zhu et al. (2020) has integrated SCA with orthogonal learning scheme, and then they optimized the model of extreme learning machine for assessment of Sino foreign cooperative education projects.

Moreover, the diversity of solutions plays an essential role in improving the optimization process (Fathy, Elaziz, Sayed, Olabi, & Rezk, 2019). Hence, the disruption operator (DO) is utilized to sustain the diversity of solutions and enhances the exploration search-ability. This operator initially examines the available search space, and as time progresses, it turns to the exploiting condition (Ibrahim, Elaziz, & Lu, 2018b; Sharma, Sharma, Sharma, & Bansal, 2016). According to these properties, DO has been used to improve the efficiency of several meta-heuristics. For example, Ibrahim et al. (2018b) developed the GWO using DO and other techniques such as DE and OBL. It has been found that the DO has a more massive effect on the performance of COGWO2D than the other components. Kaur, Sharma, and Sharma (2017) combined the disruption operator with spider monkey optimization (SMO) to solve the convergence problem. The results using the benchmark functions showed that the proposed method able to avoid the convergence problem. Sarafrazi, Nezamabadi-Pour, and Saryazdi (2011), present a new method to improve exploration and exploit abilities of the gravitational search algorithm (GSA). This method added the disruption operator into the main GSA procedures. The results showed that the proposed algorithm using the disruption operator got high performance compared to other similar algorithms as agreed with (Gouthamkumar, Sharma, & Naresh, 2015). Also, Liu, Ding, and Sun (2012) added the disruption operator in GSA to improve its ability alongside with opposition-based learning population. The results showed the stability of the proposed method in solving benchmark problems. Ding, Liu, and He (2013) proposed a new method to solve the premature convergence in particle swarm optimization (PSO) using the disruption strategy. The proposed method is tested using real-world optimization problems, and the results showed that it got high-performance values. Zhang et al. (2018) proposed a new method called SADCPSO to solve multi-task scheduling problems. This method integrated the disruption operator and chaos operator into the PSO algorithm. The results showed that the SADCPSO got better quality solutions.

In the literature, the majority of meta-heuristic algorithms can get stuck in local optimal and suffer from low convergence velocity, so several studies were performed to handle these issues (Zhao et al., 2019). For instance, Sayed, Khoriba, and Haggag (2018) presented a modified salp swarm algorithm (SSA) based on chaos theory. This algorithm named CSSA and it is evaluated using fourteen benchmark problems and applied as feature selection method to assess twenty datasets. The results illustrated that the CSSA is promising. Zhao et al. (2014) proposed an effective feature selection method using ant colony optimization (ACO) for fiber recognition in cotton. Aydilek (2018) combined fireflies and PSO to form new method named HFPSO. The combinations aimed to manage the local search by comparing the current global best fitness value. The HFPSO is faster than other compared algorithms. Ewees et al. (2018) developed an improved grasshopper optimization algorithm (GOA) using the opposition-based learning (OBL) and it is named as OBLGOA. The OBLGOA used OBL to generate initial solutions as well as the update solutions during the searching process. The comparative results illustrated that the OBLGOA provides better performance than other methods.

Data clustering has been another popular area in recent years, which is considered as a challenging optimization problem (Abualigah, 2019). A large number of meta-heuristics have been employed to find the optimal solution for the clustering problem. For example, Kuo, Amornnikun, and Nguyen (2020) introduced a possibilistic multivariate fuzzy weighted c-means algorithms based on genetic algorithm (GA), PSO, and SCA. To assess these methods different a real-world dataset certainly from UCI machine learning repository. The results illustrate that the SCA-PMFWCM, GA-PMFWCM and PSO-PMFWCM algorithms are better than other methods. Valdivia et al. (2020) presented a clustering-Based binarization using crow search algorithm. In Mousavirad, Ebrahimpour-Komleh, and Schaefer (2020), developed an automatic clustering method based on improved human mental search algorithm and applied it to image segmentation. A modified version of grey wolf optimization for clustering data (Aljarah, Mafarja, Heidari, Faris, & Mirjalili, 2020). In this clustering method, the GWO is enhanced using tabu search (TS) and a set of 13 datasets clustering.

In Yousri et al. (2022) a modified manta ray foraging optimizer was developed based on discrete fractional-order Caputo techniques to solve global optimization problems, as well as image segmentation. As concluded, the applications of the discrete fractional-order Caputo technique have significant improvements to the performance of the original MRFO method. In Abualigah et al. (2022c), a hybrid of AOA and SCA was developed to solve different engineering problems. The main idea is to employ the SCA search strategy to boost the search mechanism of the traditional AOA. In Abd Elaziz et al. (2021b), the MRFO was booted using the fractional-order (FO) calculus in the exploitation stage. The heredity and non-locality characteristics of the Grunwald–Letnikov fractional differ-integral operator is employed for simulating the manta rays' motion after the effect of their previous location and future movement directions. In Abd Elaziz et al. (2021a), a new solution for global optimization problems was developed using cooperative optimization methods, including GWO, DE, SSA, WOA, SOS, and SCA.

In the same context, the proposed BMSCD is implemented as an automatic clustering method. Since the BMSCD method has a higher potential to avoid stagnation situations, and this helps to determine the optimal number of clusters and the corresponding centers. Also, the proposed BMSCD method starts with higher disruption-based exploration potential, and according to the time-varying component of BMO, it can perform a smooth changeover phase. Indeed, it is able to emphasize on exploitation stage during the last steps efficiently. Such a performance can lead to better convergence and effectiveness of the clustering process when dealing with various regions of the feature space. In case of stagnation into local clustering solutions, the BMSCD can feasibly avoid them and scan other potential spaces. Hence, the resulted clustering method can show a more stable searching procedure with a satisfying convergence behavior.

3. Preliminaries

In this section, the background for the data clustering problem, Barnacles Mating Optimizer, Sine–Cosine Algorithm, and Disruption Operator is discussed.

3.1. Data clustering problem

The challenge of data clustering may be expressed as a challenging optimization problem. Whereby each cluster logically comprises the more similar nature things, whereas the other clusters contain a variety of distinct data objects (Abualigah, Almotairi, Abd Elaziz, Shehab, & Altalhi, 2022b; Ezugwu et al., 2022).

Normally, a set of data items is used to solve this problem ($\boldsymbol{D} = \{o_1, \ldots, o_i, \ldots, \ldots, o_n\}$), which is subdivided into a certain number of clusters ($\boldsymbol{K} = \{c_1, \ldots, c_i, \ldots, \ldots, c_K\}$). Each object ($o$) contains a collection of length (t) instances, each with its own weight value (w). The supplied $\boldsymbol{D}$ comprises a number of objects (n), each with a unique value that corresponds to the specified dataset (o_i). The distance between the data items and the cluster centroid is calculated using the centroid (c_i) of each cluster. As a result, the clustering procedure assigns each object (o) to a certain cluster in order to reduce the Euclidean distance (as shown in Eq. (1)) between the given objects and the clusters' centroids.

$$ED(o,c) = \sum_{i=1}^{n} \sum_{j=1}^{K} w_{ij} \|o_i - c_j\|^2 \tag{1}$$

The distance between the i_{th} object and the j_{th} cluster centroid is measured by $\|o_i - c_j\|$. The overall number of utilized items in the D is n, and the number of clusters is K. The value of the i_{th} item connected to the j_{th} cluster, being either 1 or 0, is represented by w_{ij}. If the i_{th} item is pationed to the j_{th} cluster, w_{ij} is 1 and if not, w_{ij} is 0 Almotairi and Abualigah (2022).

3.2. Barnacles Mating Optimizer (BMO)

BMO is a recently MH algorithm inspired by the natural behavior of the mating behavior of barnacles (Sulaiman et al., 2020). The basic steps of the MBO is described in this section as follows.

3.2.1. Initialization

The candidate of solution is assumed as barnacles, and the population vector can be formulated as:

$$\mathbf{X} = \begin{bmatrix} x_1^1 & \cdots & x_1^N \\ \vdots & \ddots & \vdots \\ x_n^1 & \cdots & x_n^N \end{bmatrix} \tag{2}$$

here, N represents the number of control variables, where n indicates the number of barnacles or population. More so, in Eq. (2), the control variables are subject to the upper and lower bounds as:

$$\mathbf{ub} = \left[ub_1, \ldots, ub_i\right] \tag{3}$$

$$\mathbf{lb} = \left[lb_1, \ldots, lb_i\right] \tag{4}$$

In which ub is the upper bounds, and lb is the lower bounds. Moreover, the evaluation of the vector $\mathbf{X}$ evaluation process is implemented initially, in addition to, the sorting process is executed for locating the best solution at the top of $\mathbf{X}$ vector.

3.2.2. Selection process

In general, the BMO uses different strategies for the selection process. The process of barnacles selection depends on the penises length (pl). This is inspired by the barnacles behavior in nature, as follows:

1. Random selection: the process is executed randomly, but it is restricted to the pl.
2. Each barnacle contributes its sperm and receives a sperm from another barnacle. Here, each one is only fertilized by one barnacle at one time (Barazandeh, Davis, Neufeld, Coltman, & Palmer, 2013).
3. self-mating case: it is rarely happened in some cases the self-fertilization is happened (Yusa et al., 2012). Following (Sulaiman et al., 2020), this rare process is ignored.
4. Sperm cast process: this process is happened if the selection process during the iteration is more than the pl.

In summary, the exploitation phase of the BMO is done as described above (point 1 and point 2), whereas the exploration phase is done as described in point 4. Following Sulaiman et al. (2020), a simple selection is applied, as in the following equations:

$$X_d = randperm(N) \tag{5}$$

$$X_m = randperm(N) \tag{6}$$

The population number is indicated by N, where X_d and X_m represent the parents to be mated (i.e., Dad and Mum, respectively).

3.2.3. Reproduction

In this process, the new off-spring generation is produced inspired by the Hardy–Weinberg principle (Sulaiman et al., 2020), as in the following formulations:

$$X_i = p \times X_d + q \times X_m \ for\ k \leq pl \tag{7}$$

$$X_i = rand() \times X_m \ for\ k > pl \tag{8}$$

where p represents pseudo random numbers in the interval $[0,1]$, X_d represents the variables of Dad of barnacles, where X_m represents the variables of Mum. More so, $rand()$ indicates a random number in the interval $[0,1]$.

Furthermore, the exploitation and exploration phases of the BMO are represented by Eqs. (7) and (8), respectively.

3.3. Sine Cosine Algorithm (SCA)

In 2016, a novel optimizer using the concepts of sine and cosine fluctuations in math was developed (Mirjalili, 2016). This method is known as Sine Cosine Algorithm (SCA) that has found its place among other swarm-based methods since the proposal.

The mechanism in this optimizer is simple. It initially generates a set of random solutions to evolve them during the predefined number of iterations. The objective is to find the best position and its fitness based on updating the initial swarm and finding regions that include the global optimum. The SCA adjusts the sine and cosine functions to cover the area of search space. Using that fluctuation, the algorithm can visit various regions using its exploration mechanism and then switches to the exploitation phase. The main mathematical models of SCA are as follows:

$$X(t+1) = \begin{cases} X(t) + rand_1 \times sin(rand_2) \times |rand_3 X_b(t) - X(t)|, & rand_4 > 0.5 \\ X(t) + rand_1 \times cos(rand_2) \times |rand_3 X_b(t) - X(t)|, & rand_4 \leq 0.5 \end{cases} \tag{9}$$

where $X(t)$ is the position of the search agent, t shows the current iteration, $rand_1$, $rand_2$, and $rand_3$ are random values updated in each iteration, x_b is target position, and $rand_4$ is attained randomly in interval [0,1].

The SCA has a specific mechanism to balance the exploratory and exploitative trends during all searching phases, which is expressed as follows:

$$rand_1 = c - t \times \frac{c}{T} \tag{10}$$

where c denotes a constant and T is the maximum iteration.

3.4. Disruption Operator

In this section, we explain an operator called Disruption Operator (D_{op}), which is based on the physical processes in astrophysics. This process can be performed using rule in Eq. (11) (Liu, Ding, & Wang, 2014):

$$D_{op} = \begin{cases} dist_{i,j} \times \delta(\frac{-1}{2}, \frac{1}{2}) & if\ \ dist_{i,best} \geq 1 \\ 1 + dist_{i,best} \times \delta(\frac{-10^{-16}}{2}, \frac{10^{-16}}{2}) & otherwise \end{cases} \tag{11}$$

where $dist_{i,j}$ is the Euclidean distance between ith agent and its nearest neighborhood jth, $dist_{i,best}$ denotes the Euclidean distance between ith agent and the best position, and $\delta(a,b)$ is a random value inside domain $[a,b]$. This operator assists the BMSCD in highlighting the exploratory behaviors throughout the searching procedure. We employ disruption operators in the BMO to substitute performance momentarily by accumulated performance, which improves the excellence of the concluding solution. The disruption operator assists the methods in making more emphasis on the diversity of initial agents, and then, smoothly changeover from initial global search steps to the final exploitative trends. Also, the resulted method benefits from the exploitative advantages of both techniques.

4. Proposed approaches

In this section, we will discuss the steps of the developed method either for global optimization or for clustering problem. The details of the steps are given in the following sections.

4.1. Proposed approach for global optimization

In this section, an improved version of the BMO algorithm is presented (see Fig. 2). The proposed method depends on using the SCA and disruption operator to improve the ability of BMO to find the optimal solution for the given problem.

The searching process using the proposed method, called BMSCD, begins by determining the initial value for N agents (which represents the solutions). Each agent has a set of D elements/variables, which represents the dimension of the given problem. The next step is to determine the quality of each agent by calculating its fitness value and find the best agent. Then, we update the current agents using the operators of BMO, SCA and DO. This is performed by determining which phase from BMO will be applied to enhance the current solution. When the exploration phase is used, however, the probability of fitness value for the current agent is computed. Based on the probability value, we use either the operators of BMO in the exploration phase or the operator of SCA. After completing the updating process of all agents, the DO operation will be applied to improve the diversity of solutions. The previous steps are repeated until stopping conditions are met. The full description of the steps of the BMSCD is given in the following parts (see Fig. 1).

4.1.1. First phase: Initial population

The main objective of this phase is to generate a set of N agents X from a specified range of real numbers, and each of them has dimension D this can be formulated using the following equation.

$$X_{ij} = (LB_{ij} + rand \times (UB_{ij} - LB_{ij})), i = [1,2,\dots,N],\ j = [1,2,\dots,D] \quad (12)$$

In Eq. (12), UB_{ij} and LB_{ij} are the upper and lower boundaries for the solution i at the dimension j. The $rand$ is a function used to generate a random number from [0,1].

4.1.2. Second phase: Updating agents

This phase aims to improve the quality of agents by updating their position inside the search space. This process starts by evaluating the quality of the current agents X and calculating the fitness value (F_i) for each agent x_i. The agent that has the best fitness value (F_b) is chosen as the best agent X_b. Then each X_i is updated based on operators of BMO and SCA. This achieved first by determining whether the exploitation or exploration ability of BMO will be used, which is determined by computing the energy of the agents using Eqs. (7) and (8). When the exploitation strategies are used, a probability for each current solution is computed as:

$$Pr_i = \frac{F_i}{\sum_{i=1}^{N} F_i} \quad (13)$$

In the case $Pr_i < 0.5$, the operators of BMO during the exploitation phase are used (as in Eq. (7)), otherwise, the operators of SCA will be used. Thereafter, the agents will be improved their diversity by computing the disruption operator (D_{op}) using Eq. (11) then apply it to the current agents as the following:

$$X = D_{op} \times X \quad (14)$$

The previous steps will be repeated until it reached the maximum number of iterations (T). Then return the best solution X_b as the output of the proposed method. These steps are given in Algorithm 2.

Algorithm 1 Steps of BMSCD algorithm

Inputs: Total number of agent N, maximum number of iterations T, dimension of the given problem D, the lower (LB) and upper (UB) boundaries
Outputs: Best solution X_b
Using Eq. (12) from the first phase to initialize N agents ($x_i (i = 1,2,\dots,N)$).
$t = 0$.
while ($t \le T$) **do**
  Compute the fitness values F_i for each agent x_i.
  Find the best fitness value F_b and its agent X_b.
  for (each barnacles (X_i)) **do**
    Update the value of pl.
    Using Eqs. (5)–(6) for selecting parents.
    if Selection of Dad and Mum is pl **then**
      if $Pr_i \le 0.5$ **then**
        Using Eq. (7) to update X_i
      else
        Using Eq. (9) to update X_i
    else if Selection of Dad and Mum $> pl$ **then**
      Using Eq. (8) to update X_i
  Apply the D_{OP} to the updated agents X using Eq. (14).
  $t = t + 1$.
Return X_*

4.2. Proposed approach for automatic clustering

The clustering techniques have gained more attention recently since they can be applied to group the samples in different clusters. Therefore, they are applied to different applications, for example, pattern recognition, medical image segmentation, bioinformatics, machine learning, and text mining (Abualigah, Khader, & Hanandeh, 2018; Elaziz, Nabil, Ewees, & Lu, 2019; Kurada, Pavan, & Rao, 2015).

4.3. Clustering problem formulation

The definition of the automatic clustering problem is introduced in this subsection. It is defined as the operation of dividing the data (S), which contains N_S samples into different K_{max} groups. This can be performed by maximizing the between-cluster variation simultaneous with minimizing the within-cluster variation (Jain, Murty, & Flynn, 1999).

Therefore, the AC problem can be defined by considering the data contains N_S samples $S = [s_1, s_2, \dots, s_{N_S}]$. In addition, $s_i = [s_{i1}, s_{i2}, \dots, s_{iD}]$ where D is the total number of features. Then clustering method aims to split S into K_{max} groups (i.e., $C_1, C_2, \dots, C_{K_{max}}$) subject to (Das, Abraham, & Konar, 2007, 2008; Das & Konar, 2009; Das & Sil, 2010; Jain et al., 1999):

$$\cup_{l=1}^{K_{max}} C_l = S,\ C_l \neq \phi,\ l = 1,\dots,K_{max} \quad (15)$$

$$C_l \cap C_{l1} = \phi,\ l, l1 = 1,2,\dots,K_{max}, l \neq l1 \quad (16)$$

Further, the proposed BMSCD method is also used to automatic clustering datasets. The main aim of the BMSCD is to determine the optimal number of clustering centers and their values.

The BMSCD starts by determining the initial value for the BMO, SCA, and other parameters related to the automatic clustering problem.

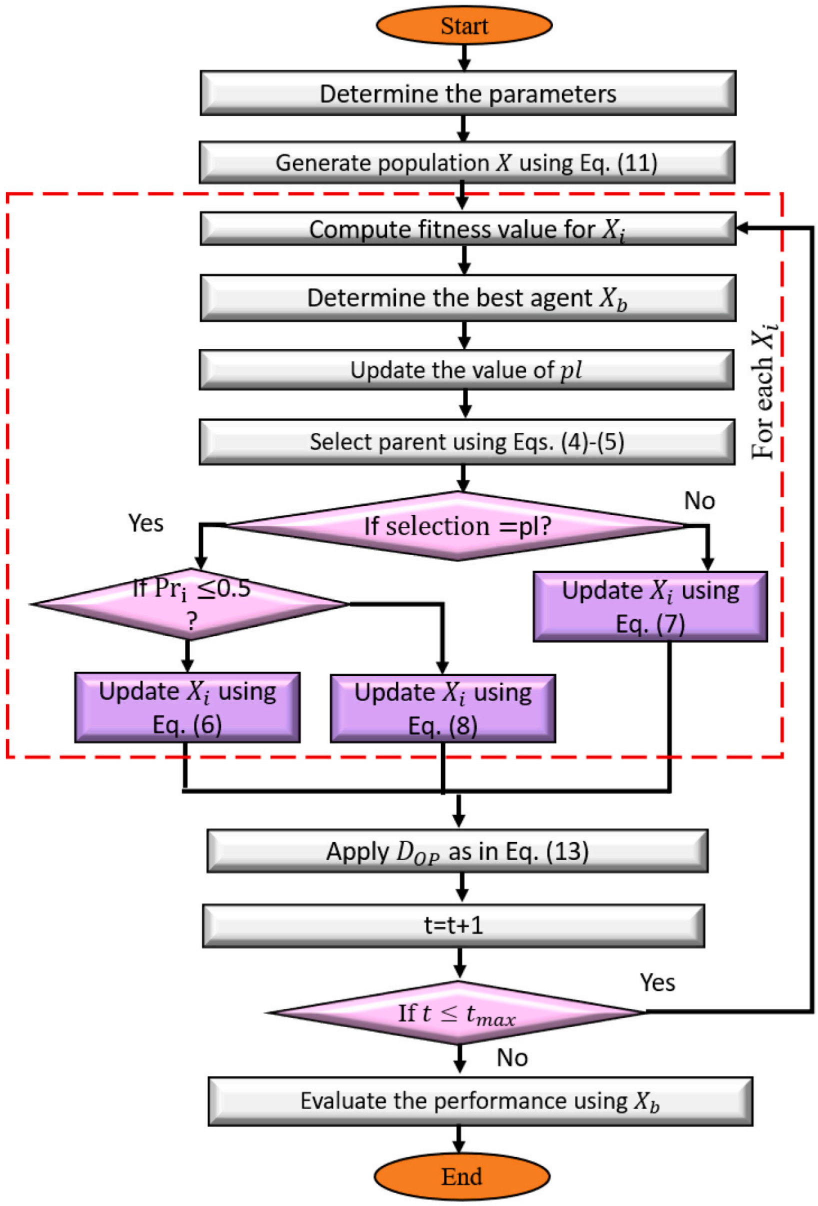

Fig. 1. The steps of the proposed BMSCD approach for global optimization.

After that, it generates a set of N solutions, which contains two parts; (1) threshold values and (2) the cluster centers. The threshold values are applied to determine if its corresponding center will passive or active during the process of determining the number of the optimal number of cluster centers. The next step is to compute the fitness value for each solution and find the best solution corresponding to the best fitness value. Then the solutions will be updated using the BMSCD as discussed in Section 4. This updating process is performed until reaching the stop conditions. The proposed BMSCD as a clustering method is given in the following.

4.3.1. Representation of solution

The cluster centroid-based encoding is applied to represent the solution for the clustering problem. For the data $S \in R^{N_S \times D}$ that required to split into K_{max}. The BMSCD generates a set of N solutions with dimension D_X, where $D_X = K_{max} + K_{max} \times D$. This dimension D_X contains two parts, the first part consist of K_{max} threshold values $th_{ij} \in [0,1], j = 1,2,\ldots,K_{max}$. Those values are used to determine whether the corresponding cluster center will active or not. Moreover, the second part consists of a set of K_{max} cluster centers and each has D dimensions. According to this formulation, $X_i \in X$ can represented as:

$$X_i = [th_{i1}, th_{i2}, \ldots, th_{iK_{max}}, C_{i1}, C_{i2}, \ldots, C_{iK_{max}}] \tag{17}$$

where C_{ij} represents the jth cluster center of the solution i. The threshold value TH_{ij} determine the activity of the center by using the following equation:

$$C_{ij} = \begin{cases} 1 & if\ TH_{ij} > \epsilon \\ 0 & if\ otherwise \end{cases} \quad , \epsilon \in [0,1] \tag{18}$$

To explain the main aim of using Eq. (18), assume $S \in R^2$ (i.e., 2D dataset) and $K_{max} = 4$, and the solution X_i is formulated as: $X_i = [0.9, 0.5, 0.2, 0.6, 15.8, 11.1, 3.5, 9.3, 2.4, 5.7, 12.3, 8.9]$ where 0.9, 0.5, 0.2 and 0.6 are thresholds which corresponding to the centers (15.8,11.1), (3.5,9.3), (2.4,5.7), and (12.3,8.9), respectively. If $\epsilon = 0.4$ then, according to Eq. (18), the first, second and fourth centers are used for partitioning the data. Followed Das and Konar (2009), the representation scheme for the solutions to clustering problem as given

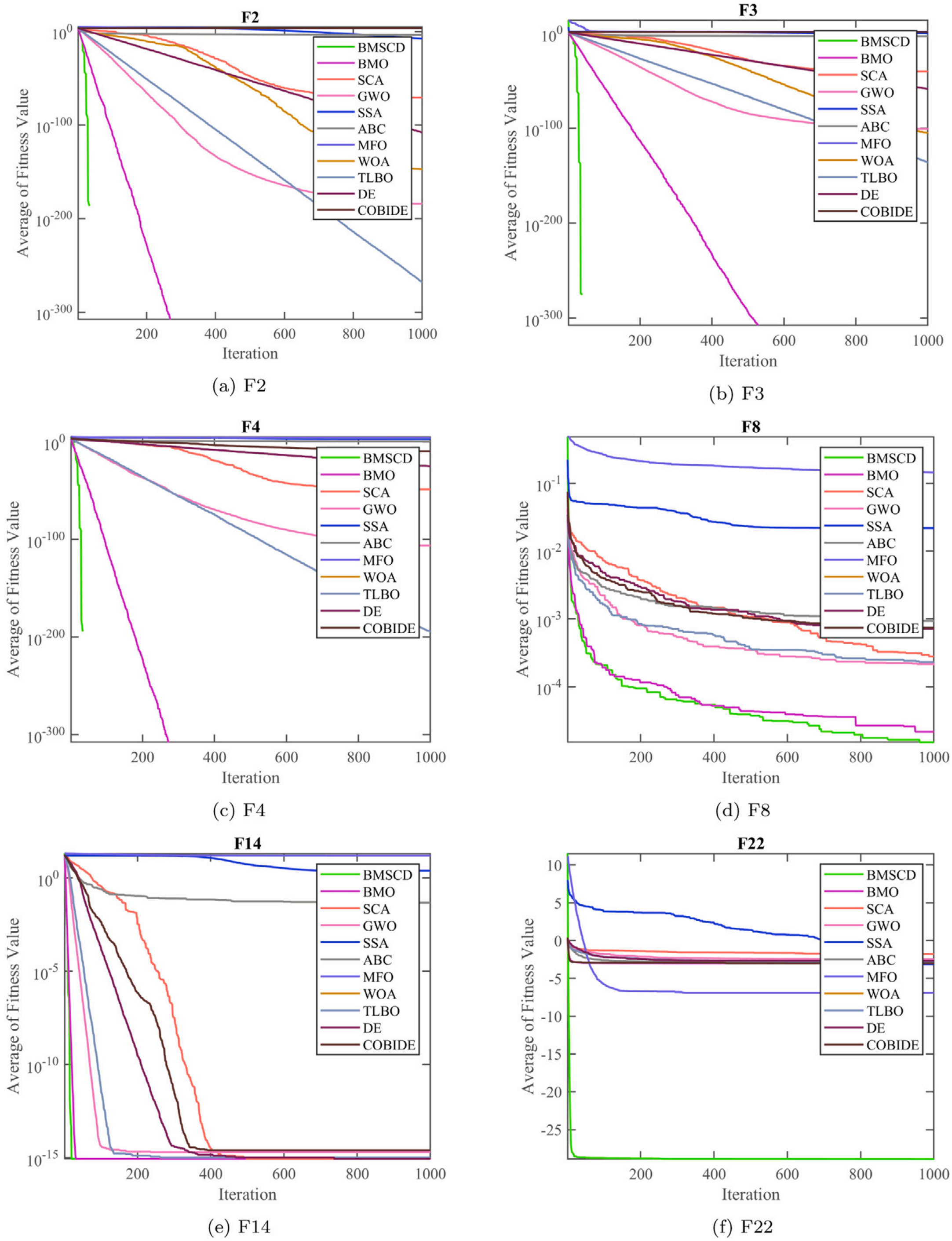

Fig. 2. Examples of the convergence curves for each method for the functions F2–F4, F8, F14, and F22.

in Eq. (17) established its quality. Since it can automatically determine the number of clustering at the same time with finding the center of each cluster.

4.3.2. Calculate fitness value

To assess the quality of $x_i \in X$, its fitness value is calculated. In this study, the CS-index (CSI) is applied to used as fitness value which defined as (Chou, Su, & Lai, 2004):

$$CSI = \frac{\frac{1}{K_{max}} \sum_{k=1}^{K_{max}} \frac{1}{|C_k|} \sum_{X_i \in C_K} \max_{X_j \in C_k} d(X_i, X_j)}{\frac{1}{K_{max}} \sum_{k=1}^{K_{max}} \min_{k,k \neq k'} d(C_k, C_{k'})} \tag{19}$$

where $|C_k|$ is the total number of samples in C_k. $k' = [1, 2, \dots, K_{max}]$ is index of cluster center subject to it not equal to k.

4.3.3. Update population

This stage started by determining X_b that has the smallest fitness value (i.e., CSI_b). The next step is to update the set of solutions using BMSCD. The steps of enhancing the solution are repeated until the stop conditions are reached then the X_b is returned. Then the performance criteria are applied to compute the quality of the output.

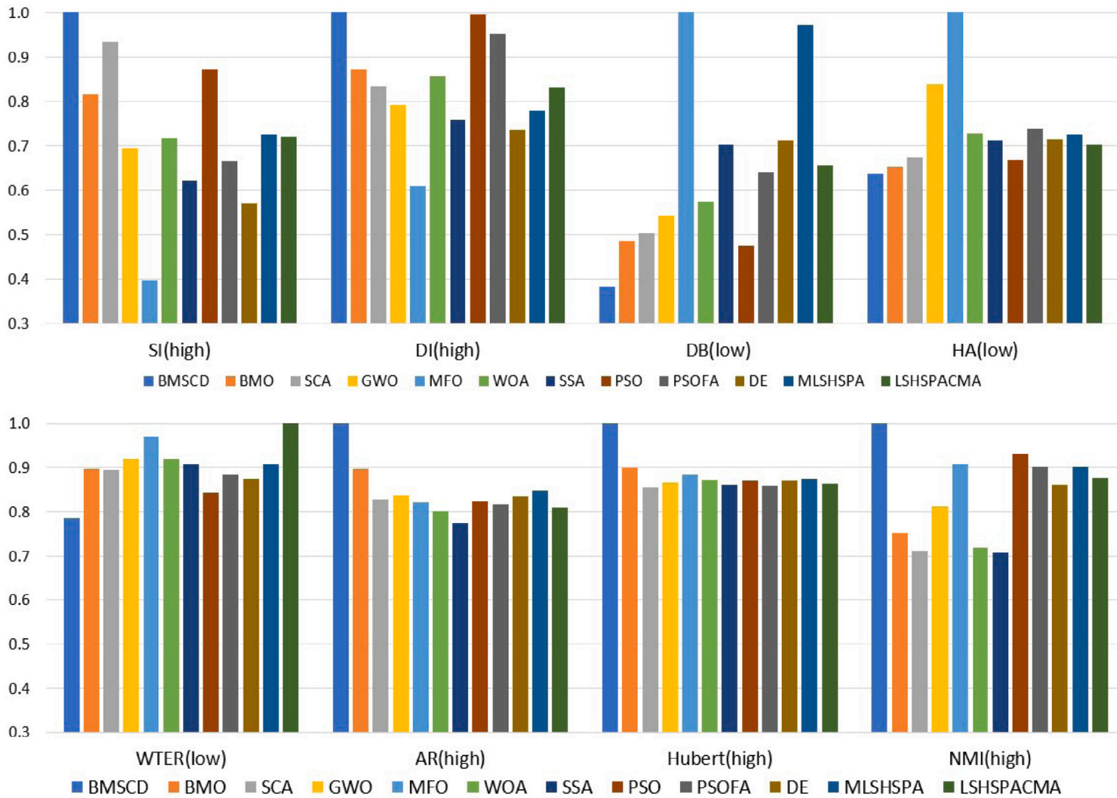

Fig. 3. Results of all algorithms using all measures based on the average of the fitness function values in all datasets.

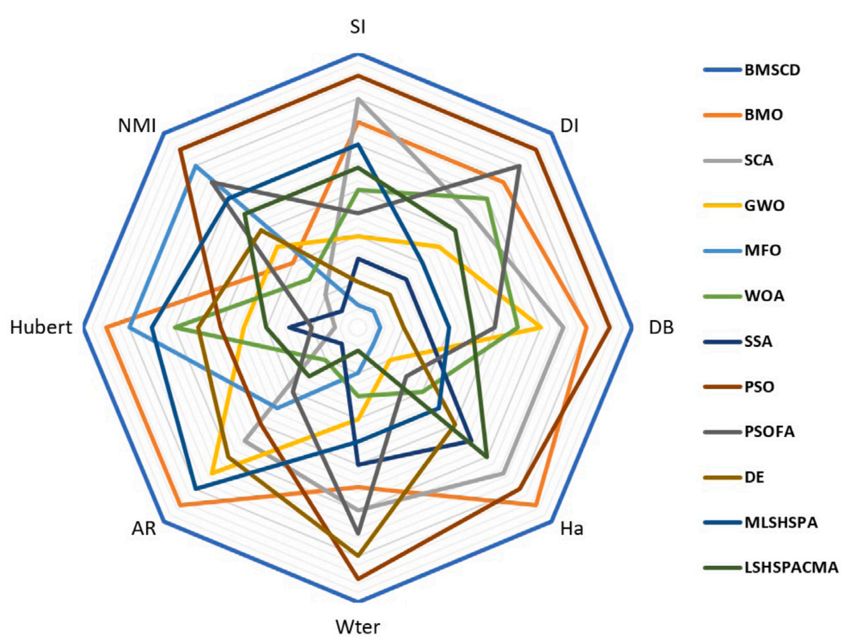

Fig. 4. Ranking of the algorithms in all measures.

5. Experimental series 1: Global optimization using BMSCD

5.1. Experimental setup

In this study, to have a fair comparison, we provided the same condition for all algorithms; they run on a 8 GB RAM Intel Core i5 processor over Matlab 2014b and Windows 10 64 bit.

5.2. Definition of well-known optimization benchmark functions

To investigate the performance of the BMSCD, a set of thirty-six global optimization problems from the well-known optimization benchmark (Suganthan et al., 2005) are used. These functions consist of three types of functions. The first type is unimodal, which contains the functions F1–F10. This type has a single extreme solution inside search domain. The second type is multimodal with variant dimension,

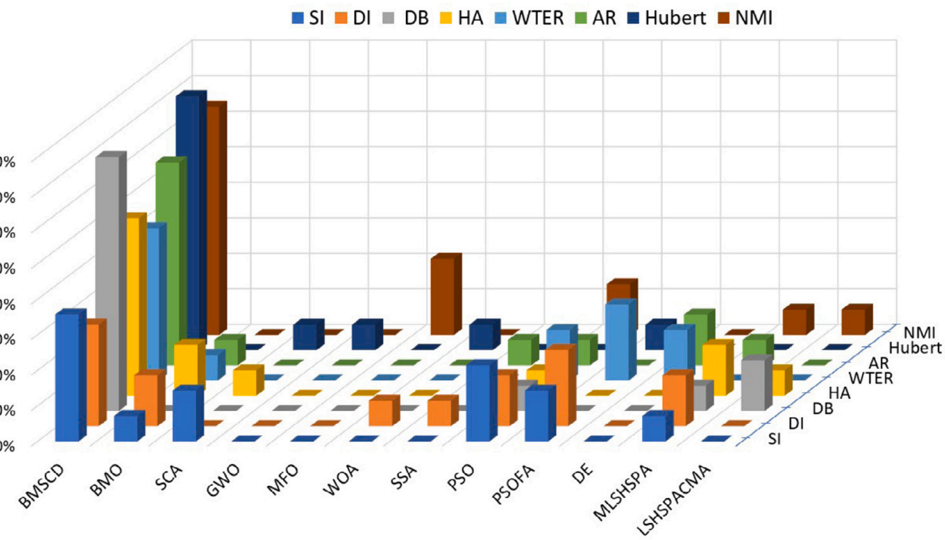

Fig. 5. Achievement ratios of each algorithm overall measures.

Table 1
Parameter settings for all methods.

Method	Parameters values
BMSCD	$pl = 7, p \in [0, 1],\ a = 2$
WOA	$a = [0, 2],\ \ b = 1,\ \ l = [-1, 1]$
BMO	$pl = 7, p \in [0, 1],$
SCA	$a = 2$
GWO	$a \in [20]$
MFO	$a \in [-2 - 1]$, Spiral factor b = 1
SSA	$C_2 \in [0,\ 1],\ \ C_3 \in [0,\ 1]$
DE	$pCR = 0.2,\ \ \beta_{min} = 0.2,\ \ \beta_{max} = 0.8$
TLBO	$T \in [1, 2]$
ABC	a = 1, employed bees = $N/2$, onlooker bees = $N/2$
COBIDE	pb = 0.4, ps = 0.5

Algorithm 2 Steps of automatic BMSCD clustering algorithm

Inputs: Total number of agent N, maximum number of iterations T, dimension of the given problem D, the lower (LB) and upper (UB) boundaries
Outputs: Best solution X_b
Using Eq. (17) from the first phase to initialize N agents $(x_i(i = 1, 2, \dots, N))$.
while (stopping condition is not met) **do**
  Calculate F_i for each agent x_i using Eq. (19).
  Determine the best fitness value F_b and its agent X_b.
  for (each barnacles (X_i)) **do**
    Update E_0 and jump strength J
    Update the value of pl.
    Using Eqs. (5)-(6) for selecting parents.
    if Selection of Dad and Mum is pl **then**
      if $Pr_i \leq 0.5$ **then**
        Using Eq. (7) to update X_i
      else
        Using Eq. (9) to update X_i
    else if Selection of Dad and Mum > pl **then**
      Using Eq. (8) to update X_i
  Apply the D_{OP} to the updated agents X using Eq. (14).
  $t = t + 1$.
Return X_*

and this is represented by F11–F26. This type assesses the ability of the method to find a global solution of the functions that have more than an extreme solution. The third type is called multimodal with fixed dimensions such as functions F27–F35. The description of the three types is given in Table A.1 (see Appendix).

5.3. Comparative methods

The results of the BMSCD is compared with other nine state-of-the-art global optimization methods in addition to the traditional BMO.

1. Whale optimization algorithm (WOA) (Mirjalili, Dong, & Lewis, 2019): It stimulates the ability of whales to catch the prey in nature.
2. Grey wolf optimizer (GWO) (Mirjalili, Aljarah, Mafarja, Heidari, & Faris, 2020): It is an algorithm that emulates the ability of grey wolves to catch the prey.
3. Sine–cosine algorithm (SCA) (Seyedali, 2016): It is global optimization that depends on the two triangle functions (i.e., sine and cosine) to solve the given problem.
4. Artificial bee colony (ABC) (Karaboga & Akay, 2009): It is a metaheuristic method which emulates the strategies of the bee colony to determine the solution for the given problem.
5. Salp swarm algorithm (SSA): This metaheuristic method emulates the strategy of the salpfish in nature (Faris, Mirjalili, Aljarah, Mafarja, & Heidari, 2020).
6. Moth flame optimization (MFO) (Mirjalili et al., 2019): It follows the strategy of the moths to find the moonlight.
7. Teaching–learning-based optimization (TLBO) (Rao, Savsani, & Vakharia, 2012): It simulates the relationship between the teachers and learners in the class.
8. Differential evolution (DE) (Sun, Yang, Yang, & Xu, 2019): It is an evolutionary algorithm that depends on the crossover, mutation, and selection process.
9. Differential evolution based on covariance matrix learning and bimodal distribution parameter setting (CoBiDE) (Wang, Li, Huang, & Li, 2014b): This algorithm improves the performance of the DE using the covariance matrix learning and the bimodal distribution parameter setting.

All results were computed over 30 independent runs. The population size was set to 30; the maximum number of iterations was set to 1000 for each algorithm. The parameters of each algorithm are set as their original implementations as in Table 1. All experiments were performed using “CPU Corei3” running with Windows 10 64 bit and 4 GB RAM using Matlab 2014b.

Table 2
Results of the average of the fitness value for all algorithms in all functions.

	BMSCD	BMA	SCA	GWO	SSA	ABC	MFO	WOA	TLBO	DE	COBIDE
F1	**0.00E+0**	1.50E−300	2.20E−71	6.50E−185	4.17E−11	2.83E−4	2.80E+3	4.72E−148	1.22E−268	1.10E−108	3.99E+3
F2	**0.00E+0**	1.44E−300	5.75E−41	1.74E−101	9.17E−7	9.16E−4	3.92E+1	6.31E−106	1.08E−136	1.93E−59	1.85E+1
F3	**0.00E+0**	2.10E−305	2.03E−49	5.83E−107	3.71E−11	2.74E+0	1.82E+4	3.33E−10	2.87E−195	1.31E−25	3.33E−10
F4	**0.00E+0**	0.00E+0	1.07E−30	1.00E−72	4.24E−6	3.33E−2	7.46E+1	2.88E−6	6.89E−119	2.22E−30	2.88E−6
F5	2.66E+1	7.35E−1	9.59E−1	1.12E+0	4.06E+1	1.36E+0	3.21E+6	1.39E+1	**3.61E−5**	5.92E−1	1.39E+1
F6	**0.00E+0**	4.61E−3	5.16E−3	1.00E−7	4.39E−11	1.93E−4	3.62E+3	1.12E−6	**0.00E+0**	**0.00E+0**	1.12E−6
F7	**1.53E−5**	2.17E−5	2.80E−4	2.08E−4	5.64E−4	9.28E−4	1.43E−1	7.37E−4	2.31E−4	7.13E−4	7.37E−4
F8	**0.00E+0**	**0.00E+0**	4.68E−72	5.88E−188	8.99E−13	2.25E−6	6.60E+2	2.35E−153	3.73E−269	3.92E−111	2.35E−153
F9	**0.00E+0**	2.66E−304	6.00E−132	**0.00E+0**	9.10E−29	1.44E−17	3.44E+0	5.43E−230	**0.00E+0**	3.30E−216	5.43E−230
F10	**0.00E+0**	1.63E−294	3.49E−64	3.70E−171	6.88E−5	2.13E−7	3.11E−9	6.42E−141	4.22E−226	4.92E−95	6.42E−141
F11	−1.63E+3	−2.02E+2	−2.18E+2	−1.54E+2	−2.18E+2	−2.18E+2	−1.36E+3	−2.18E+2	−2.09E+2	−2.13E+2	−2.18E+2
F12	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	2.27E+0	1.48E−2	1.75E+2	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F13	8.88E−16	8.88E−16	**8.88E−16**	2.03E−15	3.54E−6	4.80E−2	1.60E+1	2.59E−15	1.03E−15	**8.88E−16**	2.59E−15
F14	**0.00E+0**	**0.00E+0**	4.61E−3	1.34E−2	1.23E−1	3.84E−2	1.82E+1	7.45E−3	8.41E−3	1.14E−3	7.45E−3
F15	4.57E−4	1.18E−31	3.00E−3	1.99E−3	1.97E−12	7.76E−5	5.69E+1	2.61E−5	1.18E−31	**1.18E−31**	2.61E−5
F16	4.04E−1	9.31E−3	6.64E−3	2.81E−7	2.94E−12	1.61E−4	5.58E+5	4.53E−5	1.35E−32	**1.35E−32**	4.53E−5
F17	2.93E+0	1.04E−1	1.50E−1	1.40E−4	1.65E−4	8.38E−4	5.62E+1	7.57E−3	1.76E−2	**1.35E−31**	7.57E−3
F18	**0.00E+0**	**0.00E+0**	3.59E−16	1.56E−5	9.84E−3	4.74E−4	6.64E+0	3.09E−2	8.67E−31	4.06E−58	3.09E−2
F19	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	4.62E−14	1.53E−8	7.66E−1	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F20	**0.00E+0**	8.55E−263	7.72E−63	7.11E−146	3.61E−13	1.62E−2	3.57E+2	5.33E−47	3.86E−238	8.44E−49	5.33E−47
F21	−2.89E+1	−3.00E+0	**−1.78E+0**	−2.45E+0	−2.78E+0	−2.91E+0	−6.91E+0	−3.00E+0	−2.81E+0	−2.66E+0	−3.00E+0
F22	6.51E−1	1.35E−32	3.87E−2	9.20E−7	1.55E−13	9.15E−7	2.73E+1	1.54E−5	1.35E−32	**1.35E−32**	1.54E−5
F23	**0.00E+0**	**0.00E+0**	2.67E−65	4.41E−181	6.59E+3	1.33E+0	5.74E+7	3.45E−140	1.02E−263	5.40E−105	3.45E−140
F24	−4.77E−2	−6.00E−1	−7.27E−1	−1.00E+0	−9.20E−1	−3.32E−1	**0.00E+0**	−3.60E−1	−8.80E−1	−1.60E−1	−3.60E−1
F25	**0.00E+0**	1.93E−300	3.84E−2	8.39E−2	9.99E−2	1.88E−1	8.92E+0	7.99E−2	8.92E−2	9.22E−2	7.99E−2
F26	**0.00E+0**	7.72E−290	7.75E−3	9.72E−3	9.72E−3	2.32E−2	5.00E−1	9.26E−3	9.72E−3	9.55E−3	9.26E−3
F27	8.31E+0	7.94E+0	1.63E+0	5.03E+0	**9.98E−1**	**9.98E−1**	3.71E+0	1.79E+0	**9.98E−1**	1.59E+0	1.79E+0
F28	4.45E−4	3.45E−4	8.96E−4	4.45E−3	1.60E−3	9.17E−4	4.18E−3	8.40E−4	1.20E−3	7.88E−4	8.40E−4
F29	−1.03E+0	−1.03E+0	−1.03E+0	**−1.03E+0**	−1.03E+0	−1.03E+0	−1.03E+0	−1.03E+0	−1.03E+0	−1.03E+0	−1.03E+0
F30	3.98E−1	3.98E−1	**3.99E−1**	3.98E−1	3.98E−1	3.98E−1	3.98E−1	3.98E−1	3.98E−1	3.98E−1	3.98E−1
F31	**3.00E+0**	4.35E+0	3.00E+0	6.24E+0	**3.00E+0**	3.00E+0	**3.00E+0**	3.00E+0	**3.00E+0**	**3.00E+0**	3.00E+0
F32	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**
F33	**−3.32E+0**	**−3.32E+0**	**−3.25E+0**	−3.24E+0	**−3.32E+0**	**−3.32E+0**	−3.31E+0	−3.20E+0	**−3.32E+0**	**−3.32E+0**	−3.20E+0
F34	−5.06E+0	−5.06E+0	−1.94E+0	−9.54E+0	−7.65E+0	−1.01E+1	−5.83E+0	−8.05E+0	**−1.02E+1**	−9.64E+0	−8.05E+0
F35	−5.62E+0	−5.09E+0	−3.21E+0	**−1.04E+1**	−9.46E+0	**−1.04E+1**	−6.27E+0	−7.57E+0	−9.27E+0	**−1.04E+1**	−7.57E+0
F36	−5.40E+0	−5.13E+0	−3.81E+0	−1.03E+1	−8.10E+0	**−1.05E+1**	−7.75E+0	−7.96E+0	−8.99E+0	**−1.05E+1**	−7.96E+0

Table 3
Result of the STD of the fitness value.

	BMSCD	BMA	SCA	GWO	SSA	ABC	MFO	WOA	TLBO	DE	COBIDE
F1	**0.00E+0**	3.11E−140	1.05E−70	**0.00E+0**	2.84E−11	2.40E−4	4.58E+3	2.27E−147	**0.00E+0**	3.02E−108	1.88E+3
F2	**0.00E+0**	8.57E−130	2.62E−40	7.45E−101	4.13E−7	7.21E−4	2.10E+1	2.61E−105	3.92E−136	2.66E−59	1.17E+1
F3	**0.00E+0**	5.00E−210	9.70E−49	2.66E−106	2.72E−11	1.74E+0	1.31E+4	1.48E−9	**0.00E+0**	4.24E−25	1.48E−9
F4	**0.00E+0**	1.75E−100	3.75E−30	2.56E−72	1.79E−6	2.43E−2	7.19E+0	1.42E−5	2.45E−118	2.83E−30	1.42E−5
F5	5.44E−1	9.37E−2	2.94E−1	2.03E+0	1.17E+2	1.04E+0	1.60E+7	6.56E+1	1.67E−4	1.62E+0	6.56E+1
F6	**0.00E+0**	3.07E−3	3.54E−3	7.50E−8	3.80E−11	1.55E−4	7.04E+3	1.08E−6	**0.00E+0**	**0.00E+0**	1.08E−6
F7	**1.52E−5**	1.77E−5	2.82E−4	1.37E−4	6.19E−4	4.55E−4	6.56E−2	8.28E−4	1.89E−4	3.80E−4	8.28E−4
F8	**0.00E+0**	**0.00E+0**	2.31E−71	**0.00E+0**	6.32E−13	2.13E−6	8.42E+2	9.08E−153	**0.00E+0**	1.63E−110	9.08E−153
F9	**0.00E+0**	**0.00E+0**	2.97E−131	**0.00E+0**	1.94E−28	1.83E−17	1.07E+1	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F10	**0.00E+0**	**0.00E+0**	9.75E−64	**0.00E+0**	9.54E−5	2.30E−7	1.00E−8	2.28E−140	**0.00E+0**	2.28E−94	2.28E−140
F11	4.67E−13	1.84E+1	8.70E−14	2.58E+1	8.70E−14	8.70E−14	1.16E+2	8.70E−14	1.86E+1	1.13E+1	8.70E−14

(*continued on next page*)

Table 3 (*continued*).

	BMSCD	BMA	SCA	GWO	SSA	ABC	MFO	WOA	TLBO	DE	COBIDE
F12	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	1.56E+0	1.59E−2	3.99E+1	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F13	**0.00E+0**	**0.00E+0**	0.00E+0	1.69E−15	1.32E−6	2.23E−2	5.58E+0	1.81E−15	7.11E−16	**0.00E+0**	1.81E−15
F14	**0.00E+0**	**0.00E+0**	1.53E−2	1.02E−2	9.12E−2	1.46E−2	3.68E+1	2.45E−2	6.20E−3	2.76E−3	2.45E−2
F15	3.15E−4	2.25E−47	2.18E−3	9.93E−3	1.81E−12	9.24E−5	2.67E+2	2.98E−5	1.98E−33	**4.47E−47**	2.98E−5
F16	8.87E−1	2.10E−2	3.83E−3	2.28E−7	2.17E−12	1.27E−4	2.79E+6	4.87E−5	5.59E−48	**5.59E−48**	4.87E−5
F17	6.54E+0	2.43E−1	7.46E−2	8.07E−5	2.01E−4	6.60E−4	8.79E+1	2.18E−2	4.11E−2	**2.23E−47**	2.18E−2
F18	**0.00E+0**	**0.00E+0**	1.79E−15	3.16E−5	1.88E−2	2.37E−4	6.64E+0	6.90E−2	3.58E−30	1.23E−57	6.90E−2
F19	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	4.04E−14	1.83E−8	5.54E−1	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F20	**0.00E+0**	**0.00E+0**	3.55E−62	3.51E−145	2.20E−13	1.28E−2	1.12E+2	2.28E−46	**0.00E+0**	3.10E−48	2.28E−46
F21	3.57E−1	6.74E−3	3.78E−1	7.21E−1	4.85E−1	3.69E−2	2.07E+0	1.17E−2	4.00E−1	5.22E−1	1.17E−2
F22	1.70E+0	2.81E−48	2.29E−2	4.88E−7	1.07E−13	9.67E−7	2.79E+1	1.92E−5	**5.59E−48**	**5.59E−48**	1.92E−5
F23	**0.00E+0**	**0.00E+0**	1.33E−64	**0.00E+0**	1.14E+4	1.65E+0	6.31E+7	1.20E−139	**0.00E+0**	2.09E−104	1.20E−139
F24	2.13E−1	5.03E−1	1.87E−1	2.41E−6	2.77E−1	4.67E−1	**0.00E+0**	4.89E−1	3.32E−1	3.35E−1	4.89E−1
F25	**0.00E+0**	**0.00E+0**	4.80E−2	3.74E−2	1.85E−14	7.18E−2	3.69E+0	4.99E−2	2.67E−2	2.44E−2	4.99E−2
F26	**0.00E+0**	**0.00E+0**	3.96E−3	5.90E−10	7.29E−16	1.30E−2	2.96E−4	6.86E−3	4.98E−10	8.36E−4	6.86E−3
F27	5.14E+0	4.48E+0	9.43E−1	4.62E+0	2.32E−16	1.16E−10	3.70E+0	2.00E+0	**4.53E−17**	1.40E+0	2.00E+0
F28	1.60E−4	9.17E−5	3.34E−4	8.13E−3	3.92E−3	1.95E−4	7.23E−3	5.69E−4	4.00E−3	2.32E−4	5.69E−4
F29	3.09E−11	2.28E−16	3.00E−5	1.16E−8	9.88E−15	9.03E−10	**6.80E−16**	6.78E−10	**6.80E−16**	**6.80E−16**	6.78E−10
F30	**0.00E+0**	**0.00E+0**	6.92E−4	2.26E−7	3.74E−15	8.16E−9	**0.00E+0**	2.82E−6	**0.00E+0**	**0.00E+0**	2.82E−6
F31	6.04E+0	4.43E−15	5.68E−5	1.62E+1	1.26E−13	6.65E−4	2.57E−15	1.52E−4	5.73E−16	**4.80E−16**	1.52E−4
F32	2.88E−5	6.32E−5	1.26E−3	2.78E−5	2.37E−9	1.98E−8	2.21E−3	2.53E−4	**2.27E−15**	**2.27E−15**	2.53E−4
F33	2.63E−12	1.86E−2	2.59E−2	2.71E−1	1.45E−13	1.86E−9	3.74E−2	3.42E−1	**6.47E−16**	**4.53E−16**	3.42E−1
F34	2.88E−16	6.11E−16	2.02E+0	1.68E+0	3.46E+0	2.88E−2	3.19E+0	2.98E+0	4.12E−3	1.77E+0	2.98E+0
F35	1.64E+0	1.82E−15	2.13E+0	**4.32E−4**	2.24E+0	1.97E−2	3.54E+0	2.80E+0	2.31E+0	9.82E−2	2.80E+0
F36	1.21E+0	2.67E−15	1.94E+0	1.08E+0	3.65E+0	3.17E−2	3.53E+0	3.29E+0	2.82E+0	1.13E−1	3.29E+0

Table 4
Results of the best fitness value obtained by each method.

	BMSCD	BMO	SCA	GWO	SSA	ABC	MFO	WOA	TLBO	DE	COBIDE
F1	**0.00E+0**	1.60E−307	1.78E−84	3.97E−198	3.07E−12	8.61E−6	1.25E−4	1.46E−178	4.98E−277	2.68E−115	1.16E+3
F2	**0.00E+0**	1.42E−315	4.88E−48	1.36E−108	2.52E−7	2.46E−4	3.02E−3	9.08E−123	1.35E−141	1.29E−61	2.93E+0
F3	**0.00E+0**	2.09E−310	1.07E−66	2.20E−129	3.31E−12	4.04E−1	9.17E+2	7.33E−29	3.98E−203	3.02E−30	7.33E−29
F4	**0.00E+0**	**0.00E+0**	5.82E−37	1.38E−79	9.44E−7	3.33E−3	5.25E+1	1.18E−17	8.61E−123	7.67E−32	1.18E−17
F5	2.61E+1	4.61E−1	2.96E−1	3.82E−2	4.84E−2	1.91E−1	5.35E+1	1.12E−4	**2.64E−10**	1.58E−3	1.12E−4
F6	**0.00E+0**	1.68E−3	1.12E−3	9.78E−9	3.19E−12	2.03E−5	1.27E−4	3.52E−8	**0.00E+0**	**0.00E+0**	3.52E−8
F7	**3.86E−8**	2.87E−7	2.82E−6	9.32E−6	7.67E−5	5.96E−5	5.72E−2	2.22E−5	1.79E−5	1.76E−4	2.22E−5
F8	**0.00E+0**	**0.00E+0**	1.51E−84	1.01E−202	4.06E−14	5.54E−8	1.47E−4	2.67E−177	1.41E−278	3.83E−116	2.67E−177
F9	**0.00E+0**	**0.00E+0**	5.58E−168	**0.00E+0**	1.05E−31	9.61E−20	1.99E−10	2.77E−269	**0.00E+0**	6.45E−229	2.77E−269
F10	**0.00E+0**	**0.00E+0**	1.37E−78	1.55E−185	2.27E−6	7.04E−9	1.19E−17	1.23E−163	1.30E−233	4.52E−101	1.23E−163
F11	**−1.63E+3**	−2.18E+2	−2.18E+2	−1.87E+2	−2.18E+2	−2.18E+2	−1.63E+3	−2.18E+2	−2.18E+2	−2.18E+2	−2.18E+2
F12	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	2.34E−12	4.36E−4	1.13E+2	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F13	**8.88E−16**	**8.88E−16**	**8.88E−16**	**8.88E−16**	1.30E−6	9.94E−3	1.65E+0	**8.88E−16**	**8.88E−16**	**8.88E−16**	**8.88E−16**
F14	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	7.40E−3	1.05E−2	2.51E−4	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F15	1.51E−4	1.18E−31	6.98E−4	1.19E−8	1.18E−13	2.31E−6	1.15E−4	1.99E−7	**1.18E−31**	**1.18E−31**	1.99E−7
F16	6.19E−3	1.35E−32	3.86E−4	9.08E−9	2.82E−13	2.27E−5	1.20E−2	2.92E−6	**1.35E−32**	**1.35E−32**	2.92E−6
F17	3.05E−2	1.35E−31	2.85E−2	8.20E−6	4.42E−6	6.53E−5	6.83E−5	5.02E−5	**1.35E−31**	**1.35E−31**	5.02E−5
F18	**0.00E+0**	**0.00E+0**	3.99E−46	6.35E−106	3.07E−8	4.36E−5	9.42E−5	1.22E−116	1.90E−216	**0.00E+0**	1.22E−116
F19	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	2.39E−15	8.18E−10	1.48E−1	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F20	**0.00E+0**	**0.00E+0**	9.34E−74	1.03E−159	1.80E−14	2.80E−3	1.39E+2	1.52E−67	1.61E−245	1.17E−56	1.52E−67
F21	**−3.00E+0**	**−3.00E+0**	−2.95E+0	**−3.00E+0**	−3.00E+0	−2.98E+0	−1.07E+1	−3.00E+0	−3.00E+0	−3.00E+0	−3.00E+0
F22	**1.35E−36**	8.22E−3	8.03E−3	1.86E−7	2.42E−14	8.26E−9	5.07E−6	2.75E−7	1.35E−32	1.35E−32	2.75E−7
F23	**0.00E+0**	**0.00E+0**	1.09E−83	1.56E−194	2.95E+2	2.32E−2	2.49E+6	2.14E−164	4.43E−271	1.30E−111	2.14E−164

(*continued on next page*)

Table 4 (*continued*).

	BMSCD	BMO	SCA	GWO	SSA	ABC	MFO	WOA	TLBO	DE	COBIDE
F24	−9.54E−1	−1.00E+0	−9.53E−1	−1.00E+0	−1.00E+0	−1.00E+0	**0.00E+0**	−1.00E+0	−1.00E+0	−1.00E+0	−1.00E+0
F25	**0.00E+0**	**0.00E+0**	8.71E−36	2.21E−94	9.99E−2	9.99E−2	3.90E+0	2.55E−78	9.71E−3	1.04E−2	2.55E−78
F26	**0.00E+0**	**0.00E+0**	**0.00E+0**	9.72E−3	9.72E−3	9.78E−3	4.99E−1	**0.00E+0**	9.72E−3	5.54E−3	0.00E+0
F27	9.98E−1	9.98E−1	**9.98E−1**	**9.98E−1**	**9.98E−1**	**9.98E−1**	**9.98E−1**	**9.98E−1**	**9.98E−1**	**9.98E−1**	**9.98E−1**
F28	3.08E−4	3.08E−4	3.27E−4	3.07E−4	5.75E−4	6.08E−4	7.35E−4	3.08E−4	**3.07E−4**	4.33E−4	3.08E−4
F29	**−1.03E+0**	**−1.03E+0**	**−1.03E+0**	**−1.03E+0**	**−1.03E+0**	**−1.03E+0**	**−1.03E+0**	**−1.03E+0**	**−1.03E+0**	**−1.03E+0**	**−1.03E+0**
F30	3.98E−1	3.98E−1	**3.98E−1**	3.98E−1	3.98E−1	3.98E−1	3.98E−1	3.98E−1	3.98E−1	3.98E−1	3.98E−1
F31	**3.00E+0**	**3.00E+0**	**3.00E+0**	**3.00E+0**	**3.00E+0**	**3.00E+0**	**3.00E+0**	**3.00E+0**	**3.00E+0**	**3.00E+0**	**3.00E+0**
F32	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**
F33	**−3.32E+0**	**−3.32E+0**	−3.30E+0	**−3.32E+0**	**−3.32E+0**	**−3.32E+0**	**−3.32E+0**	**−3.32E+0**	**−3.32E+0**	**−3.32E+0**	**−3.32E+0**
F34	−5.06E+0	−5.06E+0	−8.00E+0	−1.02E+1	−1.02E+1	−1.02E+1	**−1.02E+1**	−1.02E+1	**−1.02E+1**	**−1.02E+1**	−1.02E+1
F35	**−1.04E+1**	−5.09E+0	−7.79E+0	−1.04E+1	−1.04E+1	−1.04E+1	−1.04E+1	**−1.04E+1**	−1.04E+1	−1.04E+1	**−1.04E+1**
F36	**−1.05E+1**	−5.13E+0	−7.22E+0	−1.05E+1	−1.05E+1	−1.05E+1	−1.05E+1	−1.05E+1	−1.05E+1	−1.05E+1	−1.05E+1

Table 5
Results of the worst fitness value obtained by each method.

	BMSCD	BMA	SCA	GWO	SSA	ABC	MFO	WOA	TLBO	DE	COBIDE
F1	**0.00E+0**	8.50E−290	5.26E−70	1.54E−183	1.19E−10	7.82E−4	1.00E+4	1.13E−146	1.37E−267	1.32E−107	7.96E+3
F2	**0.00E+0**	5.44E−280	1.32E−39	3.73E−100	1.75E−6	3.03E−3	7.00E+1	1.30E−104	1.96E−135	1.02E−58	5.10E+1
F3	**0.00E+0**	1.10E−300	4.86E−48	1.33E−105	1.05E−10	6.21E+0	4.63E+4	7.41E−9	7.14E−194	1.97E−24	7.41E−9
F4	**0.00E+0**	0.00E+0	1.77E−29	1.11E−71	8.38E−6	8.37E−2	8.58E+1	7.08E−5	1.23E−117	8.88E−30	7.08E−5
F5	28.65957883	0.860883532	1.32E+0	8.55E+0	4.43E+2	4.49E+0	7.99E+7	3.29E+2	**8.39E−4**	7.66E+0	3.29E+2
F6	0.013845528	9.46E−3	1.70E−2	3.66E−7	1.58E−10	5.75E−4	2.02E+4	4.35E−6	**0.00E+0**	**0.00E+0**	4.35E−6
F7	4.99528E−5	**5.65E−5**	1.32E−3	5.22E−4	3.03E−3	1.88E−3	2.89E−1	2.44E−3	8.55E−4	1.45E−3	2.44E−3
F8	**0.00E+0**	**0.00E+0**	1.15E−70	1.43E−186	2.07E−12	7.60E−6	3.60E+3	4.51E−152	4.57E−268	8.19E−110	4.51E−152
F9	**0.00E+0**	2.66E−297	1.49E−130	**0.00E+0**	9.72E−28	7.72E−17	5.37E+1	1.36E−228	**0.00E+0**	8.23E−215	1.36E−228
F10	**0.00E+0**	8.63E−287	4.32E−63	9.24E−170	4.54E−4	8.87E−7	4.26E−8	1.06E−139	1.05E−224	1.14E−93	1.06E−139
F11	**−1.63E+3**	−156.970143	−2.18E+2	−7.31E+1	−2.18E+2	−2.18E+2	−1.10E+3	−2.18E+2	−1.57E+2	−1.87E+2	−2.18E+2
F12	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	5.97E+0	6.29E−2	2.65E+2	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F13	8.88178E−16	8.88E−16	**8.88E−16**	4.44E−15	6.00E−6	9.39E−2	2.00E+1	4.44E−15	4.44E−15	**8.88E−16**	4.44E−15
F14	**0.00E+0**	**0.00E+0**	7.48E−2	3.86E−2	3.20E−1	6.59E−2	9.06E+1	1.21E−1	2.32E−2	9.86E−3	1.21E−1
F15	0.001300036	1.17791E−31	1.12E−2	4.96E−2	8.05E−12	4.40E−4	1.34E+3	9.44E−5	**1.27E−31**	**1.18E−31**	9.44E−5
F16	2.96609924	0.087261225	1.42E−2	8.88E−7	7.83E−12	4.88E−4	1.40E+7	1.71E−4	**1.35E−32**	**1.35E−32**	1.71E−4
F17	29.66737059	0.988826486	3.13E−1	3.01E−4	8.11E−4	2.69E−3	3.15E+2	1.11E−1	1.10E−1	**1.35E−31**	1.11E−1
F18	**0.00E+0**	**0.00E+0**	8.97E−15	1.13E−4	8.43E−2	1.32E−3	2.18E+1	2.32E−1	1.76E−29	4.78E−57	2.32E−1
F19	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	1.60E−13	8.35E−8	2.09E+0	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F20	**0.00E+0**	1.55E−256	1.77E−61	1.76E−144	9.23E−13	5.77E−2	5.54E+2	1.14E−45	8.60E−237	1.50E−47	1.14E−45
F21	−27.4180749	−2.97268307	**−1.43E+0**	−1.48E+0	−1.49E+0	−2.83E+0	−3.86E+0	−2.95E+0	−1.50E+0	−1.50E+0	−2.95E+0
F22	7.620915971	1.35E−32	9.51E−2	1.98E−6	4.22E−13	3.56E−6	8.76E+1	7.21E−5	**1.35E−32**	**1.35E−32**	7.21E−5
F23	**0.00E+0**	**0.00E+0**	6.67E−64	9.47E−180	5.88E+4	6.30E+0	2.38E+8	5.44E−139	1.53E−262	1.01E−103	5.44E−139
F24	−3.8299E−22	−6.5776E−9	−4.48E−7	−1.00E+0	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	−8.11E−5	**0.00E+0**	**0.00E+0**
F25	**0.00E+0**	1.93E−280	9.99E−2	9.99E−2	9.99E−2	3.03E−1	1.66E+1	2.00E−1	9.99E−2	9.99E−2	2.00E−1
F26	**0.00E+0**	7.72E−270	9.72E−3	9.72E−3	9.72E−3	3.73E−2	5.00E−1	3.72E−2	9.72E−3	9.72E−3	3.72E−2
F27	12.67050581	12.67050581	2.98E+0	1.27E+1	**9.98E−1**	9.98E−1	1.46E+1	1.08E+1	**9.98E−1**	5.93E+0	1.08E+1
F28	0.000771198	0.000712931	1.52E−3	2.04E−2	2.04E−2	1.49E−3	2.04E−2	2.25E−3	2.04E−2	1.28E−3	2.25E−3
F29	−1.03162845	−1.03162845	−1.03E+0	−1.03E+0	−1.03E+0	−1.03E+0	−1.03E+0	−1.03E+0	−1.03E+0	−1.03E+0	−1.03E+0
F30	**3.98E−1**	3.98E−1	4.01E−1	**3.98E−1**	**3.98E−1**	**3.98E−1**	**3.98E−1**	**3.98E−1**	**3.98E−1**	**3.98E−1**	**3.98E−1**
F31	**3.00E+0**	**3.00E+0**	3.00E+0	8.40E+1	**3.00E+0**	3.00E+0	**3.00E+0**	3.00E+0	**3.00E+0**	**3.00E+0**	3.00E+0
F32	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**	**−3.86E+0**
F33	**−3.32E+0**	−3.24219543	−3.19E+0	−2.34E+0	**−3.32E+0**	**−3.32E+0**	−3.19E+0	−2.25E+0	**−3.32E+0**	**−3.32E+0**	−2.25E+0
F34	−5.05519773	−5.05519773	−3.51E−1	−5.06E+0	−2.63E+0	−1.00E+1	−2.63E+0	−8.81E−1	**−1.01E+1**	−2.68E+0	−8.81E−1
F35	−5.08767183	−5.08767183	−5.22E−1	**−1.04E+1**	−2.77E+0	−1.03E+1	−1.84E+0	−3.72E+0	−4.15E+0	−9.91E+0	−3.72E+0
F36	−5.12848079	−5.12848079	−9.44E−1	−5.13E+0	−2.42E+0	**−1.04E+1**	−2.43E+0	−1.68E+0	−3.84E+0	−9.97E+0	−1.68E+0

Table 6
Sensitivity analysis of the dimension size on the BMSCD.

	Average				STD			
	Dim=30	Dim=15	Dim=50	Dim=500	Dim=30	Dim=15	Dim=50	Dim=500
F1	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F2	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F3	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F4	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F5	2.66E+1	**1.09E+1**	4.65E+1	4.95E+2	5.44E−1	**1.51E−1**	4.05E−1	5.44E−1
F6	**0.00E+0**	4.31E−6	2.51E−1	2.99E+1	**0.00E+0**	6.84E−6	2.36E−1	1.38E+1
F7	**1.53E−5**	1.91E−5	2.26E−5	1.62E−5	**1.52E−5**	2.19E−5	2.38E−5	1.35E−5
F8	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F9	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F10	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F11	**−1.63E+3**	−8.16E+2	−2.72E+3	−2.72E+4	4.67E−13	**0.00E+0**	4.67E−13	1.49E−11
F12	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F13	**8.88E−16**	**8.88E−16**	**8.88E−16**	**8.88E−16**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F14	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F15	4.57E−4	**9.19E−7**	5.50E−3	9.11E−2	3.15E−4	**8.81E−7**	4.37E−3	3.61E−2
	Best				Worst			
	Dim=30	Dim=15	Dim=50	Dim=500	Dim=30	Dim=15	Dim=50	Dim=500
F1	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F2	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F3	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F4	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F5	2.61E+1	**1.06E+1**	4.61E+1	4.94E+2	2.87E+1	**1.12E+1**	4.74E+1	4.96E+2
F6	**1.68E−3**	1.15E−6	1.77E−2	1.73E+1	1.38E−2	3.28E−5	7.43E−1	6.81E+1
F7	**3.86E−8**	3.34E−6	9.56E−7	6.00E−7	5.00E−5	9.25E−5	9.57E−5	**4.83E−5**
F8	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F9	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F10	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F11	**−1.63E+3**	−8.16E+2	−2.72E+3	−2.72E+4	**−1.63E+3**	**−8.16E+2**	−2.72E+3	−2.72E+4
F12	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F13	**8.88E−16**	**8.88E−16**	**8.88E−16**	**8.88E−16**	**8.88E−16**	**8.88E−16**	**8.88E−16**	**8.88E−16**
F14	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F15	1.51E−4	**1.79E−7**	6.34E−4	3.09E−2	1.30E−3	**3.17E−6**	1.48E−2	1.86E−1

5.4. Performance criteria

In this experimental series, a set of performance metrics are used which include average ($Average_F$), worst value ($Worst_F$), best value ($Best_F$), and standard deviation (STD_F). All metrics are calculated using the fitness values. These criteria are computed overall the total number of runs.

5.5. Global optimization problems

Table 2 shows the average fitness results, while the STD is given in Table 3. According to the results in Table 2, it is observed that the BMSCD has achieved the best values in 9 cases. In comparison, there is a tie on 12 cases, and for 15 cases, the results are not better than other competitors. If we see the other methods, the DE outperforms other peers only in three cases. In comparison, it is not better than other approaches in 25 cases, and there are similar results for nine cases. For unimodal cases, the proposed BMSCD outperforms the peers on 80% of cases and obtain the optimum solutions. At the same time, we see the methods such as SCA, SSA, COBIDE, and ABC have not shown a competitive efficacy for most of the cases. The proposed BMSCD attains the best optimum results for F12, F13, F14, F18, F19, F20, F23, and F25. The results for multimodal problems expose the improved exploratory tendencies of the modified variant compared to the basic BMO. We see for fixed dimension cases F26–F36; the results are very competitive with other techniques, especially BMO. For F32 and F33, all methods have achieved the same value.

As per STD values in Table 3, the results for the modified variant are the best on F1–F4, F6–F10, F12–F14, F18–F20, F23, F25, F26, and F30. The satisfactory performance of the BMO is also observed for 13 cases. However, SSA, ABC, SCA, and MFO have not highly accurate results compared to the TLBO, DE, COBIDE, and WOA techniques.

As per the best results in Table 4, the BMSCD is better than and similar in seven cases. We see similar patterns such as average results, and methods such as SSA, ABC, MFO, and COBIDE cannot avoid local optima. Their best-touched regions are still not satisfying.

Similar patterns can be observed according to the worst results as in Table 5. There are nine cases that the BMSCD method outperforms other peers and ten cases that the results are the same. For BMO, SCA, GWO, SSA, ABC, MFO, WOA, TLBO, DE, and COBIDE methods, there are 1, 1, 2, 0, 1, 0, 0, 2, and 1 functions that they can attain superior results, respectively.

Table 7
Sensitivity analysis of the SCA parameter on the BMSCD.

	a = 2				a = 1.5			
	Average	STD	Best	Worst	Average	STD	Best	Worst
F1	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F2	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F3	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F4	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F5	2.66E+1	5.44E−1	2.61E+1	2.87E+1	**2.64E+1**	**2.74E−1**	**2.60E+1**	**2.70E+1**
F6	**0.00E+0**	**0.00E+0**	1.68E−3	**1.38E−2**	2.72E−2	7.24E−2	**1.34E−3**	2.51E−1
F7	**1.53E−5**	**1.52E−5**	**3.86E−8**	**5.00E−5**	1.95E−5	1.73E−5	6.37E−7	6.87E−5
F8	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F9	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F10	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F11	**−1.63E+3**	**4.67E−13**	**−1.63E+3**	**−1.63E+3**	**−1.63E+3**	**4.67E−13**	**−1.63E+3**	**−1.63E+3**
F12	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F13	**8.88E−16**	**0.00E+0**	**8.88E−16**	**8.88E−16**	**8.88E−16**	**0.00E+0**	**8.88E−16**	**8.88E−16**
F14	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**	**0.00E+0**
F15	**4.57E−4**	3.15E−4	1.51E−4	**1.30E−3**	6.08E−4	**1.46E−3**	**1.34E−4**	6.79E−3

Table 8
Results of the CEC2020 benchmark functions.

Func.	Measure	BMSCD	BMO	SCA	GWO	SSA	ABC	MFO	WOA	TLBO	DE	COBIDE
F1	average	109	1042	3.0E+8	1.5E+7	469	107	1672	1166	174	**102**	1.8E+10
	STD	672.40	849.77	9.3E+7	2.6E+7	479.67	**4.39**	783.89	849.79	71.67	4.68	589.00
F2	average	1842	1955	2024	1412	1678	1152	1953	2005	1150	**1127**	3588
	STD	295.92	288.91	164.62	211.53	258.10	43.71	286.89	334.21	67.35	**10.33**	261.01
F3	average	714	756	749	719	726	714	735	764	714	**712**	1102
	STD	2.46	20.04	6.98	4.61	6.21	0.93	9.96	17.81	1.89	0.45	63.19
F4	average	**1900**	**1900**	**1900**	**1900**	1901	**1900**	1901	**1900**	**1900**	**1900**	46251
	STD	**0.00**	**0.00**	0.45	0.36	0.37	0.00	0.41	0.11	0.09	0.05	29760.48
F5	average	**1820**	10704	57004	53207	2659	2462	66548	4242	2063	1886	1.2E+7
	STD	**97.61**	8260.09	1.1E+5	1.1E+5	496.60	353.38	1.2E+5	2331.06	206.24	154.63	7.0E+6
F6	average	**1550**	1776	1617	1664	1645	1601	1694	1677	1603	1601	2399
	STD	**0.18**	20.44	57.25	70.77	58.25	0.42	88.63	76.67	1.57	9.31	168.59
F7	average	2303	4544	3587	2730	2442	2651	18854	3320	**2238**	2384	3.0E+6
	STD	152.92	580.44	637.64	302.05	203.51	197.88	60462.14	833.52	**106.73**	159.99	2.7E+6
F8	average	**2180**	2300	2610	2372	2450	2207	3104	2876	2327	2246	2369
	STD	**0.00**	**0.00**	46.00	43.25	109.83	3.03	410.13	404.50	36.57	51.95	3.99
F9	average	**2600**	2689	2827	2795	2609	2469	2813	2730	2704	2748	3268
	STD	**0.00**	**0.00**	5.25	53.15	60.29	25.61	3.95	131.04	118.09	96.36	128.11
F10	average	**2510**	2700	2945	2936	2915	2632	2939	2964	2913	2896	3963
	STD	**0.00**	0.00	21.97	25.11	23.49	10.36	28.45	31.25	24.36	1.16	366.49

The results for F1-F10 reveal that the exploitative tendencies have been enriched intensely, and the phases of exploitation are more effective in the BMSCD. The core reason is that BMO has an improved exploitation capacity. However, in initial iterations, more exploratory steps are required to diversify the agents. The disruption operator assists the methods in making more emphasis on the diversity of initial agents, and then, smoothly changeover from initial global search steps to the final exploitative trends. Also, the resulted method benefits from the exploitative advantages of both techniques.

The convergence trends of the developed BMO-based version are compared in detail against the other competitors in the following Figures. The trends show that the efficacy of BMO has significantly improved in terms of convergence speed. It shows a better trend for most of the cases, while we can see immature convergence for several competitors.

Table 9
Results of the average and STD measures obtained by the BMSCD and the hybrid techniques.

Func.	Average					STD				
	BMSCD	WCMFO	SSOSCA	BSOMFO	BMOSSA	BMSCD	WCMFO	SSOSCA	BSOMFO	BMOSSA
F1	**0.0E+0**	9.8E−12	**0.0E+0**	1.0E+0	7.8E−270	**0.0E+0**	1.2E−11	**0.0E+0**	3.3E+0	2.0E−268
F2	**0.0E+0**	1.1E−5	8.3E−159	8.3E−1	5.1E−141	**0.0E+0**	**0.0E+0**	**0.0E+0**	**0.0E+0**	**0.0E+0**
F3	**0.0E+0**	2.8E−3	1.4E−190	4.6E+3	2.6E−171	**0.0E+0**	3.4E−3	3.6E−189	2.1E+4	6.5E−170
F4	**0.0E+0**	8.2E−2	1.3E−130	1.4E+0	3.2E−137	**0.0E+0**	9.8E−2	1.9E−129	8.4E+0	7.7E−136
F5	3.1E+0	4.3E+1	2.6E+1	1.2E+2	**2.5E−2**	1.9E+1	7.6E+1	2.9E+1	3.6E+2	**1.9E−1**
F6	1.9E−2	**2.0E−4**	1.1E−1	1.4E+0	8.3E−4	8.4E−2	**3.6E−4**	5.0E−1	3.7E+0	4.4E−4
F7	**3.9E−5**	5.9E−2	6.4E−3	2.0E−2	2.5E−4	**1.1E−4**	6.8E−2	4.7E−2	6.2E−2	1.1E−3
F8	**0.0E+0**	1.1E−10	**0.0E+0**	7.3E+0	1.2E−287	**0.0E+0**	1.8E−10	**0.0E+0**	4.5E+1	2.4E−286
F9	**0.0E+0**	9.7E−23	**0.0E+0**	1.3E−2	**0.0E+0**	**0.0E+0**	1.7E−22	**0.0E+0**	1.1E−1	**0.0E+0**
F10	**0.0E+0**	1.6E−95	**0.0E+0**	4.0E−10	5.4E−209	**0.0E+0**	3.3E−95	**0.0E+0**	3.0E−9	1.3E−207
F11	−1.1E+3	−1.4E+3	−1.5E+3	−2.0E+3	**−1.1E+44**	−1.1E+3	−1.3E+3	−5.0E+2	−8.2E+2	**−8.9E+13**
F12	**0.0E+0**	2.8E+1	**0.0E+0**	5.4E+0	**0.0E+0**	**0.0E+0**	4.7E+1	**0.0E+0**	3.0E+1	**0.0E+0**
F13	**8.9E−16**	8.3E−1	2.9E−15	6.0E−1	8.9E−16	**8.9E−16**	1.7E+0	8.0E−15	1.6E+0	8.9E−16
F14	**0.0E+0**	2.6E−2	**0.0E+0**	6.4E−1	**0.0E+0**	**0.0E+0**	4.4E−2	**0.0E+0**	2.2E+0	**0.0E+0**
F15	2.1E−3	**6.4E−7**	4.2E−3	2.1E−1	9.6E−5	2.4E−2	**6.6E−7**	2.0E−2	1.4E+0	7.0E−6
F16	1.5E−2	**6.1E−5**	5.1E−1	2.1E+0	2.8E−4	8.3E−2	1.0E−4	1.3E+0	2.0E+1	**1.5E−5**
F17	6.5E−2	**9.8E−4**	2.4E+0	2.5E−1	3.7E−3	4.3E−1	**1.9E−3**	9.2E+0	1.6E+0	1.4E−2
F18	**0.0E+0**	3.6E−1	2.1E−159	2.5E−1	1.0E−147	**0.0E+0**	5.8E−1	3.6E−158	2.2E+0	1.7E−146
F19	**0.0E+0**	1.6E−10	1.0E−25	2.5E−2	**0.0E+0**	**0.0E+0**	1.6E−10	**0.0E+0**	1.8E−1	**0.0E+0**
F20	**0.0E+0**	6.9E−4	6.3E−125	2.2E+2	1.6E−83	**0.0E+0**	9.0E−4	1.6E−123	1.2E+3	3.9E−82
F21	−1.9E+1	−7.6E+0	−1.6E+1	**−2.8E+1**	−1.9E+1	−1.9E+1	−4.4E+0	−5.8E+0	**−2.7E+1**	−1.9E+1
F22	4.2E−3	**1.4E−4**	3.6E−1	2.3E−1	3.9E−4	**1.4E−3**	1.8E−3	5.9E+0	1.1E+0	1.9E−3
F23	**0.0E+0**	6.8E−9	**0.0E+0**	4.6E+6	8.7E−277	**0.0E+0**	7.3E−9	**0.0E+0**	5.1E+7	2.1E−275
F24	−4.9E−1	0.0E+0	0.0E+0	0.0E+0	**−9.2E−1**	−4.7E−15	2.2E−12	8.6E−11	6.5E−10	**−3.4E−1**
F25	**0.0E+0**	1.1E+0	1.0E−1	1.1E+0	1.5E−135	**0.0E+0**	1.2E+0	2.0E−1	5.9E+0	3.7E−134
F26	**0.0E+0**	3.9E−1	1.1E−1	2.5E−1	**0.0E+0**	**0.0E+0**	4.3E−1	5.0E−1	5.0E−1	**0.0E+0**
F27	2.6E+0	**1.0E+0**	1.0E+0	4.7E+0	1.0E+0	1.1E+1	**1.0E+0**	1.0E+0	9.8E+0	1.0E+0
F28	5.0E−4	**3.1E−4**	4.5E−4	2.6E−3	3.4E−4	**1.2E−4**	3.1E−4	1.3E−4	9.1E−3	5.1E−4
F29	**−1.0E+0**	−1.0E+0	−1.0E+0	−1.0E+0	−1.0E+0	−1.0E+0	−1.0E+0	−1.0E+0	**−1.0E+0**	−1.0E+0
F30	**4.0E−1**	4.0E−1	4.0E−1	4.0E−1	4.0E−1	4.0E−1	4.0E−1	4.0E−1	**4.0E−1**	4.0E−1
F31	3.0E+0	3.0E+0	3.0E+0	**3.0E+0**	3.0E+0	3.0E+0	3.0E+0	3.0E+0	**3.0E+0**	3.0E+0
F32	**−3.9E+0**	**−3.9E+0**	−3.8E+0	−3.9E+0	−3.9E+0	**−3.9E+0**	−3.9E+0	−3.1E+0	−3.9E+0	−3.9E+0
F33	−3.1E+0	**−3.3E+0**	−3.2E+0	−3.3E+0	−3.3E+0	−2.8E+0	**−3.3E+0**	−2.2E+0	−3.1E+0	−3.1E+0
F34	−7.5E+0	−2.7E+0	−4.1E+0	−5.2E+0	**−1.0E+1**	−4.3E+0	−2.7E+0	−1.6E+0	−2.7E+0	**−9.9E+0**
F35	−8.8E+0	**−1.0E+1**	−4.1E+0	−5.8E+0	**−1.0E+1**	−4.4E+0	**−1.0E+1**	−1.7E+0	−2.8E+0	−1.0E+1
F36	−8.1E+0	**−1.1E+1**	−4.0E+0	−5.5E+0	−1.0E+1	−4.6E+0	**−1.1E+1**	−1.6E+0	−2.9E+0	−1.0E+1

5.6. Sensitivity analysis of the BMSCD

This section shows the impact of the parameters on the proposed method. Two parameters were analyzed as follows, the dimension size of the problem and the SCA parameter (a) over 15 benchmark functions and four measures were calculated (i.e. Average, STD, Worst, and Best) for the fitness values. Tables 6–7 report these results. In Table 6 four dimensions sizes are tested (i.e. 15, 30, 50, and 500) using 1000 iterations. From this table that can be noticed, all dimensions showed the same performance in 10 out of 15 functions in all measures. That means the different dimensions did not affect the performance of the algorithm in 67% of the functions; whereas, the rest of the functions showed slight differences.

In addition, the impact of the SCA parameter (a) was tested using two values (2 and 1.5) as in Table 7. The BMSCD showed the same performance in 67% of the functions whereas, the algorithm with parameter value (1.5) outperformed in 4 out of 5 functions. However, in this paper, we followed the suggested setting by the original study of SCA (Mirjalili, 2016).

5.7. Evaluate the BMSCD using CEC2020

In this section, the proposed algorithm is evaluated using CEC2020 benchmark functions.[1] Ten algorithms are used in the comparison (BMO, SCA, GWO, SSA, ABC, MFO, WOA, TLBO, DE, and COBIDE). The parameter settings are set as the previous experiment except dimension

[1] https://www.ntu.edu.sg/home/epnsugan/index_files/CEC2020/CEC2020-2.htm

Table 10
Results of the best and worst fitness functions measures obtained by the BMSCD and the hybrid techniques.

Func.	The best fitness value					The best worst value				
	BMSCD	WCMFO	SSOSCA	BSOMFO	BMOSSA	BMSCD	WCMFO	SSOSCA	BSOMFO	BMOSSA
F1	**0.0E+0**	7.2E−12	**0.0E+0**	6.4E−3	**0.0E+0**	**0.0E+0**	1.2E−11	**0.0E+0**	3.3E+0	2.0E−268
F2	**0.0E+0**	**0.0E+0**	**0.0E+0**	**0.0E+0**	**0.0E+0**	**0.0E+0**	**0.0E+0**	**0.0E+0**	**0.0E+0**	**0.0E+0**
F3	**0.0E+0**	2.3E−3	5.2E−242	4.0E+0	**0.0E+0**	**0.0E+0**	3.4E−3	3.6E−189	2.1E+4	6.5E−170
F4	**0.0E+0**	6.7E−2	4.2E−143	1.3E−1	6.4E−165	**0.0E+0**	9.8E−2	1.9E−129	8.4E+0	7.7E−136
F5	3.9E−4	1.0E+1	2.5E+1	3.9E+0	**5.6E−6**	1.9E+1	7.6E+1	2.9E+1	3.6E+2	**1.9E−1**
F6	8.5E−4	3.8E−5	1.1E−6	6.3E−2	**3.7E−8**	8.4E−2	**3.6E−4**	5.0E−1	3.7E+0	4.4E−4
F7	**3.2E−6**	5.0E−2	4.2E−5	2.5E−3	1.4E−5	**1.1E−4**	6.8E−2	4.7E−2	6.2E−2	1.1E−3
F8	**0.0E+0**	3.6E−11	**0.0E+0**	3.0E−2	**0.0E+0**	**0.0E+0**	1.8E−10	**0.0E+0**	4.5E+1	2.4E−286
F9	**0.0E+0**	2.5E−23	**0.0E+0**	1.9E−8	**0.0E+0**	**0.0E+0**	1.7E−22	**0.0E+0**	1.1E−1	**0.0E+0**
F10	**0.0E+0**	3.4E−98	**0.0E+0**	8.6E−13	2.7E−242	**0.0E+0**	3.3E−95	**0.0E+0**	3.0E−9	1.3E−207
F11	−1.1E+3	−1.5E+3	−1.6E+3	−3.7E+3	**−1.5E+45**	−1.1E+3	−1.3E+3	−5.0E+2	−8.2E+2	**−8.9E+13**
F12	**0.0E+0**	1.0E+1	**0.0E+0**	2.0E−2	**0.0E+0**	**0.0E+0**	4.7E+1	**0.0E+0**	3.0E+1	**0.0E+0**
F13	**8.9E−16**	3.5E−6	**8.9E−16**	1.4E−2	**8.9E−16**	**8.9E−16**	1.7E+0	8.0E−15	1.6E+0	8.9E−16
F14	**0.0E+0**	7.4E−3	**0.0E+0**	3.2E−2	**0.0E+0**	**0.0E+0**	4.4E−2	**0.0E+0**	2.2E+0	**0.0E+0**
F15	8.8E−7	6.3E−7	1.5E−6	4.6E−5	**9.6E−8**	2.4E−2	**6.6E−7**	2.0E−2	1.4E+0	7.0E−6
F16	3.4E−5	2.2E−5	1.1E−1	3.7E−4	**1.9E−6**	8.3E−2	1.0E−4	1.3E+0	2.0E+1	**1.5E−5**
F17	1.4E−3	6.4E−5	4.5E−4	1.7E−3	**7.1E−6**	4.3E−1	**1.9E−3**	9.2E+0	1.6E+0	1.4E−2
F18	**0.0E+0**	1.4E−1	1.3E−172	6.0E−3	2.6E−167	**0.0E+0**	5.8E−1	3.6E−158	2.2E+0	1.7E−146
F19	**0.0E+0**	1.5E−10	**0.0E+0**	6.6E−5	**0.0E+0**	**0.0E+0**	1.6E−10	**0.0E+0**	1.8E−1	**0.0E+0**
F20	**0.0E+0**	4.9E−4	1.8E−151	2.9E−1	5.0E−291	**0.0E+0**	9.0E−4	1.6E−123	1.2E+3	3.9E−82
F21	−1.9E+1	−1.1E+1	−2.9E+1	**−2.9E+1**	−1.9E+1	−1.9E+1	−4.4E+0	−5.8E+0	**−2.7E+1**	−1.9E+1
F22	**1.3E−7**	4.2E−5	2.7E−5	3.9E−4	1.8E−6	**1.4E−3**	1.8E−3	5.9E+0	1.1E+0	1.9E−3
F23	**0.0E+0**	6.3E−9	**0.0E+0**	1.7E+3	**0.0E+0**	**0.0E+0**	7.3E−9	**0.0E+0**	5.1E+7	2.1E−275
F24	**−1.0E+0**	0.0E+0	0.0E+0	0.0E+0	−1.0E+0	−4.7E−15	2.2E−12	8.6E−11	6.5E−10	**−3.4E−1**
F25	**0.0E+0**	1.0E+0	1.0E−1	2.0E−1	**0.0E+0**	**0.0E+0**	1.2E+0	2.0E−1	5.9E+0	3.7E−134
F26	**0.0E+0**	3.5E−1	9.7E−3	3.7E−2	**0.0E+0**	**0.0E+0**	4.3E−1	5.0E−1	5.0E−1	**0.0E+0**
F27	1.0E+0	**1.0E+0**	**1.0E+0**	2.0E+0	1.0E+0	1.1E+1	**1.0E+0**	1.0E+0	9.8E+0	1.0E+0
F28	3.1E−4	3.1E−4	**3.1E−4**	4.5E−4	3.1E−4	**1.2E−4**	3.1E−4	1.3E−4	9.1E−3	5.1E−4
F29	**−1.0E+0**	−1.0E+0	−1.0E+0	−1.0E+0	−1.0E+0	−1.0E+0	−1.0E+0	−1.0E+0	**−1.0E+0**	−1.0E+0
F30	4.0E−1	4.0E−1	4.0E−1	**4.0E−1**	4.0E−1	4.0E−1	4.0E−1	4.0E−1	**4.0E−1**	4.0E−1
F31	3.0E+0	3.0E+0	3.0E+0	**3.0E+0**	3.0E+0	3.0E+0	3.0E+0	3.0E+0	**3.0E+0**	3.0E+0
F32	**−3.9E+0**	−3.9E+0	−3.9E+0	−3.9E+0	−3.9E+0	**−3.9E+0**	−3.9E+0	−3.1E+0	−3.9E+0	−3.9E+0
F33	−3.3E+0	−3.3E+0	−3.3E+0	**−3.3E+0**	−3.3E+0	−2.8E+0	**−3.3E+0**	−2.2E+0	−3.1E+0	−3.1E+0
F34	−1.0E+1	−2.7E+0	−8.1E+0	**−1.0E+1**	−1.0E+1	−4.3E+0	−2.7E+0	−1.6E+0	−2.7E+0	**−9.9E+0**
F35	−1.0E+1	−1.0E+1	−8.6E+0	**−1.0E+1**	−1.0E+1	−4.4E+0	**−1.0E+1**	−1.7E+0	−2.8E+0	−1.0E+1
F36	**−1.1E+1**	−1.1E+1	−9.3E+0	−1.1E+1	−1.1E+1	−4.6E+0	**−1.1E+1**	−1.6E+0	−2.9E+0	−1.0E+1

is 10 and the population size is 100. All algorithms are applied 30 runs to calculate the statistical results. Table 8 shows the average and the STD values for these runs.

According to the results in this table, the BMSCD algorithms showed good results. In detail, in terms of the average measure of the fitness function, the BMSCD outperformed all other methods in five functions (i.e., F5, F6, F8, F9 and F10). Also DE provides results better than other methods at two functions (i.e., F1 and F2), as well as, at F7 TLBO has better results. In F4, all algorithms except for SSA, MFO, and COBIDE obtained the best value. In general, the algorithms can be ordered as follows, BMSCD, DE, ABC, TLBO, SSA, BMO, GWO, SCA, WOA, and MFO while the worst algorithm in all function was the COBIDE.

In terms of the STD, the BMSCD algorithm was the most stable algorithm in six functions.; it also obtained the lowest STD values in three functions similar to BMO. DE obtain the second rank, followed by ABC and TLBO. At the same time, the worst stability was shown by COBIDE algorithm.

5.8. Comparison with hybrid techniques

In this section, we compare the results of the developed BMSCD with other hybrid techniques namely the hybrid water cycle and moth flame Optimization algorithms (WCMFO) (Khalilpourazari & Khalilpourazary, 2019), modified spherical search optimizer based on sine–cosine algorithm (SSOSCA) (Naji Alwerfali et al., 2020), hybrid brain storm optimization with moth flame optimization (BSOMFO) (Ibrahim, Abd Elaziz, Ewees, Selim, & Lu, 2018a), and hybrid harris hawks optimizer and slap swarm algorithm (BMOSSA) (Abd Elaziz, Heidari, Fujita, & Moayedi, 2020) in solving the global optimization problem (i.e. well-known optimization benchmark functions). These hybrid techniques are selected because they showed good results and competitive performance in the literature.

Table 9 depicts the results of the average and STD measures of the fitness function values for all hybrid techniques against the proposed method. From Table 9 we can see that the BMSCD achieved the best results in 20 out of 36 functions; in detail, it got the lowest average in 10 functions and performed similarly to other techniques in 10 functions. Followed by the WCMFO it obtained the best results in 9 functions whereas, the BMOSSA and SSOSCA came in the third and fourth rank, respectively; The BMOSSA obtained the best values in 4 functions and the SSOSCA showed competitive results in 7 functions. The worst results were recorded by the BSOMFO therefore, it was ranked last.

The BMSCD was also the most stable method based on the STD results as in Table 9. From this table, the BMSCD got the smallest STD in 18 functions followed by the WCMFO, BMOSSA, and SSOSCA with 14, 13, and 10 functions, respectively; whereas, the BSOMFO recorded the bad STD values in 89% of all functions.

In addition, the best and worst fitness value measures were listed in Table 10. According to the results in the table, the BMSCD showed a good ability to reach the minimum values of the fitness functions, it got the best results in 61% of the functions followed by BMOSSA with 50% of the functions. The SSOSCA and BSOMFO got the best values in 33% and 22% and were ranked third and fourth, respectively. The worst technique was the WCMFO, it got the best values in only 2 functions.

These results were also reported by the worst fitness value measure as in Table 10. As shown in the table, the BMSCD method outperformed the other techniques in 28% of all functions and works similarly to them in 11 functions. The second rank was achieved by the BMOSSA followed by SSOSCA, WCMFO, and BSOMFO, respectively.

In general, the proposed BMSCD method showed good ability to solve global optimization problems against hybrid techniques especially in the best fitness value measure.

From the above discussions, it can be observed that the proposed BMSCD method has a faster convergence rate because it performs a higher level exploration phase compared to the other methods as well as the hybrid techniques. The core benefits of BMO compared to other techniques are also available in the proposed BMO-based method. Also, it can bring more variety of patterns into the searching process because of the utilized components of SCA.

6. Experimental series 2: Automatic clustering using BMSCD

In this experimental series, the performance of the BMSCD as an automatic data clustering method is evaluated. The dataset description, performance metric, and the comparison results are discussed in the following sections.

6.1. Dataset description

In this paper, fourteen datasets with variant samples, features, and classes numbers were used to check and assess the quality and the ability of the proposed BMSCD in clustering data. These datasets were chosen due to their widely used in many previous studies as well as they are also used as benchmark datasets. Table 11 presents the details of these datasets and their details.

6.2. Performance measures

Eight performance measures are applied to check the quality of the proposed BMSCD compared to other algorithms. These measures are Silhouette index (SI), Dunn index (DI), Davies–Bouldin index (DB), Hartigan (HA), weighted inter/intra (WTER), adjusted rank index (AR), Hubert index, and normalized mutual information (NMI).

The best value in SI, DI, AR, Hubert, and NMI is the highest value; whereas, the lowest value is the best in DB, HA, and WTER. These measures are applied to all benchmark datasets for all methods as listed in Section 6.4.

Table 11
Datasets description used in the clustering experiment.

Name	Size of sample	Number of features	Number of clusters
Wine	178	13	3
Iris	150	4	3
Ecoli	336	7	8
Glass	214	10	7
Haberman's Survival	306	4	2
Liver Disorders	345	6	16
IonosphereEW	351	34	2
Lymphography	148	18	2
M-of-n	1000	13	2
PenglungEW3	73	325	2
Brain-T21	50	10368	4
CongressEW1	435	16	2
KrvskpEW4	3196	36	2
Gesture	9900	50	5

6.3. Parameter settings

The parameter settings of the proposed BMSCD along and the compared method are set as in the previous global optimization experiment. The proposed BMSCD was compared with eleven algorithms namely BMO, SCA, GWO, MFO, WOA, SSA, PSO, PSOFA (Aydilek, 2018), DE, MLSHADE-SPA (we refer to it as MLSHSPA) (Hadi, Mohamed, & Jambi, 2019), and LSHSPACMA (Mohamed, Hadi, Fattouh, & Jambi, 2017). All results were computed over 30 independent runs. The population size was set to 30, the maximum number of iterations was set to 100 for each algorithm, and $\epsilon = 0.4$ (defined in Eq. (18)). The parameters of each algorithm are set as their original implementations.

6.4. Experiment results and discussion

In this section, the experiment results of the proposed BMSCD and the compared algorithms are given in Table 12 and Figs. 3–5. Table 12 shows the results of each algorithm overall the dataset. Whereas, Fig. 3 illustrates the normalized average of the measures overall datasets for each algorithm (the averages are normalized for illustration purpose). Fig. 4 illustrates the ranking of all algorithms in each measure. Fig. 5 shows the achievement ratios of each algorithm in all measures.

Inspecting the results in Table 12, it can be seen that in terms of SI measure, the proposed BMSCD is outperformed all methods in 5 out of 14 datasets namely IonosphereEW, Ecol, iris, Live, and KrvskpEW4. It also achieved the best average for all datasets equals 0.3063 in the SI measure. Followed by PSO, SCA, BMO, and MLSHSPA with averages equal to 0.2671, 0.2862, 0.2502, and 0.2222 respectively. Whereas, the rest of the methods were ranked as follows: LSHSPACMA, WOA, PSOFA, GWO, and SSA. The DE and MFO recorded the worst average of the SI measure.

The BMSCD also ranked first in DI measure. It achieved the best value in 29% of all datasets, and it reached average equals to 0.281; while the PSO was ranked second with average equals to 0.280 followed by PSO, PSOFA, BMO, WOA, and SCA with 0.268, 0.245, 0.241, 0.235, and 0.234, respectively, while the MFO showed the worst results in all datasets.

Besides, when analyzing the average of the DB measure that can be concluded, the best average of DB was obtained by the proposed BMSCD method, it reached 0.568 (in this measure the smallest value is the best) and it achieved the best results in 71% of all datasets. The PSO, BMO, SCA, and GWO were ranked second, third, forth, and fifth with 0.701, 0.719, 0.744, and 0.805, respectively followed by the WOA,

Table 12
Results of all algorithms for the clustering experiment overall datasets.

Dataset	Algorithm	SI	DI	DB	HA	WTER	AR	Hubert	NMI
IonosphereEW	BMSCD	**0.2766**	0.3980	**0.4159**	**3.208**	**0.4709**	**0.3072**	**0.376**	0.728
	BMO	0.2663	0.3527	0.4391	3.802	0.4897	0.2722	0.293	0.728
	SCA	0.2750	0.3800	0.5060	4.220	0.4930	0.2943	0.348	0.726
	GWO	0.2730	0.3630	0.5240	3.440	0.4940	0.2526	0.233	0.724
	MFO	0.1210	0.1450	2.1290	184.62	0.5750	0.2960	0.229	**1.378**
	WOA	0.2480	0.3770	0.5390	10.150	0.5310	0.1406	0.319	0.713
	SSA	0.2470	0.2960	1.2990	19.880	0.4870	0.1063	0.212	0.582
	PSO	0.1760	0.2930	1.3578	192.68	0.5856	0.2439	0.327	1.269
	PSOFA	0.2003	**0.4083**	1.4183	184.01	0.5796	0.2555	0.221	1.169
	DE	0.1368	0.1712	1.7025	129.64	0.5760	0.2290	0.265	0.927
	MLSHSPA	0.1344	0.2337	1.6150	182.42	0.5402	0.1676	0.352	1.258
	LSHSPACMA	0.1407	0.2652	1.7578	132.58	0.6050	0.2309	0.372	1.120
Ecol	BMOSCD	**0.1925**	**0.1169**	**0.4935**	4.916	0.3740	**0.7862**	0.811	2.335
	BMO	0.1787	0.1146	0.7645	5.238	0.4308	0.7046	0.803	2.234
	SCA	0.1880	0.1150	0.8150	5.230	0.3840	0.7476	0.746	2.715
	GWO	0.1570	0.1100	0.7910	7.690	0.5290	0.7111	0.704	2.771
	MFO	0.0690	0.0480	1.2270	463.1	0.6000	0.6376	0.811	2.738
	WOA	0.1820	0.1150	0.7960	6.260	0.4810	0.7380	**0.869**	2.685
	SSA	0.1600	0.1160	0.7750	14.57	0.4000	0.6600	0.848	2.702
	PSO	0.1369	0.1153	0.5253	4.801	0.3909	0.6472	0.817	2.722
	PSOFA	0.1515	0.1104	0.5289	6.660	0.5924	0.6979	0.813	2.768
	DE	0.1771	0.1150	0.5440	6.371	**0.2026**	0.7517	0.782	2.689
	MLSHSPA	0.1483	0.1121	0.5092	**3.662**	0.4899	0.7240	0.767	2.773
	LSHSPACMA	0.1738	0.1148	0.5101	8.038	0.5225	0.6496	0.757	**2.774**
Glas	BMOSCD	0.3611	0.0359	0.3294	**11398**	0.5922	**0.3947**	**0.623**	**3.664**
	BMO	0.3469	0.0317	0.4093	11438	0.6034	0.3902	0.606	3.307
	SCA	0.3400	0.0300	0.4040	11983	0.5720	0.1751	0.428	2.350
	GWO	0.3740	0.0350	0.4100	14636	0.5970	0.2237	0.424	2.414
	MFO	0.3590	0.0370	0.4470	13791	0.5920	0.1679	0.383	2.379
	WOA	0.3790	0.0310	0.3950	12138	0.6060	0.1800	0.422	2.351
	SSA	0.3680	0.0350	0.4190	11613	0.6020	0.1747	0.389	2.407
	PSO	0.3751	0.0277	0.3271	11686	0.5594	0.1984	0.463	2.510
	PSOFA	**0.3899**	**0.0424**	0.2882	11800	**0.2838**	0.2169	0.357	2.370
	DE	0.2346	0.0283	0.3787	11723	0.3110	0.1971	0.401	2.371
	MLSHSPA	0.3638	0.0311	**0.2881**	11715	0.3879	0.1791	0.375	2.361
	LSHSPACMA	0.3730	0.0358	0.3400	11692	0.6080	0.1612	0.387	2.378
Habe	BMOSCD	0.3826	0.1655	0.3114	195.94	0.4808	**2.8919**	**0.314**	**0.359**
	BMO	0.3424	0.1452	0.4097	279.67	0.5712	2.5593	0.219	0.284
	SCA	**0.4360**	0.1370	0.1790	**13.860**	0.5050	0.2089	0.298	0.214
	GWO	0.3170	0.1410	0.4490	331.56	0.5540	0.2025	0.273	0.200
	MFO	0.2260	0.1150	0.7100	866.37	0.5410	0.2201	0.293	0.191
	WOA	0.2930	0.1580	0.4450	418.41	0.6020	0.2213	0.244	0.264
	SSA	0.2430	0.1090	0.6420	313.92	0.5730	0.1910	0.299	0.263
	PSO	0.3833	0.1388	0.7653	246.93	0.6276	0.2159	0.246	0.246
	PSOFA	0.2770	0.1393	0.6366	274.45	**0.3557**	0.2355	0.258	0.244
	DE	0.2800	0.1143	0.4826	249.44	0.5448	0.1959	0.285	0.200
	MLSHSPA	0.2634	**0.2080**	0.6852	248.37	0.7055	0.2373	0.289	0.206
	LSHSPACMA	0.3932	0.1457	**0.1076**	255.16	0.6852	0.2364	0.259	0.262
Iris	BMOSCD	**0.5164**	0.1280	**0.1489**	505.11	0.5622	0.7520	**0.838**	**0.769**
	BMO	0.3965	0.1186	0.2517	**414.07**	0.5887	0.7466	0.769	0.685
	SCA	0.3400	0.0880	0.2620	516.29	0.6320	0.7840	0.793	0.670

(*continued on next page*)

Table 12 (*continued*).

Dataset	Algorithm	SI	DI	DB	HA	WTER	AR	Hubert	NMI
	GWO	0.4680	0.1110	0.2350	521.39	0.7220	0.7474	0.777	0.651
	MFO	0.1830	0.1230	0.7540	1292.86	0.6370	0.7906	0.784	0.633
	WOA	0.3290	0.1000	0.2780	537.62	0.6440	0.7778	0.763	0.656
	SSA	0.3690	**0.1300**	0.2910	516.87	0.6690	**0.7984**	0.756	0.637
	PSO	0.3064	0.1083	0.1541	505.33	0.6437	0.7451	0.794	0.665
	PSOFA	0.5077	0.0435	0.2383	514.46	**0.5536**	0.7519	0.788	0.636
	DE	0.1190	0.0840	0.2372	511.22	0.5640	0.7826	0.790	0.669
	MLSHSPA	0.3773	0.0840	0.1829	519.46	0.6442	0.7557	0.772	0.633
	LSHSPACMA	0.3057	0.1166	0.2156	506.02	0.6372	0.7723	0.754	0.671
Live	BMOSCD	**0.6758**	0.1590	0.3372	203.03	**0.5810**	**0.4998**	**0.506**	**2.456**
	BMO	0.4807	**0.2396**	0.4776	400.17	0.6864	0.4954	0.505	1.525
	SCA	0.6460	0.1500	0.4020	231.58	0.6980	0.3610	0.336	2.123
	GWO	0.4810	0.1130	0.4740	383.19	0.6700	0.3592	0.480	2.009
	MFO	0.3250	0.0580	0.7930	703.66	0.6650	0.4071	0.482	1.378
	WOA	0.4960	0.1150	0.5290	459.94	0.6690	0.3813	0.447	1.943
	SSA	0.5060	0.0960	0.6370	480.30	0.5950	0.2977	0.404	1.910
	PSO	0.3526	0.1555	0.3477	209.94	0.6315	0.3993	0.435	1.491
	PSOFA	0.5364	0.2154	0.3175	205.68	0.6104	0.3037	0.420	1.589
	DE	0.6305	0.1096	0.3356	301.48	0.6339	0.3326	0.358	1.808
	MLSHSPA	0.5974	0.1463	0.3888	422.79	0.6397	0.4444	0.461	2.112
	LSHSPACMA	0.5418	0.1020	**0.3069**	**201.51**	0.6720	0.4161	0.465	1.691
Wine	BMOSCD	0.5602	0.1523	**0.1014**	3.837	**0.3620**	**0.3915**	**0.502**	**1.659**
	BMO	0.3156	**0.1892**	0.1193	**3.741**	0.5746	0.3419	0.427	1.267
	SCA	0.6020	0.1310	0.1310	21.44	0.5880	0.3558	0.442	1.556
	GWO	0.3800	0.0970	0.1190	4.780	0.4490	0.3654	0.432	1.608
	MFO	0.1720	0.0710	0.7620	938.2	0.5050	0.3699	0.497	1.547
	WOA	0.5400	0.1210	0.1360	19.61	0.5980	0.3400	0.426	1.557
	SSA	0.2280	0.0790	0.1190	8.26	0.5510	0.3353	0.457	1.530
	PSO	**0.6761**	0.0713	0.1124	7.18	0.5113	0.3311	0.431	1.533
	PSOFA	0.3593	0.1189	0.1425	14.52	0.4871	0.3319	0.401	1.615
	DE	0.2185	0.1183	0.1783	8.62	0.4530	0.3358	0.467	1.550
	MLSHSPA	0.5120	0.0985	0.1589	9.59	0.4541	0.3709	0.489	1.616
	LSHSPACMA	0.5343	0.1225	0.1472	13.31	0.4998	0.3488	0.442	1.536
Isolet	BMOSCD	0.2716	**0.4780**	**0.3070**	**2.159**	0.4093	0.8613	0.898	**1.547**
	BMO	0.2663	0.3527	0.4391	3.802	0.4897	0.6543	0.965	0.971
	SCA	0.2600	0.3710	0.4580	3.210	0.4410	0.7753	0.839	0.735
	GWO	0.2820	0.4010	0.4050	3.820	0.5090	0.8661	**0.967**	0.735
	MFO	0.1210	0.1110	2.1290	183.01	0.5880	0.7869	0.958	1.259
	WOA	0.2750	0.4170	0.5170	4.550	0.4950	0.8490	0.861	0.679
	SSA	0.2350	0.2460	1.2750	35.94	0.5340	0.8171	0.853	0.618
	PSO	**0.2959**	0.4474	0.3114	2.592	**0.3846**	0.7710	0.868	1.325
	PSOFA	0.1306	0.2307	1.4602	200.21	0.5442	0.7807	0.851	1.325
	DE	0.1398	0.2241	1.6616	149.74	0.5658	**0.9247**	0.954	1.101
	MLSHSPA	0.1446	0.1438	1.6812	158.11	0.5778	0.8949	0.846	1.230
	LSHSPACMA	0.1672	0.2008	1.5898	136.12	0.5752	0.7507	0.960	0.969
Lymphography3	BMOSCD	0.2464	0.2317	**0.3206**	**2.734**	**0.3510**	−1.9968	**0.279**	2.575
	BMO	**0.2559**	0.2325	0.3602	2.913	0.4930	−2.0319	0.158	0.957
	SCA	0.2540	0.2290	0.3810	2.970	0.4500	0.1045	0.124	0.509
	GWO	0.2470	0.2300	0.3730	3.610	0.5030	0.1751	0.210	0.506
	MFO	0.0620	0.2310	1.3330	175.55	0.5240	0.1698	0.193	**2.639**
	WOA	0.2370	0.2180	0.5250	5.270	0.3580	0.1919	0.182	0.490
	SSA	0.1630	0.2320	0.9050	32.10	0.4580	0.1909	0.196	0.694

(*continued on next page*)

Table 12 (*continued*).

Dataset	Algorithm	SI	DI	DB	HA	WTER	AR	Hubert	NMI
	PSO	0.1536	0.2186	0.3385	2.972	0.3576	**0.2000**	0.141	**2.639**
	PSOFA	0.0465	0.2506	0.9733	180.93	0.5557	0.1801	0.185	2.430
	DE	0.1182	0.2061	1.0168	134.24	0.5510	0.1323	0.213	1.757
	MLSHSPA	0.1072	**0.2856**	1.0835	159.94	0.5341	0.1025	0.200	2.020
	LSHSPACMA	0.0915	0.2730	1.0628	140.03	0.5210	0.1733	0.155	1.934
M-of-n3	BMOSCD	−0.0418	**0.3865**	**1.9347**	**158.06**	0.4956	0.2414	0.285	2.296
	BMO	−0.0450	0.3130	2.1615	165.13	**0.4329**	0.2405	0.260	2.156
	SCA	−0.0750	0.3680	2.1280	257.82	0.5890	0.2268	0.213	2.147
	GWO	−0.0470	0.3320	2.2470	298.11	0.6020	0.2566	0.197	2.277
	MFO	0.0360	0.3180	2.6040	345.66	0.6110	0.1069	0.290	2.157
	WOA	−0.0570	0.3400	2.1960	288.45	0.5890	0.1202	0.254	2.264
	SSA	−0.0510	0.3560	2.2460	297.76	0.6000	0.1083	0.207	2.345
	PSO	0.0421	0.3631	2.0403	253.33	0.5115	0.2670	0.181	2.536
	PSOFA	**0.0467**	0.3414	2.2482	445.57	0.6226	0.2119	**0.307**	2.494
	DE	0.0343	0.3340	2.4193	409.30	0.6179	0.1344	0.254	2.458
	MLSHSPA	0.0411	0.3340	2.3114	439.12	0.6202	**0.2878**	0.233	**2.539**
	LSHSPACMA	0.0396	0.3414	2.3494	427.42	0.6188	0.2084	0.172	2.472
PenglungEW3	BMOSCD	0.0495	0.4079	1.2372	**2.084**	0.3809	**0.6951**	**0.742**	3.636
	BMO	0.0416	0.3726	1.3559	4.314	0.3345	0.5931	0.622	2.849
	SCA	**0.0710**	0.3890	1.4530	5.150	0.3470	0.5897	0.672	1.547
	GWO	0.0490	0.3750	1.3060	3.870	0.3100	0.6252	0.708	4.546
	MFO	−0.0440	0.3700	2.1390	39.88	0.3900	0.6562	0.698	**4.755**
	WOA	0.0480	0.3850	1.4810	9.520	0.3340	0.6426	0.664	1.340
	SSA	0.0430	0.3890	1.7260	18.99	0.3660	0.5801	0.677	1.683
	PSO	0.0676	**0.5041**	**1.1910**	2.259	**0.2306**	0.6257	0.666	**4.755**
	PSOFA	0.0262	0.4666	1.4273	58.83	0.4012	0.6022	0.725	4.668
	DE	−0.0147	0.4071	1.8240	46.86	0.4074	0.6171	0.683	4.744
	MLSHSPA	0.0106	0.4926	1.7876	55.64	0.4220	0.6241	0.706	4.572
	LSHSPACMA	0.0123	0.4238	1.8569	55.09	0.4066	0.5861	0.693	4.666
Brain-T21	BMOSCD	0.3837	**0.6022**	**0.4587**	2.744	0.5331	0.6235	**0.792**	**2.593**
	BMO	0.2865	0.3689	0.4760	3.672	0.5372	0.6589	0.729	1.679
	SCA	0.2590	0.4500	0.5060	2.850	0.5310	0.6390	0.669	1.855
	GWO	−0.0030	0.2910	0.5310	1.560	0.5380	0.6651	0.715	1.912
	MFO	0.0670	0.3250	1.2200	27.34	0.5960	0.6675	0.672	1.886
	WOA	0.1010	0.3680	0.5080	1.840	0.5400	0.6304	0.651	2.012
	SSA	0.1190	0.3970	0.5080	1.800	0.4930	0.6655	0.721	1.967
	PSO	0.2298	0.4345	0.7107	2.822	0.5551	0.6649	0.690	1.871
	PSOFA	0.1630	0.3067	0.7392	3.864	0.5618	0.6170	0.644	2.030
	DE	0.3519	0.3739	0.5935	3.844	**0.3990**	**0.6960**	0.702	1.920
	MLSHSPA	**0.4090**	0.4287	0.5886	**1.057**	0.4425	0.6308	0.679	1.962
	LSHSPACMA	0.2761	0.5052	0.4961	2.057	0.5237	0.6339	0.671	2.000
CongressEW1	BMOSCD	0.2785	0.3312	**0.6172**	84.10	**0.4109**	**0.3029**	0.258	**2.032**
	BMO	0.2423	0.2675	0.9079	123.10	0.5303	0.2763	0.243	1.277
	SCA	0.2760	0.2370	0.9170	97.50	0.5680	0.2598	**0.292**	1.563
	GWO	−0.0190	0.3040	1.6220	144.27	0.5480	0.2224	0.218	1.273
	MFO	0.0390	0.2300	1.9480	215.94	0.5430	0.1953	0.211	1.328
	WOA	0.0310	**0.3430**	1.4390	128.10	0.5550	0.1783	0.247	1.898
	SSA	0.0390	0.2760	1.5940	191.66	0.5420	0.1921	0.251	1.152
	PSO	**0.4202**	0.2230	0.6703	**40.97**	0.4840	0.2541	0.256	1.328
	PSOFA	0.0695	0.2538	1.1582	212.39	0.5506	0.2493	0.257	0.860
	DE	0.0565	0.2804	1.4738	146.40	0.6613	0.2588	0.265	0.640

(*continued on next page*)

Table 12 (*continued*).

Dataset	Algorithm	SI	DI	DB	HA	WTER	AR	Hubert	NMI
	MLSHSPA	0.0421	0.2474	1.5049	192.85	0.5218	0.2552	0.263	0.811
	LSHSPACMA	0.0291	0.3092	1.3513	131.13	0.6004	0.2474	0.261	0.829
KrvskpEW4	BMOSCD	**0.1350**	0.3441	**0.9416**	**32.06**	**0.4040**	0.6354	**0.756**	**1.897**
	BMO	0.1280	0.3310	1.5001	39.90	0.5651	**0.7226**	0.571	1.545
	SCA	0.1350	0.2120	1.8750	174.39	0.5120	0.5921	0.617	1.597
	GWO	0.0230	0.2160	1.7860	247.66	0.4810	0.5054	0.566	1.592
	MFO	−0.0300	0.2230	2.5180	535.56	0.5600	0.5903	0.544	1.672
	WOA	−0.0260	0.2800	2.0920	359.89	0.5200	0.5190	0.611	1.629
	SSA	−0.0040	0.2300	2.1460	512.58	0.5560	0.5993	0.598	1.695
	PSO	0.1247	**0.8208**	0.9572	33.04	0.4185	0.5228	0.625	1.700
	PSOFA	−0.0476	**0.8208**	1.7063	509.56	0.5224	0.5944	0.625	1.566
	DE	−0.0352	0.3353	1.8896	295.92	0.6678	0.5811	0.533	1.724
	MLSHSPA	−0.0399	0.2232	1.5242	246.03	0.4476	0.5924	0.558	1.623
	LSHSPACMA	0.0158	0.3197	1.4985	187.39	0.6944	0.5719	0.537	1.709

Table 13
Results of Friedman test and the corresponding rank.

Algorithm	DB	AR	DI	Ha	Hubert	NMI	SI	Wter
HHOSCAD	**1.64 (1)**	**9.57 (1)**	**10.57 (1)**	**1.86 (1)**	**10.93 (1)**	**9.71 (1)**	**10 (1)**	**2.57 (1)**
HHO	4.86 (3)	7.14 (3)	6.93 (4)	4 (3)	6.21 (6)	5 (11)	7.43 (4)	6.21 (5)
SCA	5.79 (4)	6.57 (7)	6.39 (8)	5.07 (4)	5.93 (9)	4.89 (12)	8.79 (2)	5.68 (2)
GWO	6.25 (6)	6.64 (4)	5.18 (10)	6.36 (5)	6.43 (5)	5.54 (8)	6.79 (5)	6.64 (7)
MFO	11.93 (12)	6.64 (4)	3.29 (12)	11.14 (12)	6.71 (3)	6.46 (6)	2.39 (12)	8.14 (11)
WOA	7.18 (8)	5.79 (9)	6.71 (6)	7.43 (8)	5.21 (12)	5.29 (9)	6.64 (6)	7.25 (9)
SSA	8.14 (10)	4.79 (12)	6.43 (7)	7.36 (7)	6 (8)	5.14 (10)	5.39 (10)	6.14 (4)
PSO	4 (2)	5.93 (8)	6.82 (5)	3.79 (2)	6.21 (6)	8 (2)	8 (3)	5 (3)
PSOFA	6.5 (7)	5.64 (10)	7.36 (3)	9.36 (11)	5.71 (10)	7.18 (4)	5.93 (8)	7.07 (8)
DE	8.21 (11)	6.64 (4)	4.75 (11)	7.43 (8)	6.79 (2)	6.29 (7)	4.79 (11)	6.57 (6)
MLSHSPA	7.36 (9)	7.57 (2)	5.96 (9)	7.43 (8)	6.57 (4)	7.07 (5)	5.64 (9)	7.29 (10)
LSHSPACMA	6.14 (5)	5.07 (11)	7.61 (2)	6.79 (6)	5.29 (11)	7.43 (3)	6.21 (7)	9.43 (12)

PSOFA, LSHSPACMA, and MLSHSPA. The DE and MFO also recorded the worst results and were ranked in the last order in all datasets.

Regarding the HA measure, the BMSCD was got the first rank in 50% of all datasets; the average of this measure was higher than other measures. in this regard, the BMSCD obtained 900 followed by BMO with 921 whereas, the worst one was MFO it reached 1412.

In WTER measure, the proposed BMSCD method obtained the smallest value (where the smallest value is the best) in 43% of the datasets and achieved competitive results in the rest of the datasets. The PSO was ranked second, and the DE was ranked third followed by the PSOFA, SCA,BMO, and SSA. LSHSPACMA and MFO did not show any best results in this measure.

Moreover, in terms of AR measure the BMSCD recorded the best results in 57% of the datasets and obtained average equals 0.528 of all datasets followed by BMO with average equals 0.473. Whereas, the rest of the algorithms yielded close averages of this measure in all datasets to some extent. The MLSHSPA and GWO showed the same performance in most of the experiments and were ranked third and fourth, respectively. In this measure, the worst averages were recorded by WOA and SSA.

According to the Hubert index measure, the proposed method reached the best values in 71% of all experiments followed by BMO. Whereas four algorithms showed competitive performance namely SCA, GWO, WOA, and PSOFA they obtained the best Hubert value in one dataset for each one. Furthermore, the results of the NMI measure showed an evident superiority over other algorithms. It showed the best results in 64% of the datasets followed by MFO and PSO in 21% and 14% of the datasets. The MLSHSPA and LSHSPACMA got the best results in one dataset for each one. The worst algorithm in this measure was SSA.

Fig. 3 summarizes all measures' results of all methods overall datasets. In this figure, the short bar indicates the best values in DB, HA, and WTER measures, whereas, in the rest of the measures, the long bar is the best. This figure shows the superiority of the BMSCD in all measures, especially in SI, DI, AR, and DB.

In general, the proposed BMSCD method obtained a superior output than other methods in all measures overall datasets, whereas, the MFO showed the worst results in 4 out of 8 measures.

Fig. 4 illustrates the achievements of each method based on the eight measures. In this figure, each measure appears in one of the shape's corners while the lines represent the method. The line will close to the measure's (shape) corner if its method achieved the best value in this measure. For instance, the BMSCD reached all corners because it was ranked first in all measures; in contrast, the MFO could not reach any corner because it was ranked last.

Besides, Fig. 5 shows an apparent comparison between the compared methods based on their performances in each measure. The *y*-axis represents the performance ratio it was scaled from 0% to 100% (e.g. 100% means the algorithm reached the best values in 14 datasets whereas, 0% means the algorithm did not obtain any best value in all datasets). For instance, the bar of the BMSCD reached 36% of SI measure because it obtains the best SI value in 36% of the datasets while GWO obtained 0% in the same measure.

Table A.1
Definition of the optimization benchmark functions.

ID	Formula of function	LW	UW	dim_W	Type
F1	$f(x)=\sum_{i=1}^{n} x_i^2$	−100	100	30	Unimodal
F2	$f(x)=\sum_{i=1}^{n} \lvert x_i\rvert + \Pi_{i=1}^{n}\lvert x_i\rvert$	−10	10	30	Unimodal
F3	$f(x)=\sum_{i=1}^{n}(\sum_{j-1}^{i} x_i)^2$	−100	100	30	Unimodal
F4	$f(x)=max_i\{\lvert x_i\rvert, 1\leqslant i\leqslant n\}$	−100	100	30	Unimodal
F5	$f(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	−30	30	30	Unimodal
F6	$f(x)=\sum_{i=1}^{n}([x_i+0.5])^2$	−100	100	30	Unimodal
F7	$f(x)=\sum_{i=1}^{n} ix_i^4+random[0,1]$	−1.28	1.28	30	Unimodal
F8	$f(x)=\sum_{i=1}^{n} ix_i^2$	−10	10	30	Unimodal
F9	$f(x)=\sum_{i=1}^{n} ix_i^4$	−1.28	1.28	30	Unimodal
F10	$f(x)=\sum_{i=1}^{n} \lvert x_i\rvert^{i+1}$	−1	1	30	Unimodal
F11	$f(x)=\sum_{i=1}^{n} -x_i sin(\sqrt{\lvert x_i\rvert})$	−500	500	30	Multimodal
F12	$f(x)=\sum_{i=1}^{n}[x_i^2-10cos(2\pi x_i)+10]$	−5.12	5.12	30	Multimodal
F13	$f(x)=-20exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2})-exp(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i))+20+e$	−32	32	30	Multimodal
F14	$f(x)=\frac{1}{4000}\sum_{i=1}^{n} x_i^2-\Pi_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}})+1$	−600	600	30	Multimodal
F15	$f(x)=\frac{\pi}{n}\{10sin^2(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2[1+10sin^2(\pi y_{i+1})]+(y_n-1)^2\}+\sum_{i=1}^{n} u(x_i,10,100,4)$ $u(x_i,a,k,m)=\begin{cases} k(x_i-a)^m, & x_i>a \\ 0, & -a\leqslant x_i\leqslant a \\ k(-x_i-a)^m, & x_i<-a \end{cases}$	−50	50	30	Multimodal
F16	$f(x)=0.1\{sin^2(3\pi x_1)+\sum_{i=1}^{n}(x_i-1)^2[1+sin^2(3\pi x_i+1)]+(x_n-1)^2[1+sin^2(2\pi x_n)]\}+\sum_{i=1}^{n} u(x_i,5,100,4)$	−50	50	30	Multimodal
F17	$f(x)=\sum_{i=1}^{n}(x_i-1)^2+[1+sin^2(3\pi x_i+1)]+sin^2(3\pi x_1)+\lvert x_n-1\rvert[1+sin^2(3\pi x_n)]$	−10	10	30	Multimodal
F18	$f(x)=\sum_{i=1}^{n} \lvert x_i sin(x_i)+0.1x_i\rvert$	−10	10	30	Multimodal
F19	$f(x)=0.1n-(0.1\sum_{i=1}^{n} cos(5\pi x_i))-\sum_{i=1}^{n} x_i^2$	−1	1	30	Multimodal
F20	$f(x)=\sum_{i=1}^{n} x_i^2+(\sum_{i=1}^{n} 0.5ix_i)^2+(\sum_{i=1}^{n} 0.5ix_i)^4$	−5	10	30	Multimodal
F21	$f(x)=\sum_{i=1}^{n} 0.5+\frac{sin^2(\sqrt{100x_{i-1}^2}+x_i^2-0.5)}{1+.001(x_i^2-2x_{i-1}x_i+x_i^2)^2}$	−5	10	30	Multimodal
F22	$f(x)=0.1sin^2(3\pi x_1)+\sum_{i=1}^{n-1}(x_i-1)^2(1+sin^2(3\pi x_{i+1})+(x_n-1)^2(1+sin^2(3\pi x_n)))$	−5	5	30	Multimodal
F23	$f(x)=\sum_{i=1}^{n}(10^6)^{(i-1)/(n-1)}x_i^2$	−100	100	30	Multimodal
F24	$f(x)=(-1)^{n+1}\Pi_{i=1}^{n}(cos(x_i))exp(\sum_{i=1}^{n}(x_i-\pi)^2)$	−100	100	30	Multimodal
F25	$f(x)=1-cos(2\pi\sqrt{\sum_{i=1}^{n} x_i^2})+0.1\sqrt{\sum_{i=1}^{n} x_i^2}$	−100	100	30	Multimodal
F26	$f(x)=0.5+\frac{sin^2(\sqrt{\sum_{i=1}^{n} x_i^2})-0.5}{(1+0.001(\sum_{i=1}^{n} x_i^2))^2}$	−100	100	30	Multimodal

(*continued on next page*)

Table A.1 (*continued*).

ID	Formula of function	LW	UW	dim_W	Type
F27	$f(x) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2}(x_i - a_{ij})^6})^{-1}$	−65.536	65.536	2	Multimodal
F28	$f(x) = (\sum_{i=1}^{11} [a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2)$	−5	5	4	Multimodal
F29	$f(x) = (4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - x_2^2 + 4x_2^4)$	−5	5	2	Multimodal
F30	$f(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})cosx_1 + 10$	−5	5	2	Multimodal
F31	$f(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$	−2	2	2	Multimodal
	$\times[30 + (2x_1 - 3x_2) \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$				
F32	$f(x) = -\sum_{i=1}^{4} c_i exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	1	3	3	Multimodal
F33	$f(x) = -\sum_{i=1}^{4} c_i exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	0	1	6	Multimodal
F34	$f(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	0	10	4	Multimodal
F35	$f(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	0	10	4	Multimodal
F36	$f(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	0	10	4	Multimodal

Furthermore, Table 13 records the results of the Friedman test as a kind of non-parametric statistical tests. It was used in this paper to rank the algorithms to explore the effectiveness and significance of the BMSCD method. In Table 13, the Friedman values and the corresponding ranks are listed for each algorithm overall the measures. The BMSCD outperformed all algorithms in this test it was ranked first followed by the PSO, which was ranked second in three measures, namely DB, Ha, and NMI. The SCA was showed good rank and obtained the second rank in SI and WTER measures followed by the BMO, LSHSPACMA, GWO, PSOFA, and WOA, respectively. The MFO algorithm achieved the last rank.

From these results, it can be concluded that BMSCD is an effective algorithm to solve clustering problems.

From the previous results, it can be seen the superiority of the proposed BMSCD. In most of the quality measures, among the compared algorithms. Since BMSCD combined the strength of BMO, SCA, and disruption operators, this gives a suitable tool to explore the search space efficiently simultaneously with avoiding the problem of stagnation to the local point.

7. Conclusions and future directions

In this work, we utilized the components of SCA with a disruption operator to enhance the primary efficacy of BMO in terms of convergence trends and stagnation avoidance. The developed method is able to switch between strategies of SCA and BMO based on a suitable probability function. Also, it utilizes a DO process to enhance the diversity of the swarm further. The proposed optimizer was used to realize continuous problems and clustering tasks with various problems, including 36 unimodal and multimodal cases and 13 clustering datasets. Experimental results verified that the developed BMSCD method could efficiently tackle the studied problems and superior performance in most cases. The results expose the improved efficacy of the developed BMSCD in terms of the excellency of solutions, the acceptable trade-off between exploration and exploitation, and improved convergence rates. However, this method has no exceptional ability in case of stagnation avoidance, and it may find local optima in dealing with some classes of problems. Another shortcoming is the slow convergence of the method in facing larger-scale problems, which can happen occasionally.

For future works, we will propose the multi-objective variant of BMSCD. Another helpful direction is to develop a binary BMSCD and analysis the impact of different transfer functions on the results. This method has no chaos-based component. Hence, analyzing such a proposal for disruption operation can be another direction. It is also interesting to evaluate the efficacy of this method in dealing with more variety of tasks such as feature selection and machine learning area.

CRediT authorship contribution statement

Mohamed Abd Elaziz: Conceptualization, Methodology, Software, Data curation, Validation, Investigation, Writing – original draft. **Ahmed A. Ewees:** Conceptualization, Methodology, Software, Investigation, Formal analysis, Visualization, Writing – original draft, Validation. **Mohammed A.A. Al-qaness:** Validation, Investigation, Visualization, project administration, funding acquisition, Writing – review & editing, Writing – original draft. **Laith Abualigah:** Validation, Visualization, Writing – review & editing, Writing – original draft. **Rehab Ali Ibrahim:** Data curation, Formal analysis, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant No. 62150410434) and in part by LIESMARS Special Research Funding.

Appendix. Definition of benchmark functions

See Table A.1

References

Abd Elaziz, M., Ewees, A. A., Neggaz, N., Ibrahim, R. A., Al-qaness, M. A., & Lu, S. (2021a). Cooperative meta-heuristic algorithms for global optimization problems. *Expert Systems with Applications*, *176*, Article 114788.

Abd Elaziz, M., Heidari, A. A., Fujita, H., & Moayedi, H. (2020). A competitive chain-based harris hawks optimizer for global optimization and multi-level image thresholding problems. *Applied Soft Computing*, Article 106347.

Abd Elaziz, M., Yousri, D., Al-qaness, M. A., AbdelAty, A. M., Radwan, A. G., & Ewees, A. A. (2021b). A grunwald–letnikov based manta ray foraging optimizer for global optimization and image segmentation. *Engineering Applications of Artificial Intelligence*, *98*, Article 104105.

Abualigah, L. M. Q. (2019). *Feature selection and enhanced krill herd algorithm for text document clustering*. Springer.

Abualigah, L. (2020a). Group search optimizer: A nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications. *Neural Computing and Applications*, 1–24.

Abualigah, L. (2020b). Multi-verse optimizer algorithm: a comprehensive survey of its results, variants, and applications. *Neural Computing and Applications*, 1–21.

Abualigah, L., Abd Elaziz, M., Sumari, P., Geem, Z. W., & Gandomi, A. H. (2022a). Reptile search algorithm (rsa): A nature-inspired meta-heuristic optimizer. *Expert Systems with Applications*, *191*, Article 116158.

Abualigah, L., Almotairi, K. H., Abd Elaziz, M., Shehab, M., & Altalhi, M. (2022b). Enhanced flow direction arithmetic optimization algorithm for mathematical optimization problems with applications of data clustering. *Engineering Analysis with Boundary Elements*, *138*, 13–29.

Abualigah, L., & Diabat, A. (2020). A comprehensive survey of the grasshopper optimization algorithm: results, variants, and applications. *Neural Computing and Applications*, 1–24.

Abualigah, L., & Diabat, A. (2021). Advances in sine cosine algorithm: a comprehensive survey. *Artificial Intelligence Review*, 1–42.

Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2020). The arithmetic optimization algorithm. *Computer Methods in Applied Mechanics and Engineering*.

Abualigah, L., Ewees, A. A., Al-qaness, M. A., Elaziz, M. A., Yousri, D., Ibrahim, R. A., et al. (2022c). Boosting arithmetic optimization algorithm by sine cosine algorithm and levy flight distribution for solving engineering optimization problems. *Neural Computing and Applications*, 1–30.

Abualigah, L. M., Khader, A. T., & Hanandeh, E. S. (2018). Hybrid clustering analysis using improved krill herd algorithm. *Applied Intelligence*, 1–25.

Abualigah, L. M., Khader, A. T., & Hanandeh, E. S. (2019). Modified krill herd algorithm for global numerical optimization problems. In *Advances in nature-inspired computing and applications* (pp. 205–221). Springer.

Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-qanes, M., & Gandomi, A. H. (2021). Aquila optimizer: A novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering*, http://dx.doi.org/10.1016/j.cie.

Agushaka, J. O., Ezugwu, A. E., & Abualigah, L. (2022). Dwarf mongoose optimization algorithm. *Computer Methods in Applied Mechanics and Engineering*, *391*, Article 114570.

Ahmed, S., Ghosh, K. K., Bera, S. K., Schwenker, F., & Sarkar, R. (2020). Gray level image contrast enhancement using barnacles mating optimizer. *IEEE Access*, *8*, 169196–169214.

Al-Qaness, M. A., Elaziz, M. A., & Ewees, A. A. (2018). Oil consumption forecasting using optimized adaptive neuro-fuzzy inference system based on sine cosine algorithm. *IEEE Access*, *6*, 68394–68402.

Al-qaness, M. A., Ewees, A. A., Fan, H., Abualigah, L., & Abd Elaziz, M. (2022). Boosted anfis model using augmented marine predator algorithm with mutation operators for wind power forecasting. *Applied Energy*, *314*, Article 118851.

Aljarah, I., Mafarja, M., Heidari, A. A., Faris, H., & Mirjalili, S. (2020). Clustering analysis using a novel locality-informed grey wolf-inspired clustering approach. *Knowledge and Information Systems*, *62*, 507–539.

Almotairi, K. H., & Abualigah, L. (2022). Hybrid reptile search algorithm and remora optimization algorithm for optimization tasks and data clustering. *Symmetry*, *14*(458).

Askari, Q., Younas, I., & Saeed, M. (2020). Political optimizer: A novel socio-inspired meta-heuristic for global optimization. *Knowledge-Based Systems*, Article 105709.

Attia, A.-F., El Sehiemy, R. A., & Hasanien, H. M. (2018). Optimal power flow solution in power systems using a novel sine-cosine algorithm. *International Journal of Electrical Power & Energy Systems*, *99*, 331–343.

Aydilek, I. B. (2018). A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems. *Applied Soft Computing*, *66*, 232–249.

Barazandeh, M., Davis, C. S., Neufeld, C. J., Coltman, D. W., & Palmer, A. R. (2013). Something darwin didn't know about barnacles: spermcast mating in a common stalked species. *Proceedings of the Royal Society B: Biological Sciences*, *280*, Article 20122919.

Bojan-Dragos, C.-A., Precup, R.-E., Preitl, S., Roman, R.-C., Hedrea, E.-L., & Szedlak-Stinean, A.-I. (2021). Gwo-based optimal tuning of type-1 and type-2 fuzzy controllers for electromagnetic actuated clutch systems. *IFAC-PapersOnLine*, *54*, 189–194.

Cao, B., Dong, W., Lv, Z., Gu, Y., Singh, S., & Kumar, P. (2020a). Hybrid microgrid many-objective sizing optimization with fuzzy decision. *IEEE Transactions on Fuzzy Systems*.

Cao, B., Fan, S., Zhao, J., Yang, P., Muhammad, K., & Tanveer, M. (2020b). Quantum-enhanced multiobjective large-scale optimization via parallelism. *Swarm and Evolutionary Computation*, *57*, Article 100697. http://dx.doi.org/10.1016/j.swevo.2020.100697, http://www.sciencedirect.com/science/article/pii/S2210650219307990.

Cao, B., Wang, X., Zhang, W., Song, H., & Lv, Z. (2020c). A many-objective optimization model of industrial internet of things based on private blockchain. *IEEE Network*, *34*, 78–83.

Cao, B., Zhao, J., Yang, P., Gu, Y., Muhammad, K., Rodrigues, J. J., et al. (2019). Multiobjective 3-d topology optimization of next-generation wireless data center network. *IEEE Transactions on Industrial Informatics*, *16*, 3597–3605.

Chen, H., Heidari, A. A., Zhao, X., Zhang, L., & Chen, H. (2020). Advanced orthogonal learning-driven multi-swarm sine cosine optimization: Framework and case studies. *Expert Systems with Applications*, *144*, Article 113113.

Chen, H., Jiao, S., Heidari, A. A., Wang, M., Chen, X., & Zhao, X. (2019a). An opposition-based sine cosine approach with local search for parameter estimation of photovoltaic models. *Energy Conversion and Management*, *195*, 927–942.

Chen, H., Qiao, H., Xu, L., Feng, Q., & Cai, K. (2019b). A fuzzy optimization strategy for the implementation of rbf lssvr model in vis–nir analysis of pomelo maturity. *IEEE Transactions on Industrial Informatics*, *15*, 5971–5979.

Chen, H.-L., Wang, G., Ma, C., Cai, Z.-N., Liu, W.-B., & Wang, S.-J. (2016). An efficient hybrid kernel extreme learning machine approach for early diagnosis of parkinson disease. *Neurocomputing*, *184*, 131–144.

Chou, C.-H., Su, M.-C., & Lai, E. (2004). A new cluster validity measure and its application to image compression. *Pattern Analysis and Applications*, *7*, 205–220.

Das, S., Abraham, A., & Konar, A. (2007). Automatic clustering using an improved differential evolution algorithm. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *38*, 218–237.

Das, S., Abraham, A., & Konar, A. (2008). Automatic kernel clustering with a multi-elitist particle swarm optimization algorithm. *Pattern Recognition Letters*, *29*, 688–699.

Das, S., & Konar, A. (2009). Automatic image pixel clustering with an improved differential evolution. *Applied Soft Computing*, *9*, 226–236.

Das, S., & Sil, S. (2010). Kernel-induced fuzzy clustering of image pixels with an improved differential evolution algorithm. *Information Sciences*, *180*, 1237–1256.

Ding, G. Y., Liu, H., & He, X. Q. (2013). A novel disruption operator in particle swarm optimization. In *Applied mechanics and materials, Vol. 380* (pp. 1216–1220).

Elaziz, M. A., Nabil, N., Ewees, A. A., & Lu, S. (2019). Automatic data clustering based on hybrid atom search optimization and sine-cosine algorithm. In *2019 IEEE congress on evolutionary computation* (pp. 2315–2322). IEEE.

Ewees, A. A., Elaziz, M. A., & Houssein, E. H. (2018). Improved grasshopper optimization algorithm using opposition-based learning. *Expert Systems with Applications*, *112*, 156–172.

Ezugwu, A. E., Ikotun, A. M., Oyelade, O. O., Abualigah, L., Agushaka, J. O., Eke, C. I., et al. (2022). A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, *110*, Article 104743.

Fan, Y., Wang, P., Heidari, A. A., Wang, M., Zhao, X., Chen, H., et al. (2020). Rationalized fruit fly optimization with sine cosine algorithm: A comprehensive analysis. *Expert Systems with Applications*, Article 113486.

Faris, H., Mirjalili, S., Aljarah, I., Mafarja, M., & Heidari, A. A. (2020). Salp swarm algorithm: theory, literature review, and application in extreme learning machines. In *Nature-inspired optimizers* (pp. 185–199). Springer.

Fathy, A., Elaziz, M. A., Sayed, E. T., Olabi, A., & Rezk, H. (2019). Optimal parameter identification of triple-junction photovoltaic panel based on enhanced moth search algorithm. *Energy*, *188*, Article 116025.

Fu, X., Pace, P., Aloi, G., Yang, L., & Fortino, G. (2020). Topology optimization against cascading failures on wireless sensor networks using a memetic algorithm. *Computer Networks*, Article 107327.

Gouthamkumar, N., Sharma, V., & Naresh, R. (2015). Disruption based gravitational search algorithm for short term hydrothermal scheduling. *Expert Systems with Applications*, *42*, 7000–7011.

Gupta, S., Deep, K., & Engelbrecht, A. P. (2020). A memory guided sine cosine algorithm for global optimization. *Engineering Applications of Artificial Intelligence*, *93*, Article 103718.

Hadi, A. A., Mohamed, A. W., & Jambi, K. M. (2019). Lshade-spa memetic framework for solving large-scale optimization problems. *Complex & Intelligent Systems*, *5*, 25–40.

Houssein, E. H., AbdElminaam, D. S., Hassan, H. N., Al-Sayed, M. M., & Nabil, E. (2021). A hybrid barnacles mating optimizer algorithm with support vector machines for gene selection of microarray cancer classification. *IEEE Access*.

Hu, L., Hong, G., Ma, J., Wang, X., & Chen, H. (2015). An efficient machine learning approach for diagnosis of paraquat-poisoned patients. *Computers in Biology and Medicine*, *59*, 116–124.

Huang, H., Heidari, A. A., Xu, Y., Wang, M., Liang, G., Chen, H., et al. (2020). Rationalized sine cosine optimization with efficient searching patterns. *IEEE Access*, *8*, 61471–61490.

Ibrahim, R. A., Abd Elaziz, M., Ewees, A. A., Selim, I. M., & Lu, S. (2018a). Galaxy images classification using hybrid brain storm optimization with moth flame optimization. *Journal of Astronomical Telescopes, Instruments, and Systems*, *4*, Article 038001.

Ibrahim, R. A., Elaziz, M. A., & Lu, S. (2018b). Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization. *Expert Systems with Applications*, *108*, 1–27.

Ibrahim, R. A., Ewees, A. A., Oliva, D., Elaziz, M. A., & Lu, S. (2019). Improved salp swarm algorithm based on particle swarm optimization for feature selection. *Journal of Ambient Intelligence and Humanized Computing*, *10*, 3155–3169.

Ibrahim, R. A., Oliva, D., Ewees, A. A., & Lu, S. (2017). Feature selection based on improved runner-root algorithm using chaotic singer map and opposition-based learning. In *International conference on neural information processing* (pp. 156–166).

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys (CSUR)*, *31*, 264–323.

Jiang, J., Han, R., Meng, X., & Li, K. (2020). Tsasc: tree–seed algorithm with sine–cosine enhancement for continuous optimization problems. *Soft Computing*, 1–20.

Jouhari, H., Lei, D., AA Al-qaness, M., Abd Elaziz, M., Ewees, A. A., & Farouk, O. (2019). Sine-cosine algorithm to enhance simulated annealing for unrelated parallel machine scheduling with setup times. *Mathematics*, *7*(1120).

Karaboga, D., & Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation*, *214*, 108–132.

Kaur, S., Awasthi, L. K., Sangal, A., & Dhiman, G. (2020). Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Engineering Applications of Artificial Intelligence*, *90*, Article 103541.
Kaur, A., Sharma, H., & Sharma, N. (2017). Disruption operator-based spider monkey optimization algorithm. In *2017 International conference on computing, communication and automation* (pp. 216–221).
Khalilpourazari, S., & Khalilpourazary, S. (2019). An efficient hybrid algorithm based on water cycle and moth-flame optimization algorithms for solving numerical and constrained engineering optimization problems. *Soft Computing*, *23*, 1699–1722.
Kuo, R., Amornnikun, P., & Nguyen, T. P. Q. (2020). Metaheuristic-based possibilistic multivariate fuzzy weighted c-means algorithms for market segmentation. *Applied Soft Computing*, *96*, Article 106639.
Kurada, R. R., Pavan, K. K., & Rao, A. A. (2015). Automatic teaching–learning-based optimization: A novel clustering method for gene functional enrichments. In *Computational intelligence techniques for comparative genomics* (pp. 17–35). Springer.
Li, C., Hou, L., Sharma, B. Y., Li, H., Chen, C., Li, Y., et al. (2018). Developing a new intelligent system for the diagnosis of tuberculous pleural effusion. *Computer Methods and Programs in Biomedicine*, *153*, 211–225.
Lin, A., Wu, Q., Heidari, A. A., Xu, Y., Chen, H., Geng, W., et al. (2019). Predicting intentions of students for master programs using a chaos-induced sine cosine-based fuzzy k-nearest neighbor classifier. *Ieee Access*, *7*, 67235–67248.
Liu, H., Ding, G., & Sun, H. (2012). An improved opposition-based disruption operator in gravitational search algorithm. In *2012 Fifth international symposium on computational intelligence and design, Vol. 2* (pp. 123–126).
Liu, H., Ding, G., & Wang, B. (2014). Bare-bones particle swarm optimization with disruption operator. *Appl. Math. Comput.*, *238*, 106–122.
Liu, E., Lv, L., Yi, Y., & Xie, P. (2019). Research on the steady operation optimization model of natural gas pipeline considering the combined operation of air coolers and compressors. *IEEE Access*, *7*, 83251–83265.
Liu, D., Wang, S., Huang, D., Deng, G., Zeng, F., & Chen, H. (2016). Medical image classification using spatial adjacent histogram based on adaptive local binary patterns. *Computers in Biology and Medicine*, *72*, 185–200.
Liu, J., Wu, C., Wu, G., & Wang, X. (2015). A novel differential search algorithm and applications for structure design. *Applied Mathematics and Computation*, *268*, 246–269.
Mahdad, B., & Srairi, K. (2018). A new interactive sine cosine algorithm for loading margin stability improvement under contingency. *Electrical Engineering*, *100*, 913–933.
Mirjalili, S. (2016). Sca: a sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, *96*, 120–133.
Mirjalili, S., Aljarah, I., Mafarja, M., Heidari, A. A., & Faris, H. (2020). Grey wolf optimizer: theory, literature review, and application in computational fluid dynamics problems. In *Nature-inspired optimizers* (pp. 87–105). Springer.
Mirjalili, S., Dong, J. S., & Lewis, A. (2019). *Nature-inspired optimizers: theories, literature reviews and applications, Vol. 811*. Springer.
Mohamed, A. W., Hadi, A. A., Fattouh, A. M., & Jambi, K. M. (2017). Lshade with semi-parameter adaptation hybrid with cma-es for solving cec 2017 benchmark problems. In *2017 IEEE congress on evolutionary computation* (pp. 145–152).
Mousavirad, S. J., Ebrahimpour-Komleh, H., & Schaefer, G. (2020). Automatic clustering using a local search-based human mental search algorithm for image segmentation. *Applied Soft Computing*, *96*, Article 106604.
Naji Alwerfali, H. S., Al-qaness, M. A., Abd Elaziz, M., Ewees, A. A., Oliva, D., & Lu, S. (2020). Multi-level image thresholding based on modified spherical search optimizer and fuzzy entropy. *Entropy*, *22*(328).
Neggaz, N., Ewees, A. A., Abd Elaziz, M., & Mafarja, M. (2020). Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection. *Expert Systems with Applications*, *145*, Article 113103.
Osman, I. H., & Laporte, G. (1996). Metaheuristics: A bibliography.
Oyelade, O. N., Ezugwu, A. E.-S., Mohamed, T. I., & Abualigah, L. (2022). Ebola optimization search algorithm: A new nature-inspired metaheuristic optimization algorithm. *IEEE Access*, *10*, 16150–16177.
Pashiri, R. T., Rostami, Y., & Mahrami, M. (2020). Spam detection through feature selection using artificial neural network and sine–cosine algorithm. *Mathematical Sciences*.
Precup, R.-E., David, R.-C., Roman, R.-C., Szedlak-Stinean, A.-I., & Petriu, E. M. (2021). Optimal tuning of interval type-2 fuzzy controllers for nonlinear servo systems using slime mould algorithm. *International Journal of Systems Science*, 1–16.
Precup, R.-E., Hedrea, E.-L., Roman, R.-C., Petriu, E. M., Szedlak-Stinean, A.-I., & Bojan-Dragos, C.-A. (2020). Experiment-based approach to teach optimization techniques. *IEEE Transactions on Education*, *64*, 88–94.
Qu, S., Han, Y., Wu, Z., & Raza, H. (2020). Consensus modeling with asymmetric cost based on data-driven robust optimization. *Group Decision and Negotiation*, 1–38.
Rao, R. V., Savsani, V. J., & Vakharia, D. (2012). Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. *Information Sciences*, *183*, 1–15.
Rigatos, G., Siano, P., Selisteanu, D., & Precup, R. (2017). Nonlinear optimal control of oxygen and carbon dioxide levels in blood. *Intelligent Industrial Systems*, *3*, 61–75.
Sarafrazi, S., Nezamabadi-Pour, H., & Saryazdi, S. (2011). Disruption: a new operator in gravitational search algorithm. *Scientia Iranica*, *18*, 539–548.
Sayed, G. I., Khoriba, G., & Haggag, M. H. (2018). A novel chaotic salp swarm algorithm for global optimization and feature selection. *Applied Intelligence*, *48*, 3462–3481.
Seyedali, M. (2016). Sca: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, *96*, 120–133.
Sharma, N., Sharma, H., Sharma, A., & Bansal, J. C. (2016). Modified artificial bee colony algorithm based on disruption operator. In *Proceedings of fifth international conference on soft computing for problem solving* (pp. 889–900).
Shen, L., Chen, H., Yu, Z., Kang, W., Zhang, B., Li, H., et al. (2016). Evolving support vector machines using fruit fly optimization for medical data classification. *Knowledge-Based Systems*, *96*, 61–75.
Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y.-P., Auger, A., et al. (2005). *Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization* (p. 2005). Singapore: Nanyang Technological University.
Sulaiman, M. H., & Mustaffa, Z. (2021). Solving optimal power flow problem with stochastic wind–solar–small hydro power using barnacles mating optimizer. *Control Engineering Practice*, *106*, Article 104672.
Sulaiman, M. H., Mustaffa, Z., & Aliman, O. (2019a.). An application of barnacles mating optimizer for solving economic dispatch problems. In *2019 IEEE Jordan international joint conference on electrical engineering and information technology* (pp. 835–839). IEEE.
Sulaiman, M. H., Mustaffa, Z., Saari, M. M., & Daniyal, H. (2020). Barnacles mating optimizer: a new bio-inspired algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, *87*, Article 103330.
Sulaiman, M. H., Mustaffa, Z., Saari, M. M., Daniyal, H., Mohamad, A. J., Othman, M. R., et al. (2019b). Barnacles mating optimizer algorithm for optimization. In *Proceedings of the 10th national technical seminar on underwater system technology 2018* (pp. 211–218). Springer.
Sulaiman, M. H., Mustaffa, Z., Saari, M. M., Daniyal, H., Musirin, I., & Daud, M. R. (2018). Barnacles mating optimizer: An evolutionary algorithm for solving optimization. In *2018 IEEE international conference on automatic control and intelligent systems* (pp. 99–104). IEEE.
Sun, G., Yang, B., Yang, Z., & Xu, G. (2019). An adaptive differential evolution with combined strategy for global numerical optimization. *Soft Computing*, 1–20.
Tan, G. W.-H., Ooi, K.-B., Leong, L.-Y., & Lin, B. (2014). Predicting the drivers of behavioral intention to use mobile learning: A hybrid sem-neural networks approach. *Computers in Human Behavior*, *36*, 198–213.
Tawhid, M. A., & Savsani, V. (2019). Multi-objective sine-cosine algorithm (mo-sca) for multi-objective engineering design problems. *Neural Computing and Applications*, *31*, 915–929.
Valdivia, S., Soto, R., Crawford, B., Caselli, N., Paredes, F., Castro, C., et al. (2020). Clustering-based binarization methods applied to the crow search algorithm for 0/1 combinatorial problems. *Mathematics*, *8*(1070).
Wan, Y., Ma, A., Zhong, Y., Hu, X., & Zhang, L. (2020). Multiobjective hyperspectral feature selection based on discrete sine cosine algorithm. *IEEE Transactions on Geoscience and Remote Sensing*, *58*, 3601–3618.
Wang, M., & Chen, H. (2020). Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis. *Applied Soft Computing Journal*, *88*, http://dx.doi.org/10.1016/j.asoc.2019.105946, https://www.scopus.com/inward/record.uri?eid=2-s2.0-85076535070&doi=10.1016%2fj.asoc.2019.105946&partnerID=40&md5=8a1c5e7662f93045d36cfacc01b68af2.
Wang, S.-J., Chen, H.-L., Yan, W.-J., Chen, Y.-H., & Fu, X. (2014a). Face recognition and micro-expression recognition based on discriminant tensor subspace analysis plus extreme learning machine. *Neural Processing Letters*, *39*, 25–43.
Wang, M., Chen, H., Yang, B., Zhao, X., Hu, L., Cai, Z., et al. (2017). Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses. *Neurocomputing*, *267*, 69–84.
Wang, Y., Li, H.-X., Huang, T., & Li, L. (2014b). Differential evolution based on covariance matrix learning and bimodal distribution parameter setting. *Applied Soft Computing*, *18*, 232–247.
Xia, J., Chen, H., Li, Q., Zhou, M., Chen, L., Cai, Z., et al. (2017). Ultrasound-based differentiation of malignant and benign thyroid nodules: An extreme learning machine approach. *Computer Methods and Programs in Biomedicine*, *147*, 37–49.
Xu, X., & Chen, H.-L. (2014). Adaptive computational chemotaxis based on field in bacterial foraging optimization. *Soft Computing*, *18*, 797–807.
Xu, Y., Chen, H., Heidari, A. A., Luo, J., Zhang, Q., Zhao, X., et al. (2019a). An efficient chaotic mutative moth-flame-inspired optimizer for global optimization tasks. *Expert Systems with Applications*, *129*, 135–155.
Xu, Y., Chen, H., Luo, J., Zhang, Q., Jiao, S., & Zhang, X. (2019b). Enhanced moth-flame optimizer with mutation strategy for global optimization. *Information Sciences*, *492*, 181–203.
Yang, Z., Liu, Q., Zhang, L., Dai, J., & Razmjooy, N. (2020). Model parameter estimation of the pemfcs using improved barnacles mating optimization algorithm. *Energy*, *212*, Article 118738.
Yousri, D., AbdelAty, A. M., Al-qaness, M. A., Ewees, A. A., Radwan, A. G., & Abd Elaziz, M. (2022). Discrete fractional-order caputo method to overcome trapping in local optima: Manta ray foraging optimizer as a case study. *Expert Systems with Applications*, *192*, Article 116355.
Yusa, Y., Yoshikawa, M., Kitaura, J., Kawane, M., Ozaki, Y., Yamato, S., et al. (2012). Adaptive evolution of sexual systems in pedunculate barnacles. *Proceedings of the Royal Society B: Biological Sciences*, *279*, 959–966.

Zhang, X., Fan, M., Wang, D., Zhou, P., & Tao, D. (2020a). Top-k feature selection framework using robust 0-1 integer programming. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15. http://dx.doi.org/10.1109/TNNLS.2020.3009209.

Zhang, X., Jiang, R., Wang, T., & Wang, J. (2020b). Recursive neural network for video deblurring. *IEEE Transactions on Circuits and Systems for Video Technology*, 1. http://dx.doi.org/10.1109/TCSVT.2020.3035722.

Zhang, Y., & Jin, Z. (2020). Group teaching optimization algorithm: A novel metaheuristic method for solving global optimization problems. *Expert Systems with Applications*, *148*, Article 113246.

Zhang, R., Tian, F., Ren, X., Chen, Y., Chao, K., Zhao, R., et al. (2018). Associate multi-task scheduling algorithm based on self-adaptive inertia weight particle swarm optimization with disruption operator and chaos operator in cloud environment. *Service Oriented Computing and Applications*, *12*, 87–94.

Zhang, X., Wang, T., Wang, J., Tang, G., & Zhao, L. (2020c). Pyramid channel-based feature attention network for image dehazing. *Computer Vision and Image Understanding*, *197–198*, Article 103003. http://dx.doi.org/10.1016/j.cviu.2020.103003, http://www.sciencedirect.com/science/article/pii/S1077314220300709.

Zhang, X., Wang, D., Zhou, Z., & Ma, Y. (2019). Robust low-rank tensor recovery with rectification and alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1. http://dx.doi.org/10.1109/TPAMI.2019.2929043.

Zhao, X., Li, D., Yang, B., Chen, H., Yang, X., Yu, C., et al. (2015). A two-stage feature selection method with its application. *Computers and Electrical Engineering*, *47*, 114–125.

Zhao, X., Li, D., Yang, B., Ma, C., Zhu, Y., & Chen, H. (2014). Feature selection based on improved ant colony optimization for online detection of foreign fiber in cotton. *Applied Soft Computing*, *24*, 585–596.

Zhao, X., Zhang, X., Cai, Z., Tian, X., Wang, X., Huang, Y., et al. (2019). Chaos enhanced grey wolf optimization wrapped elm for diagnosis of paraquat-poisoned patients. *Computational Biology and Chemistry*, *78*, 481–490. http://dx.doi.org/10.1016/j.compbiolchem.2018.11.017, https://www.scopus.com/inward/record.uri?eid=2-s2.0-85057279588&doi=10.1016%2fj.compbiolchem.2018.11.017&partnerID=40&md5=27f45603ef6ef132a3657aad39c53dc5.

Zhou, W., Wang, P., Heidari, A. A., Wang, M., Zhao, X., & Chen, H. (2020). Multi-core sine cosine optimization: Methods and inclusive analysis. *Expert Systems with Applications*, Article 113974.

Zhu, W., Ma, C., Zhao, X., Wang, M., Heidari, A. A., Chen, H., et al. (2020). Evaluation of sino foreign cooperative education project using orthogonal sine cosine optimized kernel extreme learning machine. *IEEE Access*, *8*, 61107–61123.