Neural Computing and Applications (2023) 35:1493–1517
https://doi.org/10.1007/s00521-022-07788-z

ORIGINAL ARTICLE

Evaluating the performance of meta-heuristic algorithms on CEC 2021 benchmark problems

Ali Wagdy Mohamed[1,2] · Karam M. Sallam[3] · Prachi Agrawal[4] · Anas A. Hadi[5] · Ali Khater Mohamed[6]

Received: 13 January 2022 / Accepted: 6 September 2022 / Published online: 28 September 2022

Abstract

To develop new meta-heuristic algorithms and evaluate on the benchmark functions is the most challenging task. In this paper, performance of the various developed meta-heuristic algorithms are evaluated on the recently developed CEC 2021 benchmark functions. The objective functions are parametrized by inclusion of the operators, such as bias, shift and rotation. The different combinations of the binary operators are applied to the objective functions which leads to the CEC2021 benchmark functions. Therefore, different meta-heuristic algorithms are considered which solve the benchmark functions with different dimensions. The performance of some basic, advanced meta-heuristics algorithms and the algorithms that participated in the CEC2021 competition have been experimentally investigated and many observations, recommendations, conclusions have been reached. The experimental results show the performance of meta-heuristic algorithms on the different combinations of binary parameterized operators.

Keywords Benchmark functions · CEC2021 · Meta-heuristic algorithms · Parameterization

1 Introduction

Metaheuristic algorithms represent a class of derivative-free, nature inspired algorithms and provide robust optimization tools for problems. Mostly, the analytical process may fail for solving the complex, and real-world optimization problems. Therefore, meta-heuristic algorithms prove very efficient and effective algorithms in solving these types of problems. The applications of meta-heuristic algorithms are found in the numerous fields of science, machine learning, engineering, operations research [1–7].

The performance evaluation and comparison of algorithms are very much dependent on benchmarking. The benchmarking experiments are developed to predict/select the best algorithm for solving real-world problems [8]. In the past two decades, proposing new benchmark real-parameter single-objective optimization problems with several novel characteristics to evaluate and analyze the

✉ Ali Wagdy Mohamed
aliwagdy@staff.cu.edu.eg

Karam M. Sallam
karam.sallam@canberra.edu.au

Prachi Agrawal
Prachi.Agrawal@iit.comillas.edu

Anas A. Hadi
a.hadi@arabou.edu.sa

Ali Khater Mohamed
akhater@msa.edu.eg

[1] Operations Research Department, Faculty of Graduate Studies for Statistical Research, Cairo University, Giza 12613, Egypt

[2] Department of Mathematics and Actuarial Science School of Sciences and Engineering, The American University in Cairo, Cairo 11835, Egypt

[3] Faculty of Science and Technology, School of IT and Systems, University of Canberra, Canberra, ACT 2601, Australia

[4] Institute for Research in Technology, Comillas Pontificial University, Madrid 28015, Spain

[5] Faculty of Computer Studies, Arab Open University, P. O. Box 84901, Riyadh 11681, Saudi Arabia

[6] Faculty of Computer Science, October University for Modern Sciences and Arts (MSA), 6th October City, Giza 12451, Egypt

performance of meta-heuristic algorithms is considered as the cornerstone of research in the optimization field. Furthermore, it has been attracted by computer science, operations research practitioners and specialists as well as mathematicians and engineers. There are following important reasons for developing benchmark problems:

1. It is considered the basis for developing more complex optimization problems such as single-objective computationally expensive numerical optimization problems, single objective multi-niche optimization problems, constrained real-parameter single-objective optimization problems, constrained / bound-constrained multi/many-objective optimization problems.
2. It must simulate the degree of difficulty of the real-world optimization problems.
3. It must be able to detect the weakness and strengths of the novel optimization algorithms which have been significantly improved during the past few years.

Two benchmark series are common in evaluation of real-parameter evolutionary algorithms, namely IEEE Congress on Evolutionary Computation (CEC) competitions and the Comparing Continuous Optimizer (COCO). The COCO benchmark suite provides a platform to compare the performance of large number of algorithms for unconstrained continuous optimization problems. COCO benchmark suits specially represent the single objective noiseless and noise problems and bi-objective noiseless problems [9–11]. On the other hand, CEC benchmark represents the most elaborated platform for the comparison of stochastic search algorithms. The CEC benchmark suite includes single, multi-objective, noiseless, noise, large-scale, real-world problems, constrained optimization problems. Moreover, it also provides the performance metrices, test environment for assessment and comparison. To evaluate the performance of state-of-the-art algorithms, CEC competitions functions are most frequently used for benchmarking.

Since 2005, a new generation of benchmark problems have been developed to evaluate with the new era of nature-inspired algorithms or meta-heuristics. 2005 IEEE Congress on Evolutionary Computation was the inception of the first benchmark problems that overcome the aforementioned shortcoming features, named CEC'05 [12]. CEC'05 report included 25 benchmark functions with different properties such as separable, non-separable, rotated, unrotated, unimodal and multimodal functions with shifts in dimensionality, multiplicative noise in fitness and composition of functions, continuous functions, non-continuous functions, global optimum on the bounds, global optimum not on the bounds, function with no clear structure in the fitness landscape, the narrow global basin of attraction and so on.

Eight years later, CEC'13 test suite which includes 28 benchmark functions has been proposed [13]. In the CEC'13 test suite, the previously proposed composition functions [14] are improved and additional test functions are included. In the same context, CEC'14 test suite which includes 30 benchmark functions has been proposed [15]. CEC'14 developed benchmark problems with several novel features such as novel basic problems, composing test problems by extracting features dimension-wise from several problems, graded level of linkages, rotated trap problems, and so on. Three years later, CEC'17 test suite which includes 30 benchmark functions has been proposed [16]. In the CEC'17 test suite, similar to CE'14, new basic functions with different features have been added. These benchmark functions are discussed in detailed in the next section. The same benchmark suite of CEC'17 has been used in CEC'18, CEC'19 and CEC'20 [17]. As algorithms improve, ever more challenging functions are developed. This interplay between methods and problems drives progress, the CEC'20 [17] and CEC'21 [18] Special Sessions on Real-Parameter Optimization are developed to promote this symbiosis. In CEC'20 competition, the objective function is transformed into another function with the inclusion of the rotation matrix. Moreover, new benchmark objective functions are set in CEC'21 competition by including the different combination of bias, shift and rotation operators. These benchmark functions make a new challenge for researchers to develop meta-heuristic algorithms which handle all the complexity of the functions.

Additionally, it is pointed out that in order to compare and analyse the solution quality of different algorithms statistically and to verify the behaviour of stochastic algorithms [19], the results are compared using two non-parametric statistical hypothesis tests: I the Friedman test (to determine the final rankings of different algorithms for all functions) and (ii) the multi-problem Wilcoxon signed-rank test (to check the differences between all algorithms for all functions). Besides, the algorithm complexity has been taken into consideration by evaluating the computation time just for specific function for predefined evaluations of a certain dimension. Nonetheless CEC'17 proposed a new performance measure which is called score metric. Thus, the evaluation method for each algorithm is based on a score of 100 which is based on two criteria taking into account higher weights will be given for higher dimensions. Thus, the calculated score is used instead of using a statistical test [16].

In this paper, CEC'21 benchmark functions are considered to evaluate the performance of meta-heuristic algorithms. We divided the algorithms into three category, i.e., basic algorithms, advanced algorithms and CEC2021's Algorithms. The basic algorithms include the basic or standard version of the old and recent meta-heuristic algorithms and it includes the basic or standard versions of differential evolution (DE) [20], gaining-sharing knowledge-based algorithm (GSK) [21], grey wolf optimizer (GWO) [22], particle swarm optimization (PSO) [23], and teaching learning-based optimization algorithm (TLBO) [24]. The advanced algorithms are the adaptive and self-adaptive version of those algorithms which have been the winner of some CEC competitions. These algorithms include the following: LSHADE was ranked as the winner in real-parameter single objective optimization competition, CEC 2014 [25], AGSK was ranked second in real-parameter single objective optimization competition, CEC2020 [26], EBOwithCMAR is the winner of CEC2017 single objective with bound constraints competition [27], IMODE is the winner of CEC2020 single objective with bound constraints competition [28] and ELSHADE_-SPACMA algorithm was ranked third in real-parameter single objective optimization competition CEC 2018 and, is an enhanced version of LSHADE-SPACMA algorithm [29]. The CEC2021's algorithms are the set of algorithms that participated in the CEC2021 competition. These algorithms include Self-organizing Migrating Algorithm with CLustering-aided migration and adaptive Perturbation vector control (SOMA-CLP) [30]; A Multi-start Local Search Algorithm with L-SHADE (MLS-LSHADE); LSHADE based on ordered and roulette-wheel-based mutation (L-SHADE-OrdRw); LSHADE algorithm with Adaptive Archive and Selective Pressure (NL-SHADE-RSP) [31]; Self-adaptive Differential Evolution Algorithm with Population Size Reduction for Single Objective Bound-Constrained Optimization (j21) [32]; A New Step-Size Adaptation Rule for CMA-ES Based on the Population Midpoint Fitness (RB-IPOP-CMAES) [33]; Improved DE through Bayesian Heperparameter Optimization (MadDE) [34]; Gaining-Sharing Knowledge Algorithm with Adaptive Parameters Hybrid with Improved Multi-operator DE algorithm (APGSK-IMODE) [35] and DE with Distance-based Mutation-selection (DEDMNA) [36].

The objective functions of the benchmarks are parameterized using operators such as bias, rotation, and translation. The primary goal of parameterization is to evaluate the influence of all operator combinations on all benchmark functions. Parametric benchmarking is a first step towards gaining a comprehensive understanding of algorithmic performance and optimization concerns [37]. To this end, ten scalable benchmark challenges utilising these binary operators are proposed.

The contribution of the paper are summarized as:

- Evaluating the performance of the previously proposed swarm and evolutionary algorithms in the recently proposed CEC2021 competition on Single Objective Bound Constrained Numerical Optimization;
- Discussing the algorithms participated in the CEC2021 competition on Single Objective Bound Constrained Numerical Optimization;
- Comparing the performance of those algorithms based on different criteria, such as Wilcoxon test and Friedman ranking test;
- Proposing some future directions that would benefit the evolutionary computation research community.

The organization of the paper is as follows: Sect. 2 includes the literature review of the benchmark functions and algorithms. Section 3 presents the detailed description of parameterized benchmark for CEC 2021 competition. Section 4 represents the numerical experiments with the comparative results and Sect. 5 concludes the whole paper.

2 Related work

Liang et al. [13] stated that many of classical popular benchmark functions have some features that have been used by some algorithms to achieve excellent results. According to their experience, some of these shortcoming properties associated with the existing benchmark problems are:

- A Global optimum having the same parameter values for different dimensions/variables. In this case, the global optimum may be reached quickly if some operators exist to duplicate the value of one dimension to the other dimensions.
- At the origin, there is a global optimum. As a result, numerous mutation operators may be simply built to take use of this common trait of a large number of benchmark functions.
- The global optimum is located at the search range's centre. When the uniform initial population is generated randomly, the mean-centric technique has a tendency to lead the population to the search range's centre.
- The global optimum is located on the bounds of the search space that are easily discovered by the majority of methods.
- Local optima located along the coordinate axes or a lack of connectivity between the variables/dimensions. In this situation, the local optimal information might be used to determine the global optimum.

By analyzing these problems, they proposed a general framework to construct novel and challenging composition test functions possessing many desirable properties.

Recently, during the past few years, there have been a few attempts to investigate the relationship between the benchmark sets and the performance of the optimization algorithms. Mersmann et al. [38], discussed that the main objective of benchmarking optimization algorithms is to answer two questions. The first one: which algorithm is the best? And the second one: which algorithm can be used to solve a specific real-world problem? Therefore, they suggested that the first question can be answered by using consensus ranking procedures. Regarding the second question, they proposed a new term, named Exploratory Landscape Analysis (ELA), which is based on developing ways to cheaply and automatically extracts problem properties from a concrete problem instance. However, this procedure has not been applied in this study. On the other hand, they suggested many problem properties such as multi-modality, global structure, separability, variable scaling, search space homogeneity, basin size homogeneity, global to local optima contrast and plateau. Then, real-parameter black-box optimization benchmarking 2009, named BBOB'09 [39], has been used to analyze the performance of 30 optimization algorithms. they applied consensus ranking instead of using individual ranking. Thus, in order to gain insights to answer these questions, many classical statistical exploratory data analyses have been used.

Firstly, the expected running time of each algorithm has been calculated for each test function and dimension and consensus ranking has been applied. Then, distance measure has been used to calculate the distance between all rankings. Besides, to retrieve groups or clusters from the distance matrix, multidimensional scaling (MDS, [40]), to visualize the relationship between observations, and the clustering algorithm PAM [41], to find clusters or groups in data, have been used. Finally, to describe these groups, decision trees for modelling the unknown cluster boundary have been applied. In the same context, in order to answer the second question, Mersmann et al. [42], extended their work by applying Exploratory Landscape Analysis (ELA). They suggested several low-level features of functions which are convexity, y-distribution, level set, meta-model, local search and curvature. Then, they try to relate these features to the high-level features of Mersmann et al. [38]. The relationship between the problem properties and algorithm optimization can be estimated using a small sample of function values combined with statistical and machine learning techniques. In order to simultaneously optimize feature sets according to quality and cost, a multi-objective evolutionary algorithm SMS-EMOA [43], has been employed. Later on, Mersmann et al. [8], the 2009 and 2010 BBOB benchmark results [9, 44] are used to analyze more than 50 optimization algorithms. The compared algorithms have been divided into 11 groups such that all optimization algorithms based on the same base mechanism will be put into the same group. Then, in order to choose the best optimizer from each group as the representative for this group, the Borda [45] consensus over all algorithms in each group for the accuracy levels 10-3 and 10-4 has been used. Then, they applied the same approach used in Mersmann et al. [38]. Altogether, no consistent results have been reached. Thus, they concluded that the proposed features of the benchmark problems are not enough or not adequate to describe the groups of different algorithms.

Morgan and Gallagher [46] proposed a novel framework based on a length scale. The length scale is the change of the objective function value with respect to the change in the points in the search space. The main objective was to study the structural features of the landscape of the problems regardless of any particular algorithm. They discussed some important properties of the concept of the length scale and its distribution. Then, the proposed framework has been applied to the 2010 BBOB benchmark (BBOB'10) [44]. Experimental analysis and results showed that the proposed framework can differentiate easily between uni-modal and multi-modal functions. On the other hand, many researchers have focused on the benchmarking process of optimization algorithms that must be followed to perform a fair comparison. Opara and Arabas [47] proposed an overview of benchmarking procedures. Based on some functions from CEC and BBOB benchmarks, they discussed the main points in this field such as theoretical aspects of the comparison of the algorithms, available benchmarks, evaluation criteria and standard methods of presenting and interpreting results and the related appropriate statistical procedures. Finally, they proposed a novel concept of parallel implementation of test problems which is considered as a qualitative improvement to the benchmarking procedures. Beiranvand et al. [48] presented complete and detailed standards and guidelines on how to benchmark optimization algorithms. They reviewed the benchmark test suites for all types of optimization problems. Besides, they discussed the main performance measures to evaluate the efficiency of the optimization algorithms. Moreover, pitfalls to avoid and main issues that should be taken into account to conduct fair and systematic comparison have been highlighted. Muñoz et al. [49] introduced a survey of selected methods for optimization algorithm in the black-box continuous optimization problems. Firstly, the algorithm selection framework by Rice in [50] has been described. The four-component spaces- problem, algorithm, performance and characteristic- are described. Therefore, the fitness landscape concept is discussed. Then, they discussed the different classes of search algorithms. Besides, the methods used to measure algorithm performance have been introduced. A classification and a summary of well-known exploratory landscape analysis methods have been presented. Finally, they presented the implementations of the algorithm selection framework for solving continuous optimization problems. Opara et al. [37] extended CEC'17 benchmark problems by employing different parametrization to the problems. Each function is parametrized by interpretable, high-level characteristics (rotation vs nonrotation, noise vs noise-less, etc.) which are used in predicting multiple regression model to explain the algorithmic performance.

2.1 Description of used algorithms

In this subsection, all the algorithms that have been used in the paper to solve the parameterized CEC2021 test functions are briefly discussed and explained. Also, comparisons between these algorithms have been conducted at the end of the paper. These algorithms have been classified into basic, advanced and recently-developed ones.

2.1.1 Basic algorithms

- **Differential evolution (DE)**

  It is a type of evolutionary computing method that belongs to a broader family of evolutionary algorithms. DE algorithm, developed by Storn and Price in 1997 [20], is also a popular direct search technique such as genetic algorithm and evolution strategies which starts with a population of initial solutions. Then, by introducing mutations into the population, these initial solutions are iteratively improved. It is most popular evolutionary algorithm and has been applied to various nonlinear, high dimensional and complex optimization problem to obtain the optimal solution. Moreover, different variants of DE algorithm such as self adaptive, binary, multi-objective, etc. have been introduced.
- **Gaining sharing knowledge-based algorithm (GSK)**

  GSK algorithm [21] is based on the human behavior of gaining and sharing knowledge which consists of two phases: junior and senior gaining and sharing phase. In junior phase, initial solutions are produced and later, they are sent to senior phase to interact with others. This is key idea behind the GSK algorithm. Many different variants of GSK algorithm have been developed to show its capability in solving real-world optimization problems [51–53].
- **Grey Wolf optimizer (GWO)**

  The GWO algorithm is modelled after the natural leadership structure and hunting mechanism of grey wolves by Mirjalili et al. [22]. For modelling the leadership structure, four sorts of grey wolves are used: alpha, beta, delta, and omega. Furthermore, the three primary phases of hunting are implemented: looking for prey, surrounding prey, and attacking prey. Grey wolves have the capacity to locate and surround prey. The alpha is generally in charge of the hunt. Hunting is something that the beta and delta could do on occasion. They separate to hunt for prey and then converge to attack prey. The proposed technique was evaluated on 29 benchmark test functions and outperformed others. Researcher have been developed its variants to solve different problems.
- **Particle swarm optimization (PSO)**

  It is one of the nature-inspired evolutionary algorithms and stochastic optimisation techniques developed by James Kennedy and Russ Eberhart in 1995 [23] to solve computationally difficult and difficult optimisation problems. Since then, it has been applied to a wide variety of search and optimisation problems. It abstracts the mechanism of action of swarms such as swarms of birds, fish, and so on. It is an algorithm for population-based evolution in which each swarm (particle) represents a unique solution. Each particle updates its current position via its velocity vector and attempts to find the optimal solution.
- **Teaching Learning based optimization algorithm (TLBO)**

  This technique focuses on the impact of a teacher's influence on students. TLBO, like other nature-inspired algorithms, is a population-based approach that progresses to the global answer through a population of solutions [24]. A population is defined as a group of students or a class of students. The TLBO procedure is broken into two parts: The 'Teacher Phase' is the first section. and the 'Learner Phase' makes up the second portion. The terms 'Teacher Phase' and 'Learner Phase' refer to learning from a teacher and 'Learning by Interaction Between Learners' respectively. It has been successfully applied to several numerical optimization problems and it proved its superiority in solving them.

2.1.2 Advanced algorithms

- **LSHADE** [25]

  It is an extended version of the previously developed success history-based adaptive DE (SHADE) algorithm [54]. There are three main control parameters of DE algorithm i.e. population size, scaling factor F and crossover rate CR. SHADE algorithm employs a historical memory M_{CR}, M_F that saves a set of CR, F values which have worked successfully in the past. Based on SHADE algorithm, Tanabe and Fukunaga [25] developed LSHADE algorithm which is based on population size parameter for DE algorithm. In this, a linear population size reduction formula has been used which gradually decreases the population size during a DE run. The proposed algorithm LSHADE was applied to the special session on Real parameter single objective optimization benchmark suite. The performance of LSHADE algorithm outperformed other algorithms and it was ranked as winner in CEC2014 competition.

- **AGSK** [35]

  It depicts adaptive version of gaining-sharing knowledge (GSK) based optimization algorithm. The GSK algorithm has two main parameters: knowledge factor k_f and knowledge ratio k_r that control junior and senior phase during optimization. Therefore, to accommodate this situation, adaptation process has been included with the parameter settings. Moreover, population size reduction scheme is also employed to decrease the population size gradually. This leads to adaptive gaining sharing knowledge-based algorithm and the performance of the AGSK algorithm is evaluated on CEC2020 benchmark functions. It performed significantly better than other comparative algorithms due to the remarkably balance between the exploration and exploitation. In CEC2020, AGSK algorithm was ranked second among all competitive algorithms.
- **EBOwithCMAR** [27]

  This algorithm is a combination of global optimizer and a local optimizer. The global optimizer, i.e. effective butterfly optimizer (EBO) is a self-adaptive version of butterfly optimization algorithm which uses success history-based adaption and linear population size reduction to the increase the diversity of the population. While it also incorporates a co-variance matrix adapted retreat (CMAR) phase to produce the new solution. Moreover, it also increase the local search capability of EBO algorithm. This hybrid algorithm is tested over single objective CEC 2017 benchmark problems and ranked as winner of CEC2017.
- **IMODE** [28]

  It introduces an improve multi-operator differential evolution algorithm (IMODE). It begins by segmenting the initial population into many sub-population, each of which is developed using a different DE variation. The size of each sub-population is continually adjusted depending on two indicators: solution quality and sub-population variety. The proposed algorithm is tested over 10 benchmark functions of CEC2020 competitions and ranked as winner among all competitive algorithms.
- **ELSHADE_SPACMA** [29]

  Many researcher or practitioners have been introduced different variants of DE algorithm. In DE algorithm, crossover rate reflects the chance that the test solution inherits a certain gene. Montgomery and Chen [55] claimed that *CR* is a very sensitive parameter in order to solve optimization problems. Thus, to tackle with this problem, Mohamed et al. [56] proposed LSHADE_SPACMA algorithm. Furthermore, Hadi et al. [29] enhanced the performance of LSHADE_SPACMA by developing ELSHADE_SPACMA algorithm with the two improvements. The first was a hybridization of LSHADE_SPACMA and adaptive guided differential evolution (AGDE) in which all the population were assigned to LSHADE_SPACMA for one generation. And then all the population will be assigned to AGDE algorithm. The next improvement was made in the mutation parameter to balance the exploration and exploitation quality. The proposed technique has been applied to CEC2017 benchmark problems and obtained third rank among all algorithms.

2.1.3 CEC2021's algorithms

In this subsection, we will discuss and summarize the algorithms that have been participated in CEC2021 competition.

- **APGSK-IMODE** [35]

  A hybrid algorithm based on Adaptive Parameters gaining sharing knowledge (APGSK) [57] and improved multi-operator differential evolution (IMODE) [28]. For a predetermined number of generations (cycle), two sub-populations are evolved to an optimal or near-optimal solution using APGSK and IMODE. Probabilistically, each algorithm evolves its sub-population for the next cycle that is done via probability, which is adjusted based on each algorithm's quality. It is worth to mention that, APGSK-IMODE was the winner of the CEC2021 Single Objective Bound Constrained Numerical Optimization for non-shifted cases.
- **SOMA-CLP** [30]

  Kadavy et al. [30] introduced a new Self-organizing Migrating Algorithm (SOMA) variant with clustering-aided migration and adaptive perturbation vector control (SOMA-CLP). SOMA-CLP is a recent variation of SOMA, which is an improved SOMA-CL [58]. SOMA-CLP promotes the worldwide transition from exploration to exploitation by linearly adapting the *prt* control parameter. Its workflow is split into three phases: search space mapping, mapped space clustering, and further screening of areas of interest revealed in the first phase.

- **L-SHADE-OrdRw** [59]
  L-SHADE-OrdRw, which is an improved L-SHADE algorithm based on ordered and roulette-wheel-based mutation has been proposed in [59]. Also, in order to further improves the performance of L-SHADE-OrdRw, an adaptive and non-adaptive ensemble sinusoidal method, to automatically adjust the scaling factor values, has been used. Moreover, a selection pressure, which put more weights on the better solutions, based on the roulette-wheel strategy has been utilized to select random solutions in mutation strategy. To further enhance the efficiency of L-SHADE-OrdRw algorithm uses a local search based on Gaussian Walks [60].
- **NL-SHADE-RSP** [31]
  Stanovov et al. [31] proposed an improved LSHADE algorithm, called NL-SHADE-RSP. NL-SHADE-RSP incorporates different novel parameter control methods including nonlinear population size reduction, rank-based selective pressure in mutation strategy to select one of two mutation operators (with and without archive), adaptive archive set usage, and adapting the crossover rate control based on some rules. It also utilizes a mechanism to adapt the *p* value in the current-to-pbest mutation operator.
- **j21** [32]
  The j21 algorithm is built on the self-adapting jDE100 and j2020 algorithms. It uses two populations, a restart mechanism in both populations, a crowding mechanism, and a system to choose vectors from both sub-populations for the mutation process. The jDE100 [61] and j2020 [62] algorithms use similar processes. j21 algorithm uses a mechanism to reduce population size throughout evolution, unlike the prior two algorithms. The self-adaptive control parameter *CR* also has a wider range of values.
- **RB-IPOP-CMAES** [33]
  RB-IPOP-CMAES [33] is an extension of IPOP-CMAES, in which previous population midpoint fitness (PPMF) is used as an adaptation of the 1/5th success rule to the CMA-ES algorithm. PPMF is utilized to adjust the step-size multiplier σ which compares the current population's fitness value to the previous population's midpoint. The stepsize is changed to make sure that the success probability fluctuates around the reference target value.
- **MadDE** [34]
  In MadDE [34], the control settings and search techniques are simultaneously adapted in the optimization process. It is based on self-adaptive DE algorithms like JADE, SHADE [54] and LSHADE [25], whose core structure has been used to build modern DE algorithms [28]. Similar to IMODE [28], it uses several mutation and crossover processes to build trial vectors. MadBE uses various search algorithms because they are likely to deliver consistent results across a wide range of objective functions (landscape). The following describes the MadDE algorithm has several characteristics. First, it mixes existing powerful mutation strategies and selects them probabilistically and the likelihood of choosing a mutation strategy depends on its historical success rate. Second, it uses probabilistic crossover to choose between binomial and q-best binomial crossover. Third, it adapts the DE control parameters *NP*, *Cr*, and *F* using the LSHADE algorithm [25].
- **DEDMNA** [36]
  Bujok and Kolenovsky proposes and tests a new DE algorithm with distance-based mutation selection, population size reduction, and an archive for good old solutions [36], called DEDMNA, which is an improved variant of DE algorithm with a selection of mutation strategy based on the mutant point distance (DEMD) [63]. DEDMNA uses a linear population size reduction approach to and an archive to store old solutions.

In the summary, in this section, several algorithms have been discussed that have been used to solve the CEC2021 benchmark functions. These algorithms are categorized as basic algorithms, advanced algorithms and the recently developed algorithms used in the CEC2021 competitions. In this paper, a comparison between these algorithms have been conducted to see their performance on the parameterized CEC2021 problems, as to the best of the authors knowledge, there is no such comparison in the literature. The aim of this paper is to evaluate the performance of these algorithms on the recently proposed parameterized CEC2021 problems. As, it can be concluded from the results and analysis section, the recently developed algorithms have been performed better than others.

3 Parametrized benchmark

Based on the aforementioned discussion, it can be seen from CEC'05 to CEC'2020 that regardless of the type and feature of benchmark problems, the same experimental analysis approach has been used. All benchmark problems have been used with the fixed-parameter settings for all features i.e. no different values for different features have been experimentally investigated. All benchmark problems have been manipulated as a black-box with no permeation for a possible change to evaluate the performance of the algorithms with different combinations of parameters' values. Therefore, to the best of our knowledge, this is the first study that opens this black-box for all benchmark problems and tests different DE-based algorithm with different predetermined levels for different controlled features.

The benchmarks plays very important role in improvement of global meta-heuristics. The two-benchmark series CEC [64] and COCO [39] are proposed to check the performance of the real parameter meta-heuristic algorithms. In these competition, the benchmark functions are transformed by using different operators such as bias, rotation and shift in the objective function [14, 37, 65]. Various possible combination of the operators can be formed. The total eight combinations of operators are possible and they are as

- Only bias exists.
- Bias does not exist.
- Only rotation exists
- Rotation does not exist.
- Only shift exists.
- Shift does not exist.
- Bias and rotation exist but shift does not exist.
- Bias and shift exist but rotation does not exist.
- Rotation and shift exist but bias does not exist.
- Bias, shift and rotation exist simultaneously.

The main aim of benchmarking is to find the best transformation by checking the effect of all possible combination of the operators. Thus, the resulting set is known as parameterized benchmark.

In CEC'20 [17] benchmark, new objective function is defined by including shift vector, rotation matrix and bias in the original objective function. In the variable x, shift vector s_i is included with the multiplication of rotation matrix R and the bias vector b_i^* is added in the original objective function. Therefore, the mathematical formulation of the new benchmark is given as:

$$F_i(X) = f_i(R(x - s_i)) + b_i^* \tag{1}$$

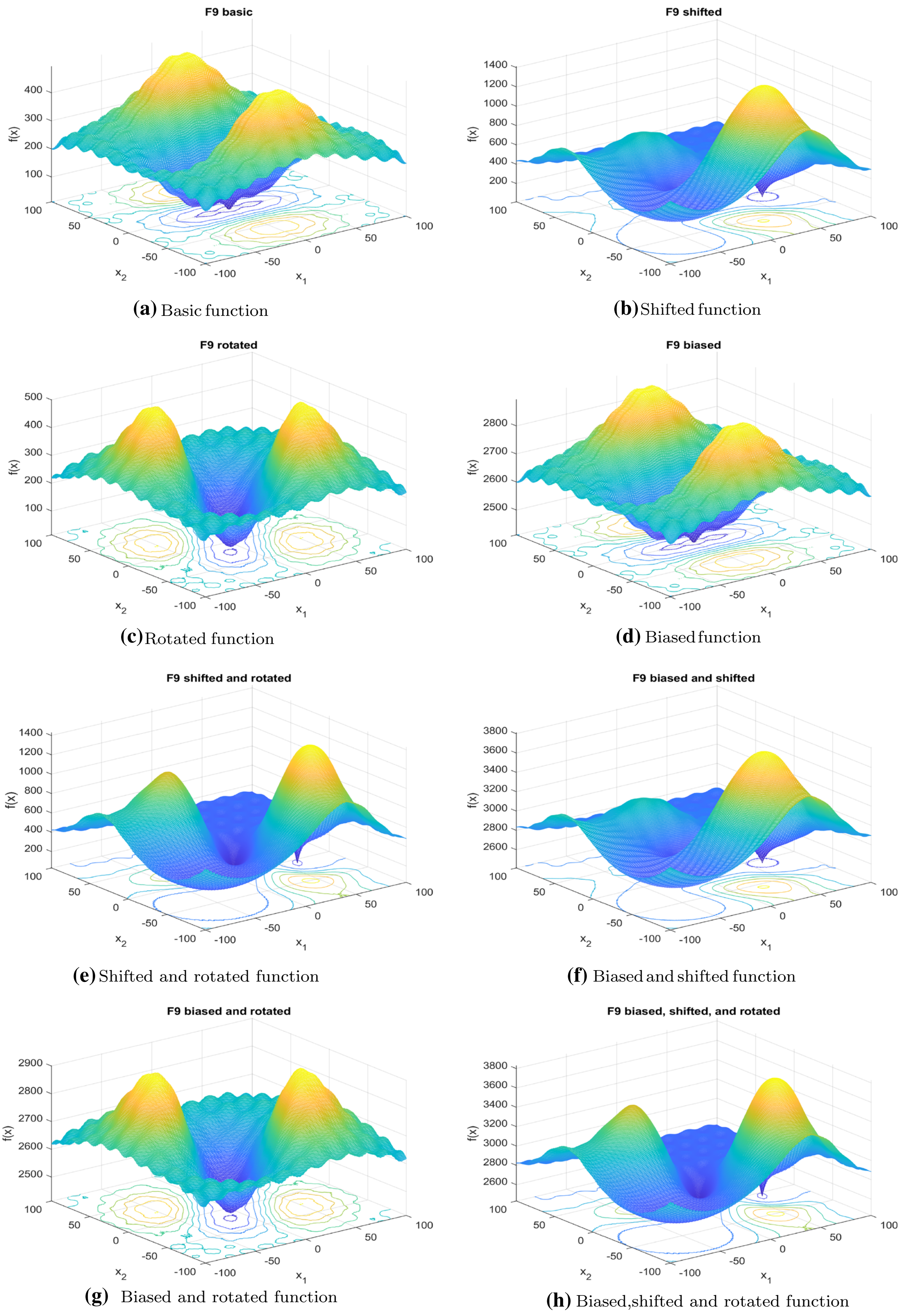
The $F_i(X)$ is known as parameterized benchmark function on which the performance of the operators will be tested. There are some detailed variations for hybrid and composition functions, which make the full pattern slightly more complicated. Therefore, the decomposition allows to define the binary parameters that demonstrate which transformation should be applied and ensures that predictors are standardized to the same scale. The values taken by the parameters is presented in Table 1 with the reference Equation number from which the value can be obtained. The operators bias, shift and rotation can be controlled, activated or deactivated. While, there are some other parameters such as problem type, separability, symmetry, and number of local optima which can be observed only. The type of problems can be unimodal, hybrid, simple multimodal, and composition, the optimization problem may be of different kind such as separable or non-separable. A few or a huge number of local optima can be existed and the shape of the problem may be symmetric or asymmetric. The values of the parameters are fixed therefore, these type of parameters can be observed only. Thus, total 8 combination are possible for each function. One example is illustrated to understand these binary parameters. For example, only shift operator exists and the rotation and bias operator do not exist then $R = I$ (identity matrix) and $F_i^* = 0$ must set on. The detailed description of each binary operator applied on function F_i has been shown in Table 2. To show the effect of these configurations on the benchmark set, F_9 has been selected as an example. The 3-D maps for 2 dimensions F_9 in all 8 configurations are shown in Fig. 1. In each figure, the subfigure (a) shows the

Table 1 Parametrization of the benchmark problems

ID	Parameter	Values	Reference	Type
C_1	Bias	F_i^* or 0	Eq. (1)	Controlled
C_2	Shift	s_i or 0	Eq. (1)	Controlled
C_3	Rotation	R or I	Eq. (1)	Controlled

Table 2 Binary parameter values for each transformation applied on the function F_i

Name of the functions	Bias	Shift	Rotation	Parameterized vector
F_i Basic	0	0	I	(000)
F_i Bias	F_i^*	0	I	(100)
F_i Shift	0	s_i	I	(010)
F_i Rotation	0	0	R	(001)
F_i Bias and shift	F_i^*	s_i	I	(110)
F_i Bias and rotation	F_i^*	0	R	(101)
F_i Shift and rotation	0	s_i	R	(011)
F_i Bias, shift and rotation	F_i^*	s_i	R	(111)



(a) Basic function

(b) Shifted function

(c) Rotated function

(d) Biased function

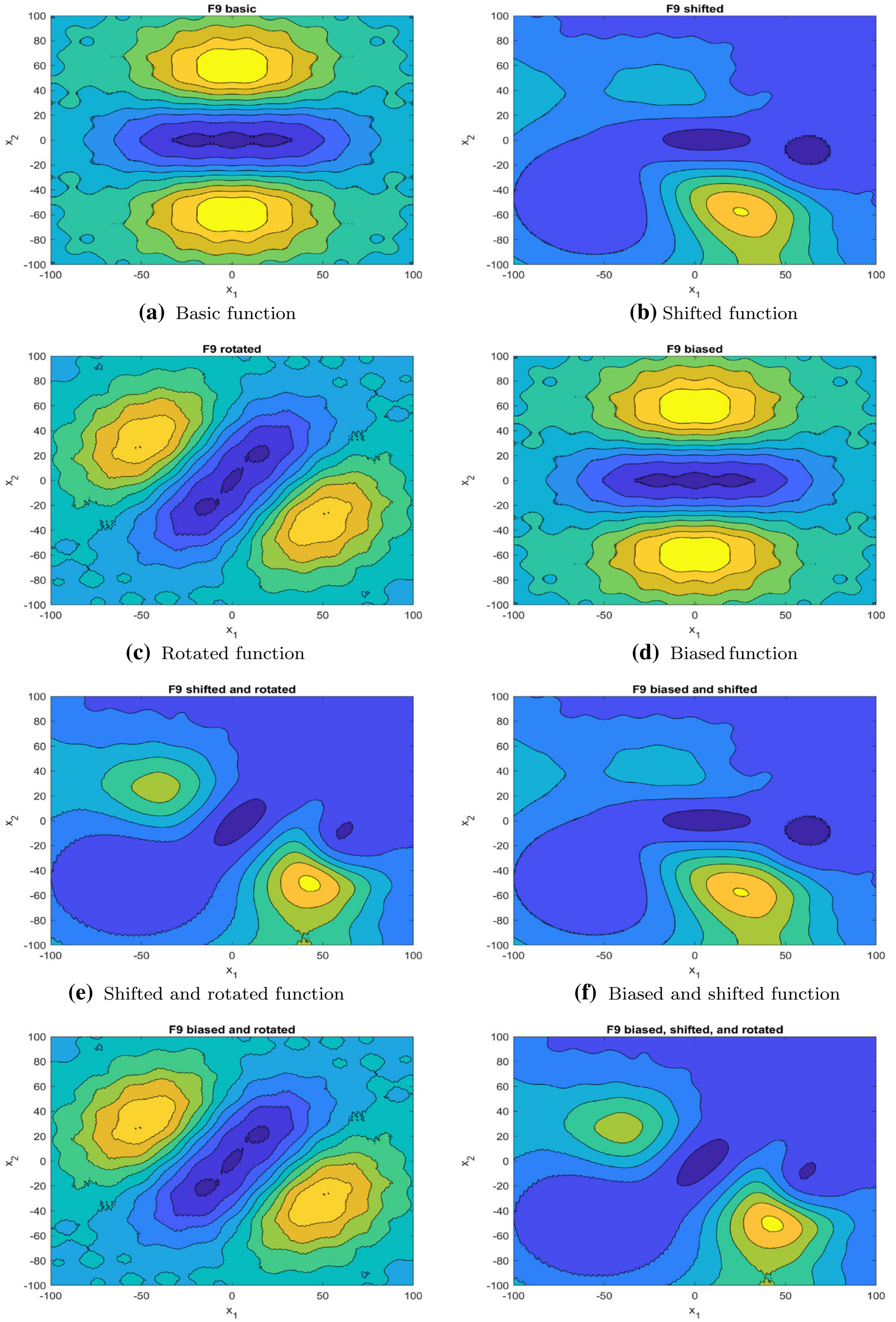
(e) Shifted and rotated function

(f) Biased and shifted function

(g) Biased and rotated function

(h) Biased,shifted and rotated function

Fig. 1 3-D map for 2-D composite parametrized function 9



(a) Basic function

(b) Shifted function

(c) Rotated function

(d) Biased function

(e) Shifted and rotated function

(f) Biased and shifted function

(g) Biased and rotated function

(h) Biased , shifted and rotated function

Fig. 2 Contour maps for composite parameterized function 9

basic 3-D map of function such that the no parametrization is used. The subfigures (b), (c), (d) present the function with only shift parameter, only rotation and only bias, respectively. In the subfigures (e), (f), (g), two operations are simultaneously used that are shift with rotation, shift with bias and rotation with bias, respectively. The subfigure (h) shows all the three parameters with the original function. These figures illustrate the effect of all parameters on the original benchmark functions. Moreover, their contour maps (For F_9) are also drawn in Fig. 2. The interested research can be found full details in CEC'21 technical report [18].

4 Numerical experiments

This section presents the numerical experiments of performance of meta-heuristic algorithms on CEC 2021 benchmark functions. The experiments conducted on two types of meta-heuristic algorithms, i.e. basic algorithms and advanced algorithms. Basic algorithms includes the basic or standard versions of differential evolution (DE) [20], gaining-sharing knowledge-based algorithm (GSK) [21], grey wolf optimizer (GWO) [22], particle swarm optimization (PSO) [23], and teaching learning-based optimization algorithm (TLBO) [24]. The advanced algorithms include the adaptive and self-adaptive version of those algorithms which have been the winner of of previously held CEC competitions. The advanced algorithms are as:

- LSHADE [25] was ranked as the winner in real-parameter single objective optimization competition, CEC 2014.
- AGSK [26] was ranked second in real-parameter single objective optimization competition, CEC2020.
- EBOwithCMAR [27] is the winner of CEC2017 single objective with bound constraints competition.
- IMODE [28] is the winner of CEC2020 single objective with bound constraints competition.
- ELSHADE_SPACMA algorithm [29] was ranked third in real-parameter single objective optimization competition CEC 2018 and, is an enhanced version of LSHADE-SPACMA algorithm [56] which was ranked fourth in real-parameter single objective optimization competition CEC 2017. Besides, IMODE, LSHADE and ELSHADE_SPACMA are DE- based algorithms.

Moreover, the performance of the state-of-the-art algorithms is evaluated on evaluation criteria [18].

The values of the parameters used for the basic algorithms are considered from the GSK algorithm paper [21] and the parameters values for the advanced algorithm are taken from their original paper. The performance of the algorithms for 10 D and 20 D are compared with the two criteria (i) among the parametrized vector for each algorithm, and (ii) among the obtained results of all algorithms for each parametrized vector.

Table 3 Ranks of all basic algorithms with each parameterized vector for 10 D

Alg	Parameterized vector	(000)	(001)	(010)	(011)	(100)	(101)	(110)	(111)
DE	R	1	5.5	3.5	7	2	5.5	3.5	8
	SE	50.00	20.85	30.59	18.50	50.00	20.85	30.59	18.50
	SR	50.00	20.45	26.79	18.75	47.87	20.45	26.79	18.75
	S	100.00	41.31	57.38	37.25	97.87	41.31	57.38	37.25
GSK	R	1	4	5.5	7.5	2	3	5.5	7.5
	SE	50.00	49.38	33.69	23.92	50.00	49.38	33.69	23.92
	SR	50.00	24.77	34.87	21.37	48.18	25.24	34.87	21.37
	S	100.00	74.14	68.56	45.29	98.18	74.62	68.56	45.29
GWO	R	1.5	3.5	5.5	7.5	1.5	3.5	5.5	7.5
	SE	50.00	0.01	0.00	0.00	50.00	0.01	0.00	0.00
	SR	50.00	35.29	22.64	17.39	50.00	35.29	22.64	17.39
	S	100.00	35.30	22.64	17.39	100.00	35.30	22.64	17.39
PSO	R	1	3	6	7	2	4	5	8
	SE	50.00	8.44	7.48	3.80	50.00	8.44	7.48	3.80
	SR	50.00	16.50	17.01	12.89	47.14	16.50	17.37	12.50
	S	100.00	24.94	24.49	16.69	97.14	24.94	24.85	16.30
TLBO	R	1.5	3.5	5.5	7.5	1.5	3.5	5.5	7.5
	SE	50.00	1.67	0.55	0.48	50.00	1.67	0.55	0.48
	SR	50.00	22.37	15.74	11.97	50.00	22.37	15.74	11.97
	S	100	24.03	16.29	12.45	100	24.03	16.29	12.45

Table 4 Ranks of all basic algorithms with each parameterized vector for 20 D

Alg	Paramertized Vector	(000)	(001)	(010)	(011)	(100)	(101)	(110)	(111)
DE	R	1.5	5.5	3.5	7.5	1.5	5.5	3.5	7.5
	SE	50.00	33.23	30.42	25.92	50.00	33.23	30.42	25.92
	SR	50.00	24.55	36.49	22.13	50.00	24.55	36.49	22.13
	S	100.00	57.78	66.91	48.05	100.00	57.78	66.91	48.05
GSK	R	1	3.5	6	7.5	2	3.5	5	7.5
	SE	50.00	36.51	26.67	23.28	50.00	36.51	26.67	23.28
	SR	50.00	23.94	23.44	17.86	47.87	23.94	24.46	17.86
	S	100.00	60.44	50.11	41.13	97.87	60.44	51.12	41.13
GWO	R	1.5	3.5	5.5	7.5	1.5	3.5	5.5	7.5
	SE	50.00	37.60	0.64	0.63	50.00	37.60	0.64	0.63
	SR	50.00	30.56	20.75	15.94	50.00	30.56	20.75	15.94
	S	100.00	68.16	21.39	16.57	100.00	68.16	21.39	16.57
PSO	R	2	4	6	8	1	3	5	7
	SE	50.00	7.87	6.78	4.29	50.00	7.88	6.78	4.29
	SR	47.30	18.42	18.04	13.06	50.00	18.82	18.42	13.06
	S	97.30	26.30	24.82	17.35	100.00	26.69	25.20	17.35
TLBO	R	2	4	6	7.5	1	3	5	7.5
	SE	50.00	4.60	0.82	0.53	50.00	4.60	0.82	0.53
	SR	45.00	23.38	16.51	13.04	50.00	24.00	16.82	13.04
	S	95.00	27.98	17.33	13.57	100	28.60	17.64	13.57

Table 5 Comparative results of basic algorithm for each parameterized vector in case of 10 dimensions

(000)						(001)					
Alg	DE	GSK	GWO	PSO	TLBO	Alg	DE	GSK	GWO	PSO	TLBO
R	3	5	1	4	2	R	2	4	3	5	1
SE	0.28	0.14	50.00	0.21	2.15	SE	49.30	28.46	50.00	22.23	41.81
SR	41.38	33.33	46.15	34.29	50.00	SR	39.22	36.36	26.32	25.64	50.00
S	41.66	33.47	96.15	34.49	52.15	S	88.52	64.83	76.32	47.88	91.81
(010)						(011)					
R	1	4	5	2	3	R	1	4	5	3	2
SE	50.00	18.01	9.56	49.97	41.72	SE	49.15	19.45	10.23	44.49	50.00
SR	50.00	35.00	21.88	38.89	43.75	SR	50.00	34.21	20.74	31.45	40.63
S	100.00	53.01	31.43	88.86	85.47	S	99.15	53.66	30.98	75.94	90.63
(100)						(101)					
R	3	5	1	4	2	R	2	4	3	5	1
SE	0.28	0.14	50.00	0.21	2.15	SE	49.30	28.46	50.00	22.23	41.81
SR	41.38	33.33	46.15	34.29	50.00	SR	39.22	36.36	26.32	25.64	50.00
S	41.66	33.47	96.15	34.49	52.15	S	88.52	64.83	76.32	47.88	91.81
(110)						(111)					
R	1	4	5	2	3	R	1	4	5	3	2
SE	50.00	18.01	9.56	49.97	41.72	SE	49.15	19.45	10.23	44.49	50.00
SR	50.00	35.00	21.88	38.89	43.75	SR	50.00	34.21	20.74	31.45	40.63
S	100.00	53.01	31.43	88.86	85.47	S	99.15	53.66	30.98	75.94	90.63

4.1 Evaluation criteria

Algorithms are evaluated with a score that is composed of two parts, SE and SR, both of which assign equal weights to 10 and 20 dimensional results. SE is based on sums of normalized error values, while SR is composed of sums of ranks. Each score contributes 50% to the total Score, which has a maximum value of 100.

Table 6 Comparative results of basic algorithm for each parameterized vector in case of 20 dimensions

(000)						(001)					
Alg	DE	GSK	GWO	PSO	TLBO	Alg	DE	GSK	GWO	PSO	TLBO
R	4	3	2	5	1	R	4	3	1	5	2
SE	2.95	3.08	16.81	4.71	50.00	SE	11.14	13.04	50.00	9.04	19.63
SR	35.61	36.72	50.00	31.76	47.96	SR	42.86	50.00	34.78	30.77	48.98
S	38.56	39.80	66.81	36.47	97.96	S	54.00	63.04	84.78	39.81	68.61
(010)						(011)					
R	3	4	5	1	2	R	3	2	5	4	1
SE	31.33	30.33	13.88	50.00	48.15	SE	33.58	34.07	16.30	38.21	50.00
SR	50.00	48.08	26.60	49.02	47.17	SR	50.00	50.00	25.00	37.10	42.59
S	81.33	78.41	40.48	99.02	95.32	S	83.58	84.07	41.30	75.31	92.59
(100)						(101)					
R	4	3	2	5	1	R	4	3	1	5	2
SE	2.95	3.08	16.81	4.71	50.00	SE	11.14	13.04	50.00	9.11	19.63
SR	35.61	36.72	50.00	31.76	47.96	SR	42.86	50.00	34.78	30.77	48.98
S	38.56	39.80	66.81	36.47	97.96	S	54.00	63.04	84.78	39.88	68.61
(110)						(111)					
R	3	4	5	1	2	R	3	2	5	4	1
SE	31.33	30.33	13.88	50.00	48.15	SE	33.58	34.07	16.30	38.21	50.00
SR	50.00	48.08	26.60	49.02	47.17	**SR**	50.00	50.00	25.00	37.10	42.59
S	81.33	78.41	40.48	99.02	95.32	S	83.58	84.07	41.30	75.31	92.59

In particular, SE begins as an average of 2 sums of normalized functional error values:

$$SNE = 0.5\sum_{m=1}^{8}\sum_{i=1}^{10} ne_{i,m}^{10D} + 0.5\sum_{m=1}^{8}\sum_{i=1}^{10} ne_{i,m}^{20D} \qquad (2)$$

where ne is an algorithm's normalized error value for a given function, configuration and dimension and SNE is the average of all normalized error values over all functions, configurations and dimensions. For this competition, ne is defined as:

$$ne = \frac{f(x_{best}) - f(x^{*})}{f(x_{best})_{max} - f(x^{*})} \qquad (3)$$

where $f(x_{best})$ is the algorithm's best result out of 30 trials, $f(x^{*})$ is the function's known optimal value and $f(x_{best})_{max}$ is the largest $f(x_{best})$ among all algorithms for the given function/dimension combination. Once SNE has been determined for all algorithms, SE is computed as:

$$SE = \left(1 - \frac{SNE - SNE_{min}}{SNE}\right) \times 50 \qquad (4)$$

where SNE_{min} is the minimal sum of normalized errors among all algorithms. $SR1$ begins as an average of 2 sums of ranks:

$$SR1 = 0.5\sum_{m=1}^{8}\sum_{i=1}^{10} rank_{i,m}^{10D} + 0.5\sum_{m=1}^{8}\sum_{i=1}^{10} rank_{i,m}^{20D} \qquad (5)$$

where rank is the algorithm's rank among all algorithms for a given function, configuration and dimension that is based on its mean error value (not normalized). Once SR has been determined for all algorithms, SR is computed as:

$$SR = \left(1 - \frac{SR1 - SR1_{min}}{SR}\right) \times 50 \qquad (6)$$

where $SR1_{min}$ is the minimal sum of ranks among all algorithms. The final Score is the sum of SE and SR:

$$Score = SE + SR \qquad (7)$$

The entries will be ranked based on the score.

Also, as the results in this paper were presented as best, medium, maximum, average and standard deviation values for each problem for the same number of runs,

Table 7 Overall ranking of all basic algorithms

Algorithms	R	SE	RS	S
DE	2	33.34803	47.67157	81.0196
TLBO	1	50	50	100
GSK	3	26.4721	42.84141	69.31351
GWO	5	24.10357	31.37097	55.47453
PSO	4	32.26599	36.76749	69.03347

Table 8 Average Rankings of the algorithms based on the Friedman Ranking test

Dimension	Algorithms	(000)	(001)	(010)	(011)	(100)	(101)	(110)	(111)
10D	DE	2.9	2.55	**2.1**	**1.95**	2.9	2.55	**2.1**	**1.95**
	GSK	3.6	2.75	3	2.85	3.6	2.75	3	2.85
	GWO	2.6	3.8	4.8	4.7	2.6	3.8	4.8	4.7
	PSO	3.5	3.9	2.7	3.1	3.5	3.9	2.7	3.1
	TLBO	**2.4**	**2**	2.4	2.4	**2.4**	**2**	2.4	2.4
20D	DE	3.3	2.8	**2.5**	2.35	3.3	2.8	**2.5**	2.35
	GSK	3.2	**2.4**	2.6	**2.25**	3.2	**2.4**	2.6	**2.25**
	GWO	**2.35**	3.45	4.7	4.6	**2.35**	3.45	4.7	4.6
	PSO	3.7	3.9	2.55	3.15	3.7	3.9	2.55	3.15
	TLBO	2.45	2.45	2.65	2.65	2.45	2.45	2.65	2.65

A result in boldface indicates the best rank

Table 9 Ranks of all advanced algorithms with each parameterized vector for 10 D

Algo	Parametrized vector	(000)	(001)	(010)	(011)	(100)	(101)	(110)	(111)
LSHADE	R	2	3	5	8	1	6	4	7
	SE	48.87	39.43	27.84	21.12	50.00	35.19	26.81	23.02
	SR	48.04	24.26	31.01	18.28	50.00	22.07	33.56	20.08
	S	96.91	63.69	58.85	39.40	100.00	57.26	60.37	43.11
AGSK	R	1	8	3.5	5	2	7	3.5	6
	SE	50.00	15.78	31.86	20.04	50.00	15.78	31.86	20.04
	SR	50.00	20.87	32.43	19.67	46.15	21.24	32.43	19.67
	S	100.00	36.64	64.29	39.72	96.15	37.01	64.29	39.72
EBOwithCMAR	R	1	4	6	8	2	3	5	7
	SE	50.00	27.26	23.13	20.25	49.15	35.61	24.20	19.94
	SR	50.00	26.80	28.26	21.14	41.94	29.89	29.55	21.85
	S	100.00	54.06	51.39	41.38	91.08	65.50	53.75	41.79
IMODE	R	1.5	4	6	8	1.5	3	5	7
	SE	50.00	49.81	12.47	7.09	50.00	49.93	12.92	7.34
	SR	50.00	31.51	22.33	16.20	50.00	32.39	21.90	17.16
	S	100.00	81.31	34.80	23.28	100.00	82.32	34.82	24.51
ELSHADE_SPACMA	R	1	4	6	7	2	3	5	8
	SE	48.96	34.84	18.77	13.33	50.00	37.80	18.88	13.34
	SR	50.00	27.78	29.41	18.52	46.30	29.07	30.12	18.25
	S	98.96	62.62	48.18	31.85	96.30	66.87	49.00	31.59

nonparametric tests such as Friedman ranking test can be used to compare the competing algorithms.

4.2 Results of basic algorithms

This section describes the performance of all basic algorithms on the CEC2021 benchmark functions. The statistical results of all algorithms are given in Tables S1-S5 in the supplementary file. Tables 3, 4, 5 and 6 show the result for the above-mentioned criteria. Table 3 presents the obtained rank, evaluation scores (SE and SR) and the total score of each algorithm for each parametrized vector in case of 10 dimensions. Table 3 shows that DE, GSK and PSO algorithm get first rank in case of basic function or when there is no parametrized vector. GWO and TLBO algorithms perform similar in case of basic function and when only bias operator is applied. Moreover, when the three binary operators are applied simultaneously, DE and PSO algorithms achieve the last rank. This implies this operator has significant effect on the both algorithms. Similarly, in case of GSK, GWO and TLBO algorithms, the effect of bias, shift and rotation operators (all of three i.e. (111)) and shift and rotation (both i.e. (110)) are same,

Table 10 Ranks of all advanced algorithms with each parameterized vector for 20 D

Algo	Parametrized Vector	(000)	(001)	(010)	(011)	(100)	(101)	(110)	(111)
LSHADE	R	2	6	3	7	1	5	4	8
	SE	49.93	25.86	30.05	19.55	50.00	27.73	29.61	18.76
	SR	50.00	27.18	37.84	21.71	50.00	28.28	35.90	22.40
	S	99.93	53.05	67.89	41.25	100.00	56.01	65.50	41.16
AGSK	R	1	5	3	7.5	2	6	4	7.5
	SE	50.00	14.46	17.60	10.05	50.00	14.04	17.39	10.05
	SR	50.00	23.79	33.11	18.85	48.04	22.48	33.11	18.85
	S	100.00	38.24	50.71	28.89	98.04	36.52	50.50	28.89
EBOwithCMAR	R	1	6	3	8	2	4	5	7
	SE	50.00	37.75	35.63	24.66	47.24	42.17	35.46	29.42
	SR	50.00	26.21	39.13	23.68	43.55	27.27	29.03	21.43
	S	100.00	63.96	74.76	48.35	90.78	69.44	64.49	50.85
IMODE	R	2	4	5	8	1	3	6	7
	SE	50.00	45.85	9.70	5.43	49.53	48.37	10.19	5.67
	SR	48.00	30.38	25.00	16.22	50.00	33.80	24.00	18.75
	S	98.00	76.23	34.70	21.64	99.53	82.17	34.19	24.42
ELSHADE_SPACMA	R	2	3	6	7	1	4	5	8
	SE	50.00	35.65	23.65	15.16	48.84	35.53	23.74	15.12
	SR	46.67	28.57	35.44	22.22	50.00	28.57	36.36	22.22
	S	96.67	64.22	59.10	37.38	98.84	64.10	60.11	37.34

as the algorithms achieve same rank in the both parameterized vectors.

Although in case of 20 dimensions, Table 4 shows that PSO and TLBO algorithm get the first rank when the bias operator is applied. PSO algorithm has significantly effect of shift and rotation operator simultaneously whereas, TLBO algorithm has similar effect of shift and rotation operator (011) and bias, shift and rotation operators (111) simultaneously. It performs similar in both the cases. DE and GWO algorithm perform similar in case of basic function and only bias operator. GSK algorithm get the first rank for basic functions.

According to (ii) criteria, the comparison is made among the algorithms for each parameterized vector. In Table 5, the comparative results are shown for dimension 10. It can be observed that, GWO algorithm outperforms others in case of basic function and only bias operator. TLBO algorithm gets the first position for only rotation operator (001) and bias and rotation operator (101). DE algorithm obtains first rank in 4 out of total 8 cases and the total score is almost equal to 100. Moreover, DE algorithm is at the second place in 2 cases ((001) and (101)). Although in case of 20 dimensions, TLBO algorithm obtains the first rank in 4 cases ((000), (011), (100), and (111)) and in the other cases it is at the second place only. The algorithm obtains total score above 90 in 6 out of 8 cases. It implies that in the higher dimensions TLBO algorithm performs better than other algorithms. When the three operators (bias, shift and rotation) has been applied to the basic function, TLBO algorithm outperforms others and the worst performance is of GWO algorithm. The total score obtained by GWO algorithm is 41.30 which is very less.

The overall comparison is made among basic algorithms and among advanced algorithms. Table 7 shows the overall comparative results obtained by all algorithms. In the basic algorithms, TLBO performs better than other basic algorithms in each parameterized vector. It gets the first rank and DE, GSK, PSO and GWO are come to next places. The score of DE algorithm which got the second rank, it score is only 81.01 out of 100.

Table 8 presents the results obtained from the basic algorithms DE, GSK, GWO, PSO and TLBO algorithms based on the Friedman ranking test. For the 10D test problems, DE algorithm obtained the best rank for the problem with shift, shift and rotation, bias and shift and the ones with bias, shift and rotation operators, while TLBO algorithm achieve the best rank for problems with other operators.

For the 20D test problems, DE obtained the best rank for the problems with shift and the ones with bias and shift operators. GWO achieved the best rank for the test

Table 11 Comparative results of advanced algorithms for each parametrized vector in case of 10 dimensions

(000)						(001)					
Alg	LSHADE	AGSK	EBOwithCMAR	IMODE	LSHADE_SPACMA	Alg	LSHADE	AGSK	EBOwithCMAR	IMODE	LSHADE_SPACMA
R	3.5	2	5	1	3.5	R	4	5	3	1	2
SE	0.00	0.00	0.00	50.00	0.00	SE	0.06	0.02	0.04	50.00	0.55
SR	29.69	31.15	26.03	50.00	29.69	SR	26.47	21.95	30.00	50.00	33.33
S	29.69	31.15	26.03	100.00	29.69	S	26.53	21.98	30.04	100.00	33.88
(010)						(011)					
R	3	2	5	1	4	R	3	2	1	4	5
SE	39.92	50.00	32.07	43.01	37.20	SE	39.05	48.02	50.00	42.56	37.72
SR	40.74	42.31	30.56	50.00	28.21	SR	35.96	31.54	50.00	31.06	28.87
S	80.66	92.31	62.62	93.01	65.41	S	75.01	79.56	100.00	73.62	66.59
(100)						(101)					
R	2	3	5	1	4	R	4	5	3	1	2
SE	0.00	0.00	0.00	50.00	0.00	SE	0.03	0.01	0.03	50.00	0.20
SR	30.33	29.84	24.03	50.00	29.37	SR	25.00	21.95	31.03	50.00	34.62
S	30.33	29.84	24.03	100.00	29.37	S	25.03	21.97	31.06	100.00	34.82
(110)						(111)					
R	3	2	5	1	4	R	2	3	1	4	5
SE	30.57	50.00	23.74	45.28	33.06	SE	44.51	50.00	45.64	43.15	38.87
SR	42.59	44.23	31.08	50.00	31.08	SR	40.74	34.92	50.00	34.38	29.33
S	73.16	94.23	54.82	95.28	64.14	S	85.25	84.92	95.64	77.53	68.20

Table 12 Comparative results of advanced algorithms for each parametrized vector in case of 20 dimensions

(000)						(001)					
Alg	LSHADE	AGSK	EBOwithCMAR	IMODE	LSHADE_SPACMA	Alg	LSHADE	AGSK	EBOwithCMAR	IMODE	LSHADE_SPACMA
R	4	2	5	1	3	R	4	5	3	1	2
SE	0.00	0.00	0.00	50.00	0.01	SE	1.25	0.79	1.24	50.00	3.25
SR	30.60	35.96	28.87	50.00	32.03	SR	27.94	24.36	32.76	50.00	32.76
S	30.60	35.97	28.88	100.00	32.04	S	29.19	25.15	34.00	100.00	36.00
(010)						(011)					
R	3	2	1	4	5	R	2	4	1	5	3
SE	44.77	46.48	50.00	42.60	41.91	SE	41.66	33.05	50.00	27.23	38.95
SR	47.46	50.00	47.46	49.12	40.58	SR	40.91	31.69	50.00	30.41	40.91
S	92.23	96.48	97.46	91.72	82.49	S	82.57	64.74	100.00	57.64	79.86
(100)						(101)					
R	4	2	5	1	3	R	4	5	2	1	3
SE	0.05	0.07	0.03	50.00	0.09	SE	0.69	0.47	0.97	50.00	2.13
SR	30.60	35.96	28.08	50.00	33.06	SR	29.41	25.64	35.71	50.00	34.48
S	30.64	36.04	28.12	100.00	33.15	S	30.11	26.11	36.69	100.00	36.61
(110)						(111)					
R	3	2	4	1	5	R	2	4	1	5	3
SE	47.67	44.32	50.00	46.66	42.99	SE	44.12	34.25	50.00	29.04	41.39
SR	45.76	50.00	40.30	49.09	41.54	SR	37.72	29.45	50.00	32.58	35.25
S	93.43	94.32	90.30	95.75	84.53	S	81.84	63.70	100.00	61.61	76.63

Table 13 Overall ranking of all advanced algorithms

Algorithms	R	SE	RS	S
LSHADE	3	39.49021	39.58333	79.07354
AGSK	5	36.90186	37.3442	74.24606
IMODE	1	50	50	100
EBOwithCMAR	4	38.40685	40.11329	78.52013
ELSHADE_SPACMA	2	46.06877	38.07429	84.14306

problems with basic and bias, while GSK obtained the best rank for the remaining categories.

4.3 Results of Advanced Algorithms

This section includes the performance of advance algorithms on CEC2021 benchmark functions. The statistical results of all algorithms are given in Tables S6-S10 in the supplementary file. Tables 9, 10, 11 and 12 show the results obtained by the advanced algorithms. Table 9 shows the rank obtained by the algorithms for each parameterized vector for dimension 10. LSHADE algorithm performs better when the bias operator is applied. It obtains the first rank in this case. There is a significant effect of shift and rotation operators on LSHADE algorithm therefore, it did not perform well in (011) case. AGSK and EBOwithCMAR get first rank only in the basic function means that no operator has been applied, whereas the only rotation operator has significant effect on AGSK algorithm. The combination of shift and rotation operators affect the EBOwithCMAR algorithm. IMODE algorithm perform similar in case of basic functions and with the bias operator. Therefore, it obtains the same rank in the both the cases. ELSHADE_SPACMA algorithm gets the first rank in the basic function only and did not perform well in case of all three operators.

Table 10 presents the result of advanced algorithms of dimensions 20. When the dimension increases, ELSHADE_SPACMA and IMODE algorithms obtain the first rank in the (100) case only. AGSK and EBOwithCMAR algorithms outperform in the only basic functions and LSHADE algorithm gets first rank in case of the inclusion of bias operator only. According to Table 10, LSHADE and ELSHADE_SPACMA algorithms have the significant effect of all three operators simultaneously.

Now the comparison is made among the advanced algorithms for each operator. Table 11 presents the obtained rank and score for the dimensions 10. It shows that IMODE algorithm obtains first rank in 6 out of 8 cases and EBOwithCMAR algorithm get first rank in rest of two cases. When the only shift and rotation operator (011) and three operator (111) are applied, IMODE does not perform better. However, in case of 20 dimensions, IMODE algorithms outperforms other advanced algorithms in 5 cases. The results are depicted in Table 12. AGSK algorithm obtains first rank in only 1 case that is (010) and EBOwithCMAR algorithm gets first rank in (011) and (111) case.

For the overall comparison, the results are shown in Table 13. IMODE algorithm outperforms other in the advanced algorithm category and obtains the first place among others. ELSHADE_SPACMA, LSHAEDE, EBOwithCMAR and AGK algorithm take place. It can be said that TLBO and IMODE algorithm are the most efficient and capable algorithm among the mentioned algorithms to solve the CEC2021 benchmark problems.

Table 14 presents the results obtained from the advanced algorithms LSHADE, AGSK, EBOwithCMAR, IMODE and LSHADE_SPACMA algorithms based on the Friedman ranking test. For the 10D test problems, EBOwithCMAR algorithm obtained the best rank for the problem with (011) and (111) operators, while IMODE algorithm achieved the best rank for the remaining problems.

Table 14 Average Rankings of the algorithms based on the Friedman Ranking test

Dimension	Algorithms	(000)	(001)	(010)	(011)	(100)	(101)	(110)	(111)
10D	LSHADE	3.2	3.4	2.7	2.9	3.05	3.6	2.7	2.75
	AGSK	3.05	4.1	2.6	3.3	3.1	4.1	2.6	3.15
	EBOwithCMAR	3.65	3	3.6	**2.05**	3.85	2.9	3.7	**2.25**
	IMODE	**1.9**	**1.8**	**2.2**	3.25	**1.85**	**1.8**	**2.3**	3.2
	LSHADE_SPACMA	3.2	2.7	3.9	3.5	3.15	2.6	3.7	3.65
20D	LSHADE	3.25	3.35	2.9	2.7	3.25	3.4	2.9	2.8
	AGSK	2.95	3.9	**2.85**	3.55	2.95	3.9	**2.7**	3.65
	EBOwithCMAR	3.55	2.95	2.95	**2.25**	3.65	2.8	3.35	**2.15**
	IMODE	**2.05**	**1.9**	**2.85**	3.7	**2.05**	**2**	2.75	3.3
	LSHADE_SPACMA	3.2	2.9	3.45	2.8	3.1	2.9	3.3	3.1

A result in boldface indicates the best rank

Table 15 Ranks for all CEC2021's algorithms on each parameterized vector for 10D

Alg	Criteria	000	001	010	011	100	101	110	111
APGSK_IMOSE	R	1.5	3	5.5	7.5	1.5	4	5.5	7.5
	SE	50.00	50.00	0.00	0.00	50.00	50.00	0.00	0.00
	SR	50.00	47.46	25.23	21.05	50.00	45.90	25.23	21.05
	S	100.00	97.46	25.23	21.05	100.00	95.90	25.23	21.05
DEDMNA	R	2	6	3	7.5	1	5	4	7.5
	SE	50.00	4.86	20.85	4.63	50.00	4.86	19.00	4.63
	SR	39.80	18.93	22.41	15.23	50.00	19.31	22.94	15.23
	S	89.80	23.80	43.26	19.86	100.00	24.16	41.94	19.86
MadDE	R	1.5	3	5	8	1.5	4	6	7
	SE	50.00	50.00	0.00	0.00	50.00	50.00	0.00	0.00
	SR	50.00	47.41	25.94	19.93	50.00	45.83	24.12	20.52
	S	100.00	97.41	25.94	19.93	100.00	95.83	24.12	20.52
RB_IPOP_CMAES_PPMF	R	1	3	8	7	2	4	5	6
	SE	39.99	34.65	4.51	6.23	50.00	29.67	5.76	6.03
	SR	50.00	46.55	20.45	22.69	39.71	43.55	24.55	23.08
	S	89.99	81.20	24.97	28.92	89.71	73.22	30.31	29.11
J21	R	1	7	3	8	2	5	4	6
	SE	50.00	0.00	0.01	0.00	15.01	0.00	0.01	0.01
	SR	46.08	19.75	30.92	19.58	50.00	21.17	30.13	19.92
	S	96.08	19.75	30.94	19.59	65.01	21.17	30.14	19.92
NL-SHADE-RSP	R	1	5	3	7	2	6	4	8
	SE	50.00	1.01	8.58	1.11	21.99	0.88	1.56	0.88
	SR	50.00	21.63	26.47	18.60	50.00	19.40	27.78	18.29
	S	100.00	22.65	35.05	19.70	71.99	20.27	29.34	19.18
SOMA-CLP	R	2	5	4	7	1	8	3	6
	SE	22.91	3.28	6.23	2.10	50.00	3.10	7.74	2.83
	SR	43.15	30.58	44.37	28.64	43.15	25.61	50.00	30.29
	S	66.06	33.87	50.60	30.73	93.15	28.71	57.74	33.12
MLS-LSHADE	R	3.5	2	6	8	3.5	1	5	7
	SE	0.00	0.00	0.00	0.00	0.00	50.00	0.00	0.00
	SR	37.50	50.00	21.43	16.78	37.50	50.00	21.82	18.32
	S	37.50	50.00	21.43	16.78	37.50	100.00	21.82	18.32
L-SHADE-OrdRW	R	2.5	4	5.5	8	1	2.5	5.5	7
	SE	50.00	50.00	0.00	0.00	50.00	50.00	0.00	0.00
	SR	44.55	42.98	21.88	17.38	50.00	44.55	21.88	17.63
	S	94.55	92.98	21.88	17.38	100.00	94.55	21.88	17.63

For the 20D test problems, AGSK obtained the best rank for the problems with (010), and (110) problems. EBOwithCMAR achieved the best rank based on the Friedman test for (011) and (111) test problems, while the IMODE attained the best rank for the remaining test functions.

4.4 Results and discussion of CEC2021 algorithms

In this section, the performance of the algorithms that participated in the CEC2021 is presented and analyzed. The list of the algorithms are below:

1. SOMA-CLP: Self-organizing Migrating Algorithm with CLustering-aided migration and adaptive Perturbation vector control [30];
2. MLS-LSHADE: A Multi-start Local Search Algorithm with L-SHADE;
3. L-SHADE-OrdRw: LSHADE based on ordered and roulette-wheel-based mutation.
4. NL-SHADE-RSP: LSHADE algorithm with Adaptive Archive and Selective Pressure [31].
5. j21: Self-adaptive Differential Evolution Algorithm with Population Size Reduction for Single Objective Bound-Constrained Optimization [32].

Table 16 Ranks for all CEC2021's algorithms on each parameterized vector for 20D

Alg	Criteria	000	001	010	011	100	101	110	111
APGSK_IMOSE	R	2.5	2.5	5.5	7	2.5	2.5	5.5	8
	SE	50.00	50.00	0.00	0.00	50.00	50.00	0.00	0.00
	SR	50.00	50.00	29.00	20.57	50.00	50.00	29.00	19.73
	S	100.00	100.00	29.00	20.57	100.00	100.00	29.00	19.73
DEDMNA	R	2	5	4	8	1	6	3	7
	SE	50.00	4.12	7.33	2.84	49.96	4.12	7.34	2.84
	SR	43.10	26.04	31.25	19.23	50.00	25.00	32.89	19.23
	S	93.10	30.16	38.58	22.07	99.96	29.12	40.23	22.07
MadDE	R	2.5	2.5	5	8	2.5	2.5	6	7
	SE	50.00	50.00	0.00	0.00	50.00	50.00	0.00	0.00
	SR	50.00	50.00	28.71	20.28	50.00	50.00	28.16	20.57
	S	100.00	100.00	28.71	20.28	100.00	100.00	28.16	20.57
RB_IPOP_CMAES_PPMF	R	2	3	5	7	4	1	6	8
	SE	44.74	27.92	8.91	8.30	28.63	50.00	7.27	7.98
	SR	43.65	42.31	26.44	23.31	36.67	50.00	25.46	20.83
	S	88.39	70.23	35.35	31.60	65.29	100.00	32.73	28.81
J21	R	2	5	3	8	1	6	4	7
	SE	50.00	2.19	4.25	1.09	47.46	1.75	4.26	1.16
	SR	46.23	23.56	35.51	18.28	50.00	23.56	31.82	18.85
	S	96.23	25.75	39.76	19.37	97.46	25.31	36.08	20.01
NL-SHADE-RSP	R	2	6	3	7	1	5	4	8
	SE	8.83	0.18	0.24	0.11	50.00	0.23	0.31	0.12
	SR	50.00	21.00	25.93	15.79	50.00	21.88	24.14	15.11
	S	58.83	21.18	26.17	15.90	100.00	22.11	24.45	15.22
SOMA-CLP	R	1	5	4	8	2	6	3	7
	SE	50.00	12.76	13.14	7.30	41.38	13.14	14.23	7.06
	SR	50.00	25.48	36.30	20.54	46.49	22.84	40.77	21.54
	S	100.00	38.24	49.44	27.84	87.87	35.99	55.00	28.60
MLS-LSHADE	R	2	1	5	7	3	4	6	8
	SE	0.00	50.00	0.00	0.00	0.00	0.00	0.00	0.00
	SR	45.38	50.00	31.05	23.98	42.75	38.31	30.41	21.85
	S	45.38	100.00	31.05	23.98	42.75	38.31	30.41	21.85
L-SHADE-OrdRW	R	4	1	5	7	2.5	2.5	6	8
	SE	50.00	50.00	0.00	0.00	50.00	50.00	0.00	0.00
	SR	44.64	50.00	22.52	18.38	45.45	45.45	21.37	17.86
	S	94.64	100.00	22.52	18.38	95.45	95.45	21.37	17.86

6. RB-IPOP-CMAES: A New Step-Size Adaptation Rule for CMA-ES Based on the Population Midpoint Fitness [33].
7. MadDE: Improved DE through Bayesian Heperparameter Optimization [34].
8. APGSK-IMODE: Gaining-Sharing Knowledge Algorithm with Adaptive Parameters Hybrid with Improved Multi-operator DE algorithm [35]
9. DEDMNA: DE with Distance-based Mutation-selection [36].

The detailed of the obtained results (Best, Worst, Median, Mean and Std) form the above-mentioned algorithms for 10*D* and 20*D* are presented at the Supplementary material file in Table S11-S19.

Tables 15, 16, 17 and 18 present the results produced from the CEC2021's algorithms. Table 15 presents the rank obtained from the competing algorithms for every parameterized vector for the 10D problems. APGSK_IMODE and MadDE algorithms obtain the first rank for basic and bias cases. They also gives better results for rotation and when the combination of bias and rotation are applied. Their performance is affected when the shift operator is applied.

Table 17 Comparative results of CEC2021's algorithms for each parameterized vector in case of 10D problems

Alg	Criteria	DEDMNA	APGSK_IMODE	MadDE	RB_IPOP_CMAES_PPMF	j21	NL-SHADE-RSP	SOMA-CLP	MLS-LSHADE	L-SHADE-OrdRW
(000)	R	6	1.5	1.5	9	7	5	8	4	3
	SE	0.00	50.00	50.00	0.00	0.00	0.00	0.00	0.00	50.00
	SR	32.84	50.00	50.00	20.94	28.39	36.02	25.00	37.64	47.86
	S	32.84	100.00	100.00	20.94	28.39	36.02	25.00	37.64	97.86
(001)	R	6	1	2	8	7	5	9	4	3
	SE	0.00	50.00	50.00	0.00	0.00	0.00	0.00	0.00	50.00
	SR	25.00	50.00	49.14	20.21	20.65	25.45	17.92	45.97	48.31
	S	25.00	100.00	99.14	20.21	20.65	25.45	17.92	45.97	98.31
(010)	R	1	4	6	9	3	2	7	5	8
	SE	50.00	19.20	7.18	1.95	45.44	40.87	8.26	14.26	3.46
	SR	50.00	30.12	27.17	15.24	30.12	41.67	20.16	25.77	17.01
	S	100.00	49.33	34.36	17.20	75.56	82.53	28.42	40.03	20.47
(011)	R	4	1	5	9	2	3	7	6	8
	SE	36.71	47.45	24.80	9.81	50.00	30.98	19.23	19.14	13.39
	SR	41.78	44.20	32.45	21.33	33.15	50.00	23.46	26.99	24.40
	S	78.49	91.66	57.25	31.14	83.15	80.98	42.70	46.14	37.79
(100)	R	6	2	2	9	7	5	8	4	2
	SE	0.00	50.00	50.00	0.00	0.00	0.00	0.00	0.00	50.00
	SR	33.33	50.00	50.00	21.25	28.81	36.56	25.37	38.20	50.00
	S	33.33	100.00	100.00	21.25	28.81	36.56	25.37	38.20	100.00
(101)	R	5	1.5	3	8	7	6	9	4	1.5
	SE	0.00	50.00	50.00	0.00	0.00	0.00	0.00	50.00	50.00
	SR	25.45	50.00	48.31	20.50	20.65	24.57	17.92	45.24	50.00
	S	25.45	100.00	98.31	20.50	20.65	24.57	17.92	95.24	100.00
(110)	R	1	4	6	9	2	3	7	5	8
	SE	50.00	9.08	6.70	2.04	45.12	7.60	9.88	15.71	3.42
	SR	50.00	30.77	25.26	14.91	25.81	33.33	21.82	24.00	16.78
	S	100.00	39.85	31.96	16.95	70.92	40.93	31.70	39.71	20.21
(111)	R	4	3	6	9	2	1	7	5	8
	SE	24.53	27.00	15.25	5.07	50.00	41.88	14.94	21.23	7.77
	SR	38.89	40.91	30.88	22.03	36.63	50.00	26.69	28.90	26.03
	S	63.42	67.91	46.14	27.10	86.63	91.88	41.63	50.13	33.80

A comparison is conducted among the CEC2021's algorithms for every parameterized operator. Table 17 shows the obtained rank and score for problems with 10 variables. The results show that APGSK_IMODE achieves the first rank in 5 cases out of 8, DEDMNA obtains the first in 2 cases while NL-SHADE-RSP obtains the first rank in the remaining case. APGSK_IMODE is ranked 4th when used to solve (010) and (110) and 3rd when all the three operators are used (111).

Table 18 shows the obtained rank and score for problems with 20 variables. The results show that APGSK_IMODE and MadDE obtain the first rank in 4 cases out of 8. However, APGSK_IMODE obtains better rank than MadDE in the other 4 cases. J21 and NL-SHADE-RSP obtain the first rank in 2 and 2 cases out of 8 cases, respectively.

5 Conclusions and future directions

This paper presents the performance of various meta-heuristic algorithms on CEC 2021 benchmark functions. The benchmark objective functions are parameterized by including the different combinations of bias, shift and

Table 18 Comparative results of CEC2021's algorithms for each parameterized vector in case of 20D problems

Alg	Criteria	DEDMNA	APGSK_IMODE	MadDE	RB_IPOP_CMAES_PPMF	j21	NL-SHADE-RSP	SOMA-CLP	MLS-LSHADE	L-SHADE-OrdRW
(000)	R	6	1.5	1.5	9	7	4	8	5	3
	SE	0.00	50.00	50.00	0.00	0.00	0.00	0.00	0.00	50.00
	SR	30.77	50.00	50.00	19.51	29.63	37.21	22.86	32.00	45.71
	S	30.77	100.00	100.00	19.51	29.63	37.21	22.86	32.00	95.71
(001)	R	6	2	2	8	7	5	9	4	2
	SE	0.00	50.00	50.00	0.00	0.00	0.00	0.00	50.00	50.00
	SR	22.92	50.00	50.00	18.84	22.18	23.31	17.63	38.73	50.00
	S	22.92	100.00	100.00	18.84	22.18	23.31	17.63	88.73	100.00
(010)	R	3	4	5	9	1	2	6	7	8
	SE	24.61	21.09	16.00	7.62	50.00	25.18	18.56	14.19	12.28
	SR	32.24	26.92	27.22	15.12	31.82	50.00	21.88	24.75	17.01
	S	56.84	48.01	43.23	22.74	81.82	75.18	40.44	38.93	29.29
(011)	R	4	6	5	9	3	1	7	2	8
	SE	48.30	35.48	36.79	18.24	41.02	50.00	33.32	47.18	30.32
	SR	30.50	31.12	32.45	22.43	38.61	50.00	23.64	39.61	24.21
	S	78.80	66.60	69.24	40.66	79.63	100.00	56.97	86.79	54.52
(100)	R	6	1.5	1.5	9	7	4	8	5	3
	SE	0.00	50.00	50.00	0.00	0.00	0.00	0.00	0.00	50.00
	SR	31.86	50.00	50.00	19.82	30.09	36.93	23.21	32.50	47.79
	S	31.86	100.00	100.00	19.82	30.09	36.93	23.21	32.50	97.79
(101)	R	6	1.5	1.5	8	7	5	9	4	3
	SE	0.00	50.00	50.00	0.00	0.00	0.00	0.00	0.00	50.00
	SR	22.08	50.00	50.00	18.66	21.03	22.46	16.99	34.87	47.32
	S	22.08	100.00	100.00	18.66	21.03	22.46	16.99	34.87	97.32
(110)	R	3	4	5	9	1	2	6	7	8
	SE	24.73	21.20	15.50	5.97	50.00	27.71	20.42	13.95	12.21
	SR	36.96	29.31	28.65	15.55	31.48	50.00	23.39	23.83	17.83
	S	61.69	50.51	44.15	21.52	81.48	77.71	43.82	37.78	30.04
(111)	R	3	4	5	9	2	1	7	6	8
	SE	47.38	41.54	35.90	17.87	42.52	50.00	32.03	36.17	29.46
	SR	29.50	32.42	34.30	21.38	39.33	50.00	22.52	32.78	22.69
	S	76.88	73.96	70.20	39.25	81.86	100.00	54.55	68.95	52.16

rotation operators. These parameterized functions are evaluated on the some basic and advanced meta-heuristic algorithms. The performance of the meta-heuristic algorithms are based on the evaluation metric which includes the score. The score is composed of two parts, i.e. sum of normalized error values and sum of the ranks and each score contributes 50% to the total score of 100. 10 benchmark functions are considered with 10 and 20 dimensions. Two types of meta-heuristic algorithms, basic and advanced, are considered which includes different algorithms. The obtained results of these algorithms state that Teaching learning-based optimization algorithm outperforms other in case of basic algorithm. It gets the first rank among others for overall comparison. It presents the best results with the bias operator in case of 20 dimensions. Although it depicts that it has significant effect for both parametrized vectors i.e. shift and rotation operator (011) and bias, shift and rotation operators (111). Moreover, IMODE is one of the advanced algorithms which performs better than others and place itself at the first position. It performs best with the case of bias operator only for 20 dimensions. IMODE algorithm presents the best results among other algorithms in case of (000), (001), (010), (100), (101) and (110) for 10 dimensions. Moreover, It gets the total score of 100 for (000), (001), (100), (101), (110) parameterized vectors in case of 20 dimensions. Among the algorithms that participated in CEC2021 competitions, APGSK-IMODE obtains the best results for both 10D and 20D.

This paper benefits to researchers to get the various meta-heuristic algorithms under one roof. It presents the results of the standard and advanced algorithms for 10 and 20 dimensions. It can be extended for the higher dimensions such as 30, 50, 100, 500, and 1000. Besides the mentioned algorithms, more algorithms must be included to check the performance on benchmark functions. Moreover, the other evaluation criteria or performance metrices must be developed to evaluate the performance of the algorithms. The interested researchers can be considered these objectives for the further study.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00521-022-07788-z.

References

1. Michalewicz Z, Dasgupta D, Le Riche RG, Schoenauer M (1996) Evolutionary algorithms for constrained engineering problems. Comput Ind Eng 30(4):851–870
2. Gusel L, Rudolf R, Brezocnik M (2015) Genetic based approach to predicting the elongation of drawn alloy. Int J Simul Modell 14(1):39–47
3. Zhang J, Zhan Z-H, Lin Y, Chen N, Gong Y-J, Zhong J-H, Chung HS, Li Y, Shi Y-H (2011) Evolutionary computation meets machine learning: a survey. IEEE Comput Intell Mag 6(4):68–75
4. Collange G, Delattre N, Hansen N, Quinquis I, Schoenauer M (2010) Multidisciplinary optimization in the design of future space launchers
5. Mora AM, Squillero G (2015) Applications of evolutionary computation. In: Proceedings 18th European Conference, EvoApplications 2015, Copenhagen, Denmark, April 8-10, 2015,. Springer, vol. 9028
6. Forestiero A (2021) Metaheuristic algorithm for anomaly detection in internet of things leveraging on a neural-driven multiagent system. Knowl-Based Syst 228:107241
7. Cicirelli F, Forestiero A, Giordano A, Mastroianni C (2016) Transparent and efficient parallelization of swarm algorithms. ACM Trans Autonom Adap Syst (TAAS) 11(2):1–26
8. Mersmann O, Preuss M, Trautmann H, Bischl B, Weihs C (2015) Analyzing the bbob results by means of benchmarking concepts. Evol Comput 23(1):161–185
9. Hansen N, Auger A, Ros R, Finck S, Pošík P (2010) Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009. In: Proceedings of the 12th annual conference companion on Genetic and evolutionary computation, pp. 1689–1696
10. Hansen N, Auger A, Finck S, Ros R (2012) Real-parameter black-box optimization benchmarking: experimental setup," Orsay, France: Université Paris Sud, Institut National de Recherche en Informatique et en Automatique (INRIA) Futurs, Équipe TAO. Tech, Rep
11. Hansen N, Auger A, Mersmann O, Tusar T, Brockhoff D (2016) Coco: a platform for comparing continuous optimizers in a black-box setting. arXiv preprint arXiv:1603.08785
12. Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y-P, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. KanGAL Rep 2005005(2005):2005
13. Liang J, Qu B, Suganthan P, Hernández-Díaz AG (2013) Problem definitions and evaluation criteria for the cec, (2013) special session and competition on real-parameter optimization. Technical Report 201212
14. Liang J-J, Suganthan PN, Deb K (2005) Novel composition test functions for numerical global optimization. In: Proceedings, (2005) IEEE swarm intelligence symposium. SIS 2005. IEEE 2005:68–75
15. Liang JJ, Qu BY, Suganthan PN (2013) Problem definitions and evaluation criteria for the cec, (2014) special session and competition on single objective real-parameter numerical optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore 635
16. Wu G, Mallipeddi R, Suganthan PN (2017) Problem definitions and evaluation criteria for the cec 2017 competition on constrained real-parameter optimization. National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report
17. Yue C, Price K, Suganthan P, Liang J, Ali M, Qu B, Awad N, Biswas P (2019) Problem definitions and evaluation criteria for the cec 2020 special session and competition on single objective bound constrained numerical optimization. Comput Intell Lab, Zhengzhou Univ., Zhengzhou, China, Tech. Rep, vol. 201911
18. Mohamed A, Hadi A, Mohamed A, Agrawal P, Kumar A, Suganthan P (2020) Problem definitions and evaluation criteria for the cec 2021 special session and competition on single objective bound constrained numerical optimization. In: Tech. Rep
19. García S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary

algorithms' behaviour: a case study on the cec'2005 special session on real parameter optimization. J Heurist 15(6):617

20. Storn R, Price K (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11(4):341–359
21. Mohamed AW, Hadi AA, Mohamed AK (2020) Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm. Int J Mach Learn Cybern 11:1501–1529
22. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
23. J. Kennedy and R. C. Eberhart (1997) A discrete binary version of the particle swarm algorithm. In: IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation, 1997, vol. 5. IEEE, pp. 4104–4108
24. Rao RV, Savsani VJ, Vakharia D (2012) Teaching-learning-based optimization: an optimization method for continuous nonlinear large scale problems. Inf Sci 183(1):1–15
25. Tanabe R, Fukunaga AS (2014) Improving the search performance of shade using linear population size reduction. In: IEEE congress on evolutionary computation (CEC). IEEE, 2014, pp. 1658–1665
26. Mohamed AW, Hadi AA, Mohamed AK, Awad NH (2020) Evaluating the performance of adaptive gaining sharing knowledge based algorithm on cec, (2020) benchmark problems. in: 2020 IEEE congress on evolutionary computation (CEC). IEEE, pp. 1–8
27. Kumar A, Misra RK, Singh D (2017) Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase. In: IEEE congress on evolutionary computation (CEC). IEEE, 2017, pp. 1835–1842
28. Sallam KM, Elsayed SM, Chakrabortty RK, Ryan MJ (2020) Improved multi-operator differential evolution algorithm for solving unconstrained problems. In: IEEE congress on evolutionary computation (CEC). IEEE, 2020, pp. 1–8
29. Hadi AA, Mohamed AW, Jambi KM (2021) Single-objective real-parameter optimization: enhanced lshade-spacma algorithm. In: Heuristics for optimization and learning. Springer, pp. 103–121
30. Kadavy T, Pluhacek M, Viktorin A, Senkerik R (2021) Soma-clp for competition on bound constrained single objective numerical optimization benchmark: a competition entry on bound constrained single objective numerical optimization at the genetic and evolutionary computation conference (gecco) 2021. In: Proceedings of the genetic and evolutionary computation conference companion, pp. 11–12
31. Stanovov V, Akhmedova S, Semenkin E (2021) Nl-shade-rsp algorithm with adaptive archive and selective pressure for cec, (2021) numerical optimization. In: IEEE congress on evolutionary computation (CEC). IEEE, 2021, pp. 809–816
32. Brest J, Maučec MS, Bošković B (2021) Self-adaptive differential evolution algorithm with population size reduction for single objective bound-constrained optimization: algorithm j21. In: IEEE congress on evolutionary computation (CEC). IEEE, 2021, pp. 817–824
33. Warchulski E, Arabas J (2021) A new step-size adaptation rule for cma-es based on the population midpoint fitness. In: IEEE congress on evolutionary computation (CEC). IEEE, 2021, pp. 825–831
34. Biswas S, Saha D, De S, Cobb AD, Das S, Jalaian BA (2021) Improving differential evolution through Bayesian hyperparameter optimization. In: IEEE congress on evolutionary computation (CEC). IEEE, 2021, pp. 832–840
35. Mohamed AW, Hadi AA, Agrawal P, Sallam KM, Mohamed AK (2021) Gaining-sharing knowledge based algorithm with adaptive parameters hybrid with imode algorithm for solving cec, (2021) benchmark problems. In: 2021 IEEE congress on evolutionary computation (CEC). IEEE, pp. 841–848
36. Bujok P, Kolenovsky P (2021) Differential evolution with distance-based mutation-selection applied to cec, (2021) single objective numerical optimisation. In: 2021 IEEE congress on evolutionary computation (CEC). IEEE, pp. 849–856
37. Opara KR, Hadi AA, Mohamed AW (2020) Parametrized benchmarking: an outline of the idea and a feasibility study. In: Proceedings of the 2020 genetic and evolutionary computation conference companion, pp. 197–198
38. Mersmann O, Preuss M, Trautmann H (2010) Benchmarking evolutionary algorithms: towards exploratory landscape analysis. In: International conference on parallel problem solving from nature. Springer, pp. 73–82
39. Hansen N, Finck S, Ros R, Auger A (2009) Real-parameter black-box optimization benchmarking 2009: noiseless functions definitions
40. Friedman J, Hastie T, Tibshirani R (2001) The elements of statistical learning, vol 1. Springer, New York
41. Kaufman L, Rousseeuw PJ (2009) Finding groups in data: an introduction to cluster analysis. Wiley, New York, p 344
42. Mersmann O, Bischl B, Trautmann H, Preuss M, Weihs C, Rudolph G (2011) Exploratory landscape analysis. In: Proceedings of the 13th annual conference on genetic and evolutionary computation, pp. 829–836
43. Beume N, Naujoks B, Emmerich M (2007) Sms-emoa: Multiobjective selection based on dominated hypervolume. Eur J Oper Res 181(3):1653–1669
44. Auger A, Finck S, Hansen N, Ros R (2010) Bbob 2009: comparison tables of all algorithms on all noiseless functions
45. Borda Jd (1784) "Mémoire sur les élections au scrutin," Histoire de l'Academie Royale des Sciences pour 1781 (Paris, 1784)
46. Morgan R, Gallagher M (2012) Length scale for characterising continuous optimization problems. In: International conference on parallel problem solving from nature. Springer, pp. 407–416
47. Opara K, Arabas J (2011) Benchmarking procedures for continuous optimization algorithms. J Telecommun Inf Technol pp. 73–80
48. Beiranvand V, Hare W, Lucet Y (2017) Best practices for comparing optimization algorithms. Optim Eng 18(4):815–848
49. Muñoz MA, Sun Y, Kirley M, Halgamuge SK (2015) Algorithm selection for black-box continuous optimization problems: a survey on methods and challenges. Inf Sci 317:224–245
50. Rice JR (1976) The algorithm selection problem. Adv Comput 15:65–118
51. Agrawal P, Ganesh T, Mohamed AW (2021) A novel binary gaining-sharing knowledge-based optimization algorithm for feature selection. Neural Comput Appl 33(11):5989–6008
52. Agrawal P, Ganesh T, Mohamed AW (2021) Chaotic gaining sharing knowledge-based optimization algorithm: an improved metaheuristic algorithm for feature selection. Soft Comput 25:9505–9528
53. Agrawal P, Ganesh T, Oliva D, Mohamed AW (2021) S-shaped and v-shaped gaining-sharing knowledge-based algorithm for feature selection. Appl Intell pp. 1–32
54. Tanabe R, Fukunaga A (2013) Success-history based parameter adaptation for differential evolution. In: IEEE congress on evolutionary computation. IEEE, pp. 71–78
55. Montgomery J, Chen S (2010) An analysis of the operation of differential evolution at high and low crossover rates. In: IEEE congress on evolutionary computation. IEEE, pp. 1–8
56. Mohamed AW, Hadi AA, Fattouh AM, Jambi KM (2017) Lshade with semi-parameter adaptation hybrid with cma-es for solving cec, (2017) benchmark problems. In: 2017 IEEE congress on evolutionary computation (CEC). IEEE, pp. 145–152

57. Mohamed AW, Abutarboush HF, Hadi AA, Mohamed AK (2021) Gaining-sharing knowledge based algorithm with adaptive parameters for engineering optimization. IEEE Access, 9: 934–946
58. Kadavy T, Pluhacek M, Viktorin A, Senkerik R (2020) Self-organizing migrating algorithm with clustering-aided migration. In: Proceedings of the 2020 genetic and evolutionary computation conference companion, pp. 1441–1447
59. Mousavirad SJ, Moghadam MH, Saadatmand M, Chakrabortty RK (2021) An ordered and roulette-wheel-based mutation incorporated l-shade algorithm for solving cec2021 single objective numerical optimisation problems. In: Proceedings of the genetic and evolutionary computation conference companion, pp. 1–2
60. Awad NH, Ali MZ, Suganthan PN, Reynolds RG (2016) An ensemble sinusoidal parameter adaptation incorporated with l-shade for solving cec2014 benchmark problems. In: IEEE congress on evolutionary computation (CEC). IEEE, 2016, pp. 2958–2965
61. BrestJ, Maučec MS, Bošković B (2019) The 100-digit challenge: algorithm jde100. In: 2019 IEEE congress on evolutionary computation (CEC). IEEE, pp. 19–26
62. Brest J, Maučec MS, Bošković B (2020) Differential evolution algorithm for single objective bound-constrained optimization: Algorithm j2020. In: IEEE congress on evolutionary computation (CEC). IEEE, 2020, pp. 1–8
63. Bujok P (2016) Improving the convergence of differential evolution. In: International conference on numerical analysis and its applications. Springer, pp. 252–260
64. Awad N, Ali M, Liang JJ, Qu B, Suganthan P (2016) Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective real-parameter numerical optimization. In: Technical Reports
65. Liang JJ, Baskar S, Suganthan PN, Qin AK (2006) Performance evaluation of multiagent genetic algorithm. Nat Comput 5(1):83–96

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.