Engineering Science and Technology, an International Journal 41 (2023) 101388

HOSTED BY

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

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

Chaotic golden ratio guided local search for big data optimization

Havva Gül Koçer [a], Bahaeddin Türkoğlu [b], Sait Ali Uymaz [c,*]

[a] Selcuk University, Konya, Turkey
[b] Department of Computer Engineering, Nigde Omer Halisdemir University, Nigde, Turkey
[c] Department of Computer Engineering, Konya Technical University, Konya, Turkey

ARTICLE INFO

Article history:
Received 13 December 2022
Revised 4 March 2023
Accepted 11 March 2023
Available online 22 March 2023

Keywords:
Big Data Optimization
Local search
Golden ratio
Chaotic map
Memetic algorithm

ABSTRACT

Biological systems where order arises from disorder inspires for many metaheuristic optimization techniques. Self-organization and evolution are the common behaviour of chaos and optimization algorithms. Chaos can be defined as an ordered state of disorder that is hypersensitive to initial conditions. Therefore, chaos can help create order out of disorder. In the scope of this work, Golden Ratio Guided Local Search method was improved with inspiration by chaos and named as Chaotic Golden Ratio Guided Local Search (CGRGLS). Chaos is used as a random number generator in the proposed method. The coefficient in the equation for determining adaptive step size was derived from the Singer Chaotic Map. Performance evaluation of the proposed method was done by using CGRGLS in the local search part of MLSHADE-SPA algorithm. The experimental studies carried out with the electroencephalographic signal decomposition-based optimization problems, named as Big Data optimization problem (Big-Opt), introduced at the Congress on Evolutionary Computing Big Data Competition (CEC'2015). Experimental results have shown that the local search method developed using chaotic maps has an effect that increases the performance of the algorithm.

1. Introduction

Optimization is the technique of finding the most suitable one among the possible solutions for a particular problem. Many problems we encounter in the real world, such as timetabling, path planning, packing, traveling salesman, trajectory optimization, and engineering design problems, basically point to an optimization problem. The problem dimension, the number of possible solutions and the problem-specific constraints are several factors that affect the complexity of an optimization problem. Finding the ideal solution for highly complex optimization problems increases the cost (time, memory, etc.). The methods used to solve optimization problems fall into two categories: deterministic and stochastic. Deterministic methods use gradient methods to reach the ideal solution. Deterministic methods achieve the same ideal solution when operated under the same initial conditions. On the other hand, stochastic methods try to reach the ideal solution using gradient-free techniques and contain randomness. Metaheuristic methods which are stochastic, aim to improve multiple agents until they reach the specific stopping criteria by operating multiple iterations with their own rules. As the final solution, the best solution obtained at the end of the iteration is presented [1]. Although metaheuristic methods do not guarantee the optimum solution, they stand out with their ability to be applied easily to many problems and find solutions quickly [2]. These methods have a wider application area since they are not problem-dependent [3].

Metaheuristic optimization algorithms have two important search strategies: exploration and exploitation. Exploration refers to the ability of a method to explore the search space globally, avoiding and getting rid of the local minimum. Exploitation, on the other hand, is the ability to discover possible new solutions close to the current solution and improve the existing solution to obtain better solutions. The performance of an algorithm depends on using exploration and exploitation in a balanced way.

The Big-Opt problem set, which is utilized for performance evaluation in this paper, is modeled in the Optimization of Big Data 2015 Competition by Goh et al. It is an NP-Hard Big Data optimization problem (Big-Opt) abstracted from dealing with EEG signals through ICA. This problem was categorized as a big data problem which is complex and challenging because it involves a significant number of decision variables, is nonlinear, and necessitates handling EEG signals in real-time [4,5]. As the number of decision variables increases, the complexity of these problems grows exponentially, making it increasingly difficult to explore all possi-

* Corresponding author.
E-mail address: sauymaz@ktun.edu.tr (S.A. Uymaz).
Peer review under responsibility of Karabuk University.

https://doi.org/10.1016/j.jestch.2023.101388
2215-0986/

ble solutions and more likely to stuck with local solutions. This results in a high computational cost for evaluating the objective function, which is used to determine the quality of a solution. Given these challenges, it is needed to develop efficient, effective, and robust methods for solving these problems, with a high-quality solution. One of such effective approaches is the memetic algorithm.

Memetic Algorithms, which get their inspiration from a combination of population's individuals' natural adaptation and lifelong learning, are created by combining population-based algorithms and local search methods. Thanks to the collaboration between global search and local search, memetic algorithms provide higher performance than using either stand-alone local search or stand-alone population-based global search. Local search methods aim to move an existing solution of the problem to a better solution in each new iteration, and the search continues until the improvement stops. In global search, unlike local search, some solutions that do not create improvement are also considered in order to escape from the local optimum solutions. The effectiveness of the memetic algorithm is highly influenced by the local search technique. This study's goal is to provide a local search method that can be used within memetic algorithms, one of the most effective techniques for solving large scale optimization problems (LSGO).

Chaos is a nonlinear dynamic system with random behavior that is sensitive to initial conditions. It appears random, but randomness is not necessarily required to provide chaotic behavior. In recent years, chaotic maps have been used instead of random number generators used in metaheuristic methods. In particular, many successful applications have presented improvements in the performance of metaheuristic approaches by using chaotic maps to create the initial population, and the optimization technique's convergence speed to the global optima and search the solution space. In the part of the proposed method which provides the movements that guide the local search, the number obtained from the chaotic map is used as the coefficient. Thus, a chaotic effect on the step size change which causes causatives in motions that drive the local search has occurred. As a result, CGRGLS is designed.

The reminder of the paper is organized as follows. In section 2, related works as literature review about Big-Opt Problems and Chaos are given. Section 3 explains the proposed local search method in detail. Performance evaluation and experimental results are given in section 4. Finally, in section 5, conclusions are made.

2. Related works

The aim of this study is to improve the Golden Ratio Guided Local Search method, inspired by chaos theory, and to use the new approach on Big Data Optimization Problems. Source research was carried out in this direction.

Chaos is defined mathematically as "randomness" produced by simple deterministic systems. Although it may seem random and unpredictable due to the sensitivity of chaotic systems to initial conditions, it has a regularity quality as well, so chaos can provide order to arise from disorder [6]. When the uses of chaos with metaheuristic methods are examined, it is seen that the studies are carried out for two basic purposes. One of them is to use chaos as a random number generator, and another is to utilize chaotic search as the local search method. Typically, the initial population of individuals representing possible solutions to the problem is one of the influential factors for metaheuristic algorithms performance. Because algorithms improve these individuals in the population with operations to reach the best solutions in the search space. Therefore, chaos is used as a random number generator in metaheuristic algorithms and the number obtained from this generator is used for the initial population generation or as a coefficient in the operations performed during the possible solution improvement process [6–11].

The chaos approach is preferred as a useful method to ensure performance stability of algorithms by balancing exploration and exploitation [8]. There are many algorithms which have chaotic version such as genetic [12], Firefly [13], Krill Herd [14], PSO [15], Differential Evolution [11], Butterfly [16], Grey Wolf[17], Grasshopper [18], Fruit Fly [19] etc. The methods using chaos have a high-level mixing capability in all of these approaches and this leads to solutions that are more diverse and mobile. These kinds of algorithms behave better than original forms in terms of the output due to the ergodicity and mixing properties of chaos [19].

Local search methods act by applying local changes from solution to solution in the space of candidate solutions until a satisfactory solution is found. Therefore, different local search methods are incorporated into optimization algorithms for enhancing their search ability. There are a lot of local search methods designed for this purpose and used within optimization algorithms such as chaotic search, gravitational search, golden section search, multiple trajectory search, variable neighborhood search, greedy randomized adaptive search, etc. [20–26].

In this paper, Big-Opt benchmark is used for evaluation the performance of proposed method. This problem set has a multi-objective version and a single objective version. This publication focuses on a single objective version therefore, the studies were examined in this context. When the studies on big data optimization problems are examined, it is understood that swarm intelligence-based algorithms or evolutionary algorithms cannot perform successfully on these problems as they are designed. For this reason, it is seen that these algorithms are modified according to the problem and its properties or strengthened with new mechanisms before being used to solve Big-Opt problems.

One of these mechanisms is a population initiation technique introduced by Majdouli et al. They used data directly from the original EEG signal rather than randomly initializing parameters of the potential solutions within the specified lower and upper bounds in the Fireworks Algorithm [27]. Subsequently, they introduced a Fireworks Algorithm based Framework (FAF) for Big Data optimization that uses besides the population initiation technique a modified search strategy to enhance the exploration capabilities of the classical fireworks algorithm [28]. Loukdache et al. proposed a new variant of the Clonal Selection Algorithm (Big-OPT CLONALG) by using this population initiation technique and a search strategy based on the hyper-mutation operator for the single objective form of Big optimization problems [29]. Another algorithm using this technique is the source-dependent Harmony Search Algorithm (slinkHSA) [30].

Differential Evolution (DE) algorithm is one of the best performing algorithms in the field of large-scale optimization [6,8]. Therefore, it has also been preferred in solving Big-Opt problems and many improvements were made on. Elsayed and Sarker proposed an adaptive configuration of DE (ACDEs) which determines the best variant to evolve individuals from three different DE variants in parallel and additionally applies a local search to enhance the exploitation capability of the algorithm [4]. In their other study, they also proposed a general differential evolution framework that starts solving problems with multiple initial populations instead of a single initial population, and uses DE algorithms with different candidate generation mechanisms and control parameters for each population [5]. Another DE algorithm (CC-HDE) which is supported by a cooperative co-evolution method and heterogeneous memetic approach yields superior outcomes compared to other adaptive DE methods for the mentioned big data optimization problems [31].

Meselhi et al. designed a fast differential evolution algorithm which is a DE Framework by using the parallel computing power

of a graphics processing unit (GPU). The techniques it uses can be listed as an adaptive way to control the DE parameters, parallel operations to reduce computational time, a gradient based local search to increase the algorithm's rate of convergence [32,33].

Arslan and Aslan modified the standard ABC algorithm with some techniques (initial solution generation approach, neighborhood competition, crossover operators, lattice-based approach) taking into account the big data features, and suggested the lattice-based ABC (LBABC) algorithm [34]. By adapting the ABC algorithm to take into account the unique properties of big data optimization issues, a novel variant of the algorithm known as genetic big data ABC, or simply gdatABC is introduced by Aslan and Karaboğa [35]. In another study of Aslan, the standard ABC algorithm and its well-known variants on handling big data optimization challenges are compared. According to his experimental investigations, the conventional ABC (COABC) algorithm is still protecting its advantageous sides [36].

Many different algorithms have been used to solve Big-Opt problems. Cao et al. proposed a nature inspired algorithm (PBO) that mimics the motion properties of materials in three different phase types (solid phase, liquid phase and gas phase) by using corresponding operators (perturbation operator, flowing operator, diffusion operator) [37]. A novel form of the Immune Plasma algorithm (IPA) known as regional IPA (rIPA) used a pandemic management strategy centered on restricting the free movement across regions to solve Big-Opt problems [38]. Zhang et al. proposed the multi-agent genetic algorithm for solving a single objective EEG optimization problem and called as MAGA-BigOpt. In this algorithm, neighborhood competition and self-learning operators are redesigned by combining with crossover and mutation operators [39].

3. Methodology

3.1. Chaotic maps

Chaos theory refers to chaotic dynamic nonlinear systems. These systems show high sensitivity to initial parameters. Minor chaotic improvements in the preliminary parameters cause significant changes in the output and performance of the system. However, chaos systems are random. Although chaotic systems show random behavior, they do not need random parameters. Another feature of these systems is ergodicity. The ergodicity property of chaos can ensure that chaotic variables traverse all states non-repeatedly inside a certain range according to their laws. So, this can be used to avoid falling into the local minimum solution. Thanks to these features, chaotic maps inspired by chaos systems are frequently used to increase the performance of population-based optimization methods [40].

Chaotic maps are stochastic, deterministic, and nonlinear strategy frequently used to generate long-term random numbers. The numbers in the sequences created by these maps have great advantages such as not falling into repetitions, being spread over a wide spectrum instead of being stuck in a certain region, and low sequence production and storage costs. Because the randomly generated numbers in metaheuristic algorithms could be the same and within a certain range, these random numbers sometimes cause algorithms to be stuck into local minimums. For such reasons, chaotic sequences with a certain systematic, which can encompass the entire spectrum and do not fall into repetition, can be produced, and these sequences can be used instead of using randomly generated numbers. Although there are many different types of chaotic maps in the literature, there are ten chaotic maps that are frequently preferred in studies. These maps: Chebyshev [41], Circle [42], Gaussian [43], Iterative [44], Logistic [45], Piecewise [46], Sine [47], Singer [15], Sinusoidal [48] and Tent [13] chaotic maps. Chaotic sequences are created from these chaotic map equations with an initial parameter of the desired size. Instead of using random numbers in the required position, the next number is drawn from the generated chaotic sequence and used. The numbers in this sequence are unlikely to be the same and spread over a wide spectrum [49]. The chaotic maps formulation is given in Table 1. As a result of the evaluations detailed in section 4, it was decided to use singer as a chaotic map within the proposed method. The spectrums of 100 numbers generated using Singer Chaotic Map is shown in Fig. 1.

The use of long-period random number sequences plays an important role in optimization processes. Ideal sequences of random numbers should have a spread spectrum, be non-uniform, have a low cost of storage, and should not take much time to generate. However, since the random number sequences used generally have a uniform distribution, the generated numbers can be stacked in a certain range or the same values can be repeated. This may cause the algorithm to decrease the diversity in the production of new individuals or to get stuck in the local minimum. The use of chaotic maps can help to eliminate these negativities. Because chaotic structures make it possible to scan a larger solution space, increasing the probability of convergence to the global optimum without being stuck with the local optimum.

3.2. Chaotic golden ration guided local search (CGRGLS)

In this paper, Golden Ratio Guided Local Search was improved using chaotic step size and its performance was tested in MLSHADE-SPA which is a memetic framework. In CGRGLS, a local search process is implemented on the best individual which has the best fitness value in the population along each dimension from the first to the last. There is an initial step size that is 0.1 corresponds for each dimension of problem and these step sizes are stored in a Dynamic Step Size (dss) array which has an equal item with dimension number. The step size of each dimension varies throughout the search process depending on how the best individual's fitness value improves. In order to obtain a better new solution by improving the existing solution, opposite points are evaluated in each dimension as much as the step size of that dimension. The most important factor affecting the performance of this method is the adaptive step size. In the previous version of GRGLS [50], the adaptive step size was changed using a coefficient based on the golden ratio and the inverse golden ratio. In the new proposed version, which is named as CGRGLS, the value produced by the chaotic map is used as another coefficient in addition to the existing equation. In this way, the change of the step size, which provides the movements that guide the local search, in the golden ratio guidance, is affected chaotically. Flowchart of the CGRGLS is presented in Fig. 2. All following equations in flowchart, x_i^t represents current position,x_i^{t+1} represents next position, dss_i^t represents current step size of corresponding dimension, dss_i^{t+1} represents next step size of corresponding dimension, GR represents golden ratio 1.618, IGR represents inverse golden ratio 0.618, CN represents chaotic number produced by Singer Chaotic Map.

The time complexity of an algorithm is a measure of the amount of time it takes for the algorithm to complete its task as a function of the size of its input. Big O notation provides an upper bound on the number of operations required as a function of the size of the input. The main subject of this study is to develop a local search algorithm that can be used within memetic algorithms or frameworks. Therefore, the local search algorithm proposed in the time complexity analysis is focused on. The proposed method CGRGLS works on each dimension of best solution one by one for improving

Table 1
Ten Different Chaotic Maps.

No.	Map Name	Definition
M1	Chebysev map [41]	$x_{k+1} = coscos(k(x_k))$
M2	Circle map[42]	$x_{k+1} = mod(1)$
M3	Gauss [43]	$x_{k+1} = \{0 x_k = 0, \frac{1}{x_k mod(1)} = \frac{1}{x_k} - \left[\frac{1}{x_k}\right] \frac{1}{x_k mod(1)} otherwise, x \geq 0$
M4	Iterative map [44]	$x_{k+1} = sinsin(\frac{a\pi}{2}), a \in (0,1)$
M5	Logistic map [45]	$x_{k+1} = ax_k(1-x_k)$
M6	Piecewise map [46]	$x_{k+1} = \{\frac{x_k}{P}, 0 \leq x_k < P \frac{x_k - P}{0.5 - P}, P \leq x_k < \frac{1}{2}\frac{1-P-x_k}{0.5-P}, \frac{1}{2} \leq x_k < 1 - P\frac{1-x_k}{P}, 1-P \leq x_k < 1$
M7	Sine map [47]	$x_{k+1} = \frac{a}{4} sinsin(\pi x_k), 0 \leq a < 4$
M8	Singer map [15]	$x_{k+1} = \mu(7.86x_k - 23.31x_k^2 + 28.75x_k^3 - 13.302875x_k^4) \mu = 1.07$
M9	Sinusodial map [48]	$x_{k+1} = ax_k^2 sinsin(\pi x_k)$
M10	Tent map[13]	$x_{k+1} = \{\frac{x_k}{0.7}, x_k < 0.7 \frac{10}{3}(1-x_k), x_k \geq 0.7$

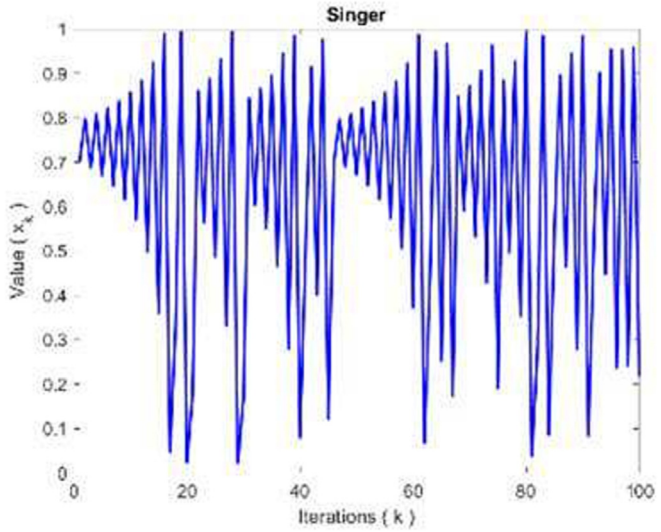

Fig. 1. Singer Chaotic Map Spectrum.

the quality of the best solution found. As seen in Fig. 2, the proposed algorithm consists of only one loop containing equation, assignment, comparison instructions, each of which costs O(1). This loop is repeated as many as the number of dimensions. Therefore, the time complexity of proposed local search algorithm is O (n) according to Big O notation. n represents dimension size of best individual in this notation. This means that the running time of the algorithm increases linearly as the input size increases.

3.3. Main algorithm MLSHADE-SPA

Algorithm created by combining population-based algorithms and local search methods are called memetic algorithm. It is an effective option for solving large-scale optimization problems. The objective of this study is to develop a local search method that can be utilized within memetic algorithms. So a memetic algorithm is needed to evaluate the performance of the proposed local search method.

Memetic algorithms or frameworks may use one or more local search methods due to their design. In order to evaluate the effectiveness and success of the proposed method, memetic algorithms using a single local search method have been considered. In particular, algorithms using local search methods that have common features with the proposed method are emphasized. For this reason, MLSHADE-SPA, which has proven its success in LSGO competitions at CEC conferences, has been chosen as the main framework.

MLSHADE-SPA is a memetic framework designed by combining three population-based algorithms with a local search method to solve large-scale global optimization problems [51]. While MLSHADE-SPA stands for LSHADE quasi-parameter adaptation memetic framework, LSHADE-SPA means "differential evolution based on success history with linear population size reduction and quasi-parameter adaptation". In this algorithm, the population-based algorithms LSHADE-SPA [52], EADE [53], and ANDE [54] are employed for global exploration, while the local search method MMTS [51] is employed for local exploitation. As a part of its search process, MLSHADE-SPA employs a variety of techniques, including population reduction, divide and conquer, and sharing computational resources. For detail information [50,51].

In this study, MLSHADE-SPA is improved by replacing its local search method (MMTS) with Chaotic Golden Ratio Guided Local Search (CGRGLS) and it is named **Chaotic Improved MLSHADE-SPA (CIMLSHADE-SPA).** Fig. 3 shows in blue where CGRGLS is used in the corresponding algorithm.

4. Performance evaluation

4.1. Experimental environment

The computer platform used for the performance evaluation tests is based on an Intel(R) Core i7-7700HQ processor running at 2.80 GHz, 16.0 GB of RAM, and the Microsoft Windows 10 Enterprise operating system. MATLAB was used for all experimental work. The MLSHADE-SPA code, which served as the basic basis for our tests, can be downloaded from the authors' website [55]. In addition, the proposed method CGRGLS was also coded in MATLAB.

4.2. Benchmark functions

EEG is a test that measures the electrical activity in the brain and records brain wave patterns through electrodes placed on the scalp. EEG signals play an important role in detecting neurological disorders and certain conditions in the nervous system. With EEG, non-brain electrical activities which are called artifact such as swallowing, coughing, eye and heart movements can be recorded by electrodes. Noise in EEG signals increases the examination and interpretation time and might prevent diagnosing neurological disorders accurately [56]. For the purpose of removing artifacts, the signals are passed through an Independent Component Analysis, which is a well-known blind source separation (BSS) technique for the removal of artifacts, to reconstruct the supposed true brain signals [57]. Goh et al. modeled an NP-Hard Big Data optimization problem (Big-Opt) abstracted from dealing with EEG signals through ICA in the Optimization of Big Data 2015 Competition. This problem has a huge number of decision variables, nonlinear and requires a real-time handling of EEG signals, so it

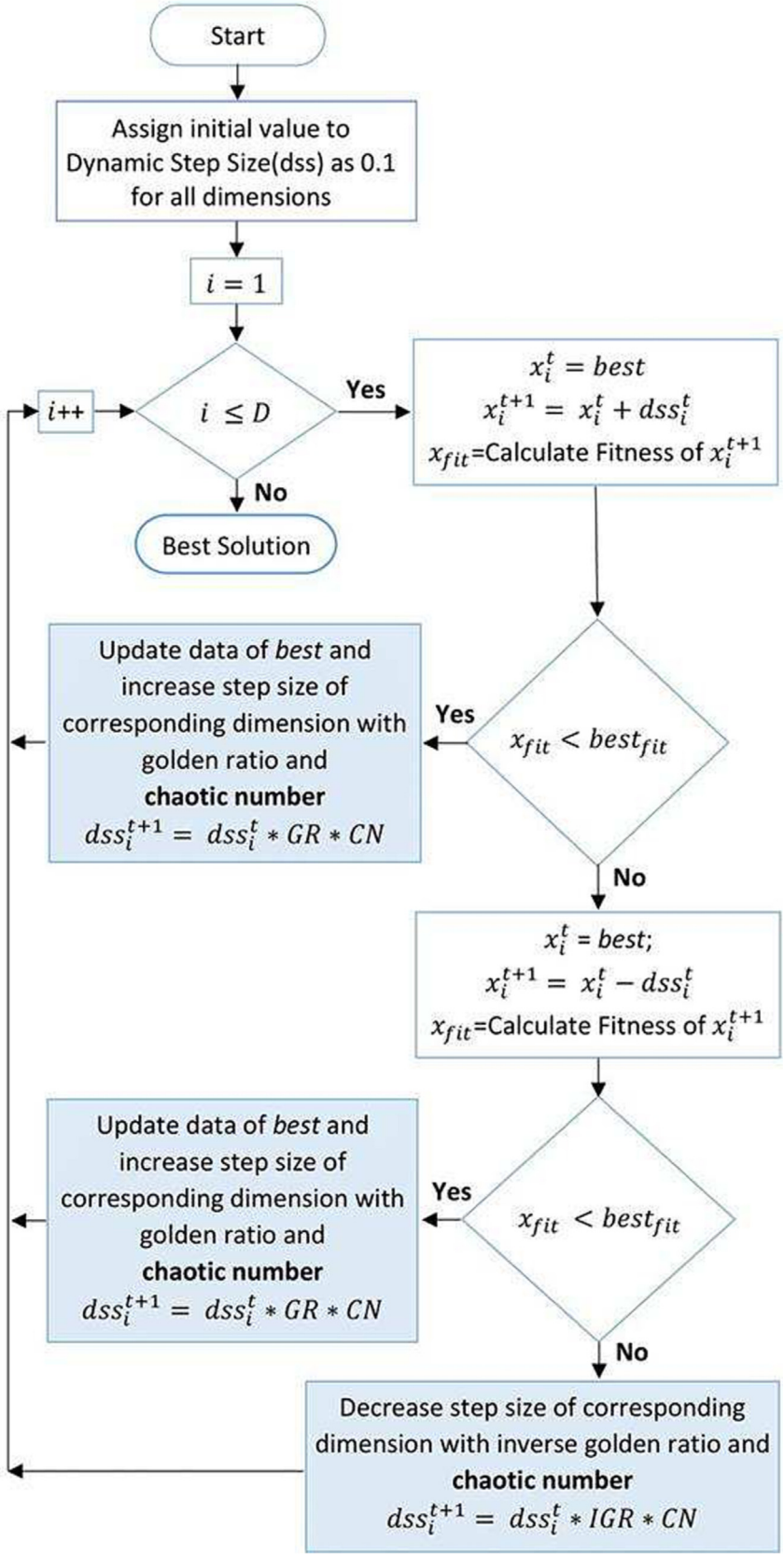

Fig. 2. Flowchart of the CGRGLS.

was categorized as a big data problem which is complex and challenging [57,58].The main purpose of the Big-Opt problem is to reach the correct EEG signals by minimizing the artifacts for every second measurement from the device. Therefore, the problem tries

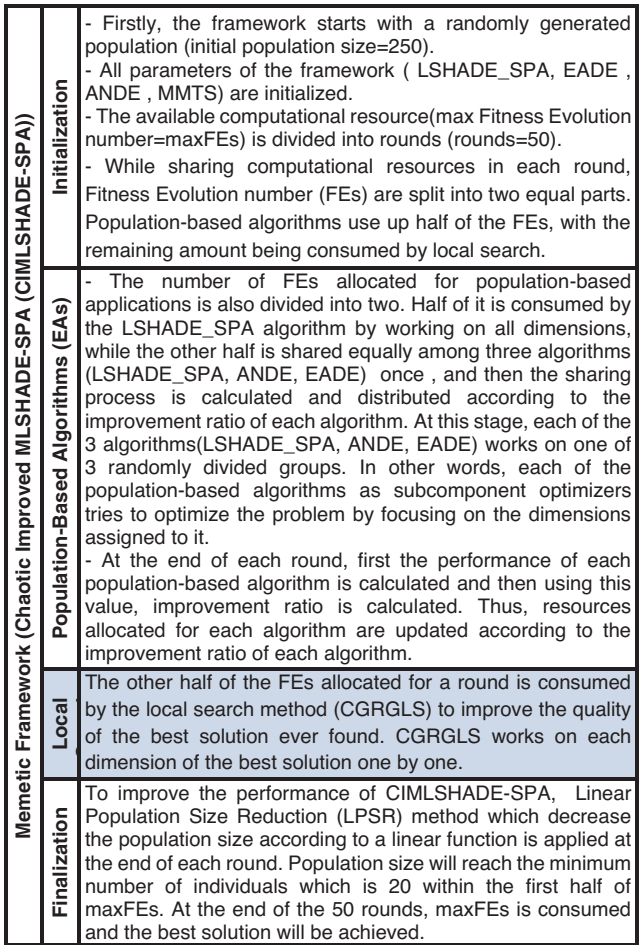

Fig. 3. Structure of Chaotic Improved MLSHADE-SPA.

to separate the obtained EEG signal into two components, one of which focuses on getting the most similar to the original signal, while the other tries to remove artifacts as much as possible.

Goh et al. took 256 samples over 4, 12 and 19 interdependent time series, taking one second of measurements and recording the measurement results. Thus, they generated synthetic data from the measurements of the EEG signals and separated data as six different problem instances. Six datasets which have different sizes and difficulties in terms of the number of local signals, artifact sources and noise level are as shown in Table 2. While the names D4, D12 and D19 are used for the default state of recorded one second measurements by referencing the number of interdependent time series, the names D4N, D12N and D19N are used for the state of added a certain amount of artifact to the default state. The number of variables in the problem is defined by multiplying the number of time series and the length of EEG signals. For each time series, the length is determined as 256 so for example, if the dataset has 4 time series, this will define a 1024 variables optimization problem. Therefore D4, D12 and D19 problems have 1024, 3072, and 4864 variables respectively. This benchmark set consists of real-world problems, provides a larger number of different datasets than those used in LSGO benchmarks, and contains noise and noiseless problems that are not common in the LSGO domain [59].

Let N indicates the number of inter-dependent time series, M represents the number of samples obtained, and NxM represent the size of the X matrix which store the results of measurement. A linear transformation matrix and S signal matrix consisting of N rows and N columns, X matrix can be defined as follows:

$$X = A * S \tag{1}$$

The problem is to divide the matrix S which represents EEG signal into two matrices, S1 and S2, each of which has the same number of dimensions as S [58]. The S1 matrix represents the noise-separated part of the S matrix, and the S2 matrix represents the noise-correlated part of the S matrix [34].

$$S = S1 + S2 \tag{2}$$

$$X = A * S1 + A * S2 \tag{3}$$

Goh et al. determined that Pearson correlation coefficients calculated as in equation (4) by using X, A and S1 matrices can be used to divide the S matrix into $S1$ and $S2$ matrices in the most appropriate way. In Eq.4, C represents the Pearson correlation coefficient between X and A × S1,covar(.) represents the covariance matrix, and σ(.) represents the variance. Maximizing C's diagonal elements is the goal, while reducing off-diagonal elements to zeros. The difference between S and S1 should also be as small as possible; in other words, S1 should be as identical to S as possible [58].

$$C = \frac{covar(X, AxS_1)}{\sigma(X)x\sigma(AxS_1)} \tag{4}$$

The objective functions aimed at determining the quality of the signal segment are defined as follows.[57,58]

$$\min f_1 = \frac{1}{N^2 - N}\sum_i \sum_{j \neq i} C_{ij}^2 + \frac{1}{N}\sum_i (1 - C_{ii})^2 \tag{5}$$

$$\min f_2 = \frac{1}{NXM}\sum_i \sum_j (S_{ij} - S_{1ij})^2 \tag{6}$$

$$\min F = f1 + f2 \tag{7}$$

This benchmark has a multi-objective version and a single objective version. While multi-objective version is related with the minimization and maximization properties of the off-diagonal and diagonal elements of the C matrix, single objective version is used to evaluate how the S1 matrix is similar to the S matrix [58,60]. The single objective which is studied in this paper occurs by combining linearly the results of two fitness functions.

Table 2
Problem Details of Big-Opt Benchmark Set.

Dataset Instance	Number of Artifact	Number of Source	Data Length	Size	Noise
D4	2	4	256	1024	No
D4N	2	4	256	1024	Yes
D12	6	12	256	3072	No
D12N	6	12	256	3072	Yes
D19	6	19	256	4864	No
D19N	6	19	256	4864	Yes

4.3. Experimental results

GRGLS was improved using chaotic step size and the performance of proposed CGRGLS was tested with Big-Opt Problems in MLSHADE-SPA which is a memetic framework.

Within the scope of the study, four performance analysis experiments were performed to verify the efficiency of the proposed CGRGLS which is a local search method.

These performance analysis experiments can be listed as follow:

1. Choosing the most successful chaotic map by comparing 10 different chaotic maps that are well known in the literature.
2. Comparison of the data obtained from running the algorithms created by using MMTS, GRGLS and the proposed method (CGRGLS), one by one as a local search method within the same memetic framework, under the same conditions.
3. To show the performance of a memetic framework which use proposed CGRGLS as local search method over Big-Opt problems by comparing with other algorithms.
4. The success of the proposed method was verified by comparing the performance with the winners of the LSGO Competitions of CEC.

4.3.1. Chaotic map selection

When publications about chaotic maps are examined, it is not easy to identify which chaotic map will perform best. For this reason, 10 different chaotic maps, which are well known in the literature and shown in Table 1, were tested separately, and the most suitable chaotic map that would increase the performance of the method was tried to be determined.

Experimental results were obtained from the mean of 25 independent runs. As the stopping criterion, 50.000 Fes (Fitness Evolution) for D4 and D4N which has 1024 variables, 150.000 FEs for D12 and D12N which has 3072 variables, 250.000 FEs for D19 and D19N which has 4864 variables. Table 3 lists the results of the tests performed using each map separately. The results of 25 independent runs are shown as std, mean, and best values. The best average value obtained in each function is marked with a bold style. At the bottom of the table, there are the overall total ranking values.

The results in Table 3 show that the algorithm using Map8 achieves the best results among the ten maps. It is seen that 4 out of 6 functions obtained the best value, in other word, map8 is successful in 67 % of the functions. The most successful maps after Map8 are Map2 and Map10, with the best value in only one out of the entire dataset. We understand that since map8 has better values in most of the functions, it creates a better exploration–exploitation balance than other methods, so it is more performance. Therefore, it was decided to use the numbers produced by the Singer Chaotic Map (Map8) as a multiplier within the proposed local search method.

Chaotic maps are mostly used as random number generators in metaheuristic algorithms. In the proposed first version of the local search method, named as GRGLS, the step size change was made with the golden ratio (GR = 1.618) or inverse golden ratio (IGR = 0.618) according to the conditions. In new proposed version of the local search method, named as CGRGLS, by using the value obtained from singer chaotic map as another multiplier besides GR and IGR in the equation, it is ensured that changing the motion step size corresponding to each dimension that play an important role in the performance of the local search method is affected chaotically.

4.3.2. Comparison of local search methods

When the memetic algorithms used in the Large-Scale Global Optimization (LSGO) field are examined, MTS-LS1 came into prominence and it is seen that it and its variants are preferred in many algorithms as local search method [26,50,51,61–64]. This technique was proposed in Multiple Trajectory Search for LSGO (MTS) by Tseng and Chen in 2008 [26]. In this algorithm, 3 different local search methods are used but MTS-LS1 is the most effective one for LSGO among these methods. This local search method, which strikes a fair balance between exploration and exploitation, makes it possible to construct powerful and effective algorithms to deal with the difficulties of LSGO.

In order to test the performance of CGRGLS, the local search method proposed in this publication, MMTS and GRGLS methods which were used as local search methods within the MSHADE SPA, were chosen. The common point of the three methods is that they are the suggested methods inspired by the MTS-LS1 method. MMTS was used in MLSHADE-SPA algorithm as a local search method by Hadi et al. in 2019 [51]. In this method, the difference between the current minimum and maximum values of each dimension is multiplied by a random number in the range [0,1], and this yields the initial step values for each dimension. In order to obtain a better new solution by improving the existing solution, opposite points are evaluated in each dimension as much as the step size of that dimension. As long as the result is successful, the step size is increased linearly.

GRGLS was proposed in MLSHADE-SPA algorithm as a local search method by Koçer and Uymaz 2021 [50]. In this method, there is an initial step size that is 0.1 corresponds for each dimension of problem. The step size of each dimension varies throughout the search process depending on how the best individual's fitness value improves. Opposite points are evaluated in each dimension as much as the step size of that dimension as in the other method MMTS. The adaptive step size is changed using a coefficient based on the golden ratio and inverse golden ratio unlike MMTS. Although the CGRGLS method works the similar as the GRGLS local search method, there is a difference in adaptive step size change. Step size changes by multiplying by the number produced by the singer chaotic map in addition to the existing multipliers which are golden ratio or inverse golden ratio.

As a result of the map selection tests explained in detail in the previous section, it has been determined that the chaotic map that provides the most effective and efficient contribution to the proposed method is Singer. CGRGLS, which was developed with the effect of Singer chaotic map used as a random number generator, was compared with the other two local search strategies. The parameter values and the termination conditions selected same as Chaotic Map Selection part.

Table 4 lists standard deviations, mean, best and time values of the tests results. According to the Table 4, proposed local search method achieved the first place by obtaining the best results in four of the six data sets (D12, D12N, D19, D19N), while the MMTS method achieved the best results in the D4 and D4N data sets and became the second method. However, the GRLGS algorithm, which is the previous version of the proposed method, could not obtain the best value in any data set. This status proves that proposed CGRGLS method effectively improves the previous version of itself. The strong performance of the CGRGLS algorithm is the result of the ergodicity of chaotic maps and an effective exploration of the solution space. Also, it is seen that the local search methods close to each other in terms of time and the developed method has slightly better results than the others.

Optimization methods show unique search behaviors while searching for the optimum solution. Convergence curves are the most important tool in revealing the solution-seeking behavior of an optimization method. Convergence curves are generated based

Table 3
Results of Chaotic Maps Comparison.

		Map1	Map2	Map3	Map4	Map5	Map6	Map7	Map8	Map9	Map10
D4	Best	6.895E-02	6.973E-02	6.824E-02	6.814E-02	6.953E-02	6.960E-02	6.795E-02	6.831E-02	6.741E-02	6.901E-02
	Mean	7.315E-02	7.492E-02	7.208E-02	7.310E-02	7.209E-02	7.333E-02	7.299E-02	**7.168E-02**	7.261E-02	7.358E-02
	Std.	2.980E-03	7.918E-03	3.018E-03	3.975E-03	1.500E-03	3.939E-03	3.530E-03	2.413E-03	3.553E-03	5.764E-03
	Rank	7	10	2	6	3	8	5	1	4	9
D4N	Best	6.682E-02	6.647E-02	6.648E-02	6.697E-02	6.578E-02	6.682E-02	6.642E-02	6.675E-02	6.592E-02	6.727E-02
	Mean	7.066E-02	7.099E-02	7.088E-02	7.105E-02	7.135E-02	7.233E-02	7.130E-02	**6.948E-02**	6.961E-02	7.058E-02
	Std.	3.488E-03	3.645E-03	3.952E-03	4.380E-03	3.761E-03	4.065E-03	3.413E-03	2.373E-03	2.267E-03	3.021E-03
	Rank	4	6	5	7	9	10	8	**1**	2	3
D12	Best	3.601E-02	3.776E-02	4.020E-02	4.094E-02	3.569E-02	4.632E-02	3.589E-02	3.408E-02	3.601E-02	3.924E-02
	Mean	5.392E-02	5.544E-02	5.908E-02	5.736E-02	5.959E-02	5.713E-02	5.341E-02	**5.287E-02**	5.749E-02	5.836E-02
	Std.	1.025E-02	1.010E-02	1.373E-02	1.005E-02	1.061E-02	7.781E-03	8.253E-03	1.009E-02	1.344E-02	9.868E-03
	Rank	3	4	9	6	10	5	2	**1**	7	8
D12N	Best	4.278E-02	4.344E-02	3.867E-02	4.285E-02	4.395E-02	4.151E-02	4.170E-02	3.417E-02	4.111E-02	3.878E-02
	Mean	5.658E-02	5.696E-02	5.451E-02	5.694E-02	5.855E-02	5.943E-02	5.507E-02	5.606E-02	5.653E-02	**5.280E-02**
	Std.	8.928E-03	9.255E-03	8.064E-03	1.066E-02	1.130E-02	9.942E-03	8.839E-03	1.161E-02	1.021E-02	1.046E-02
	Rank	6	8	2	7	9	10	3	4	5	**1**
D19	Best	4.844E-02	5.093E-02	4.790E-02	4.830E-02	4.800E-02	4.451E-02	4.846E-02	4.743E-02	4.793E-02	4.987E-02
	Mean	6.314E-02	**6.223E-02**	6.276E-02	6.293E-02	6.384E-02	6.223E-02	6.238E-02	6.298E-02	6.256E-02	6.277E-02
	Std.	8.622E-03	9.358E-03	8.925E-03	8.969E-03	9.451E-03	9.304E-03	7.469E-03	1.086E-02	9.130E-03	9.808E-03
	Rank	9	**1**	5	7	10	2	3	8	4	6
D19N	Best	4.642E-02	4.412E-02	5.230E-02	4.612E-02	4.992E-02	4.738E-02	4.700E-02	4.587E-02	4.678E-02	4.755E-02
	Mean	6.340E-02	6.210E-02	6.333E-02	6.228E-02	6.219E-02	6.215E-02	6.221E-02	**6.046E-02**	6.178E-02	6.169E-02
	Std.	7.960E-03	9.513E-03	8.579E-03	9.655E-03	8.541E-03	9.583E-03	9.081E-03	8.496E-03	8.875E-03	9.267E-03
	Rank	10	4	9	8	6	5	7	**1**	3	2
Best count		0	1	0	0	0	0	0	**4**	0	1
Total Rank		39	33	32	41	47	40	28	**16**	25	29

Table 4
Comparison results of CGRGLS with other local search methods.

	MMTS				**GRGLS**				**CGRGLS**			
	Best	Mean	Std	*Time*	Best	Mean	Std	*Time*	Best	Mean	Std	*Time*
D4	6.683E-02	**7.052E-02**	2.119E-03	*1.87E + 01*	9.376E-01	1.023E + 00	5.247E-02	*1.96E + 01*	6.831E-02	7.168E-02	2.413E-03	*1.86E + 01*
D4N	6.559E-02	**6.817E-02**	1.347E-03	*2.06E + 01*	8.651E-01	1.031E + 00	8.335E-02	*2.16E + 01*	6.675E-02	6.948E-02	2.373E-03	*1.86E + 01*
D12	5.097E-02	5.975E-02	3.771E-03	*1.30E + 03*	5.299E-01	5.798E-01	3.430E-02	*1.26E + 03*	3.408E-02	**5.287E-02**	1.009E-02	*1.25E + 03*
D12N	5.328E-02	5.953E-02	4.309E-03	*1.32E + 03*	5.228E-01	6.020E-01	3.990E-02	*1.31E + 03*	3.417E-02	**5.606E-02**	1.161E-02	*1.28E + 03*
D19	7.653E-02	8.520E-02	5.668E-03	*3.05E + 03*	3.082E-01	3.611E-01	2.347E-02	*3.10E + 03*	4.743E-02	**6.298E-02**	1.086E-02	*3.04E + 03*
D19N	7.743E-02	8.711E-02	5.848E-03	*3.12E + 03*	3.118E-01	3.687E-01	2.469E-02	*3.11E + 03*	4.587E-02	**6.046E-02**	8.496E-03	*3.10E + 03*

on the best solution obtained by the method throughout the iterations. The convergence curves obtained by three various local search techniques in six well-known Big-Opt Benchmark Functions were produced as a result of the experimental study. These convergence curves are shown in Fig. 4.

As seen in Fig. 4, CGRGLS and MMTS methods obtain close results on 1024 dimensional D4 and D4N than GRLGS method. As the problem size increases, the convergence curves of D12 and D19 show that the chaotic approach increases the success of the local search method, and this situation supports the results in Table 4. The success of the chaotic method increases with the size of the problem. It also has a regular improvement as iterations progress. In other words, when the curves on D19 and D19N are examined, it can be concluded that if more iterations are run, CGRGLS will reach a much better solutions than the other methods.

The MMTS method performs better than the GRGLS algorithm which is the previous version of the proposed method. The previous version, GRGLS, converged early and stuck to local minima and could not form a significant curve in any dataset. The performance of GRGLS and CGRGLS shows that algorithms with local search strategies perform much more successfully when powered by chaotic maps.

The Wilcoxon signed-rank test was used to analyze the problem-solving performance of the local search methods. The test was conducted by using the global minimum values obtained as a result of 25 runs for problem-based pairwise comparison of the algorithms. Table 5 shows statistical comparisons by the Wilcoxon signed-rank test between CGRGLS and other local search methods. Table 5 present the p-Value, T, and W values. If the W value is "=", there is no statistical difference (equal), if it is "+", there is a meaningful difference between the results and the CGRGLS method is more successful than the other method (win), if it is "-", there is a meaningful difference between the results and the other method is more successful (lose). From the last line of Table 5, total win, equal, and lose values are given for the two methods compared as a result of the Wilcoxon signed-rank test. Table 5 shows that CGRGLS can achieve statistically better results than the comparison algorithms, with a level of significance a = 0.05. The advantage of the CGRGLS seems better as the fitness evaluation number increases.

4.3.3. Comparison with other algorithms

In this paper, MLSHADE-SPA is improved by replacing its local search method (MMTS) with CGRGLS and it is named Chaotic Improved MLSHADE-SPA (CIMLSHADE-SPA). In order to analyze whether the performance of the proposed algorithm is sufficient compared to other evolutionary or swarm intelligence-based optimization algorithms, the results obtained are compared with the results of 5 algorithms directly taken from [36]. The average results were obtained after CIMLSHADE-SPA was performed 30 runs for each problem for 100.000 fitness evaluations. Comparison results are shown in Table 6. The results obtained from CIMLSHADE-SPA are compared with the other algorithms in literature. These algorithms are as follows:

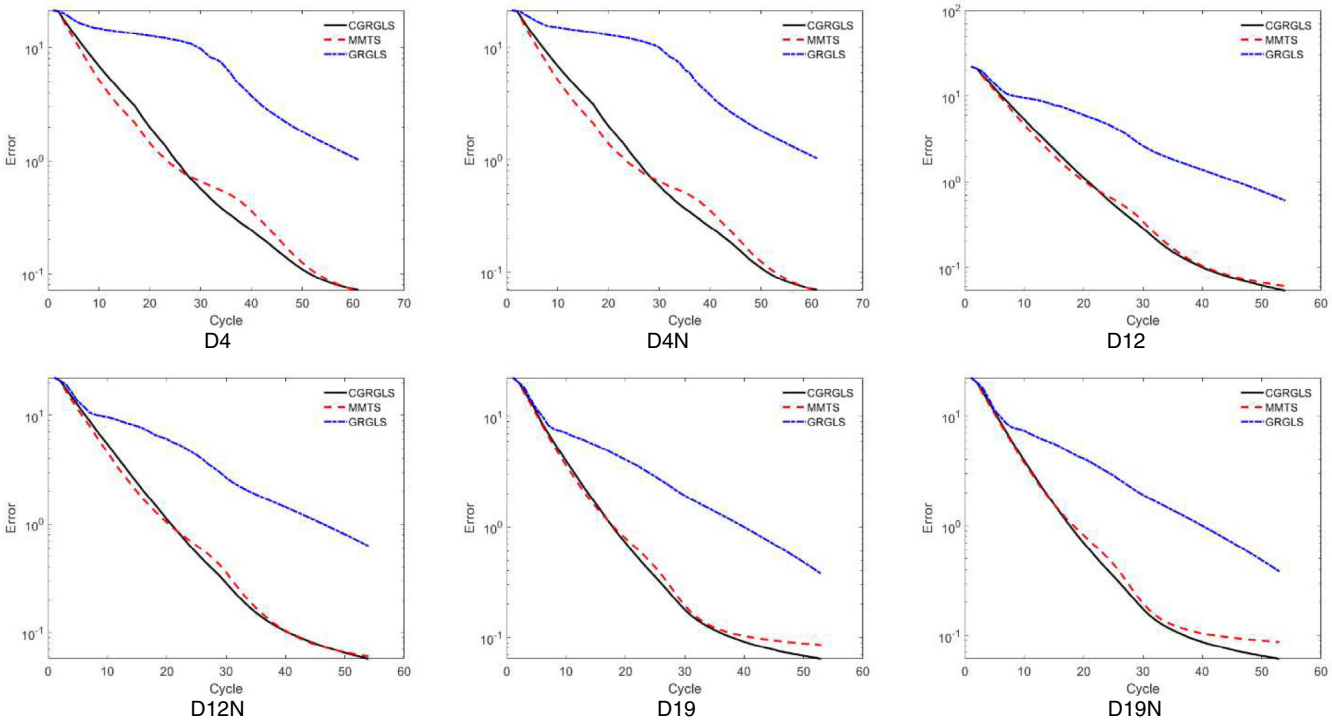

Fig. 4. Convergence curves of Local Search Methods on Big-Opt Problems.

Table 5
The results of two-sided Wilcoxon signed-rank test for local search methods.

Func.	CGRGLS versus GRGLS			CGRGLS versus MMTS		
	p Value	*T*	*W*	*p Value*	*T*	*W*
D4	1,23E-05	0	+	1,50E-01	216	=
D4N	1,23E-05	0	+	8,71E-03	260	–
D12	1,23E-05	0	+	6,31E-03	61	+
D12N	1,23E-05	0	+	1,15E-01	104	=
D19	1,23E-05	0	+	1,23E-05	0	+
D19N	1,23E-05	0	+	1,23E-05	0	+
	+ / = / –	**6/0/0**		**+ / = / –**	**3/2/1**	

1. COABC: Converge-onlookers ABC is a modified version of the ABC algorithm which includes a new solution search equation in the onlooker stage [65].

2. ACDE: Automatic Configurations of DEs is a memetic framework which combines DE variants as population-based global search and the interior point method as local search algorithm [4].

3. SaNSDE-CC: Self-Adaptive Differential Evolution with Neighborhood Search with Cooperative Co-evolution is an improved DE variant [66].

4. JADE is a variant of DE algorithm, which implements a mutation strategy "DE/current-to-pbest" with optional archive and controls F and C R in an adaptive manner [67].

5. SHADE: Success-history-based adaptive DE is an adaptive differential algorithm which uses a history-based parameter adaptation scheme [68].

When Table 6 is examined, CIMLSHADE-SPA, COABC and ACDE methods are close to each other and give more successful results than the others. CIMLSHADE-SPA gives the best results on D4 and D4N, COABC D19 and ACDE D12, D12N and D19N data. CIMLSHADE-SPA method was one of the most successful methods on all data. Total rank values prove that these 3 methods are far more successful than the others. CIMLSHADE-SPA shows that it is a method that gives consistent results by giving the best or near-best values.

The Friedman test is a non-parametric statistical test used to test for differences between multiple variables measured in the same sample. It is preferred when the assumption of normal distribution is not met in small samples and produces results using the ordering of the data. In Table 6 and Table 7, the results obtained by other algorithms are obtained from related publications, and only the average of the results is available instead of each run result. For this reason, Friedman test was applied to show whether there is a significant difference between the listed results. The significance level is set at 0.05 for this statistical test. If the p-value is less than 0.05, it means that there is a statistically significant difference between all results obtained. Otherwise, there is no significant difference. It is seen in Table 6 that the p-value obtained from the Friedman test is smaller than the level of significance. It shows that there is a statistically significant difference between all results obtained.

4.3.4. Comparison with large scale competition winners

LSGO is a term for a special category of global optimization problems involving a large number of decision variables. In LSGO, both the modeling and the solution process can become quite com-

Table 6
Comparison results of CIMLSHADE-SPA with other algorithms.

		CIMLSHADE-SPA	COABC	ACDE	SaNSDE-CC	JADE	SHADE
D4	Mean	**6.14E-02**	6.35e − 02	6.55e − 02	5.11E + 00	2.70E + 00	3.22E + 00
	Best	6.11E-02	6.29e − 02	6.13e − 02	3.28E + 00	1.81E + 00	1.42E + 00
	Std.	1.51E-04	2.80e − 04	3.01e − 06	1.57e − 01	4.76e − 01	7.23e − 01
	Rank	***1***	***2***	***3***	***6***	***4***	***5***
D4N	Mean	**5.91E-02**	6.14e − 02	5.93e − 02	1.11E + 01	9.45E + 00	4.19E + 00
	Best	5.90E-02	6.10e − 02	5.93e − 02	1.79E + 00	1.02E + 00	2.11E + 00
	Std.	8.09E-05	2.62e − 04	3.51e − 06	2.11E + 00	9.62e − 01	1.41e − 01
	Rank	***1***	***3***	***2***	***6***	***5***	***4***
D12	Mean	2.72E-01	6.26e − 02	**5.12e − 02**	6.22E + 00	8.05E + 00	5.16E + 00
	Best	1.49E-01	5.26e − 02	3.61e − 02	4.22E + 00	4.12E + 00	2.64E + 00
	Std.	5.09E-02	4.62e − 03	9.60e − 03	1.02e − 01	2.73E + 00	1.84e − 01
	Rank	***3***	***2***	***1***	***5***	***6***	***4***
D12N	Mean	2.56E-01	6.21e − 02	**5.09e − 02**	1.27E + 01	1.38E + 01	7.73E + 00
	Best	1.68E-01	5.48e − 02	3.63e − 02	5.11E + 00	7.28E + 00	3.17E + 00
	Std.	5.13E-02	4.54e − 03	1.11e − 02	1.13E + 00	1.09E + 00	1.27e − 01
	Rank	***3***	***2***	***1***	***5***	***6***	***4***
D19	Mean	1.07E + 00	**1.58e − 01**	2.15e − 01	1.97E + 02	1.08E + 01	9.10E + 00
	Best	8.41E-01	1.40e − 01	6.28e − 02	1.23E + 02	5.17E + 00	6.17E + 00
	Std.	1.24E-01	1.08e − 02	7.49-e02	2.32E + 01	2.30E + 00	1.76e − 01
	Rank	***3***	***1***	***2***	***6***	***5***	***4***
D19N	Mean	1.15E + 00	1.59e − 01	**1.36e − 01**	1.86E + 02	1.67E + 01	8.26E + 00
	Best	7.41E-01	1.40e − 01	7.02e − 02	1.47E + 02	9.72E + 00	4.42E + 00
	Std.	1.90E-01	8.88e − 03	5.16e − 02	1.26E + 01	1.28E + 00	1.282e − 01
	Rank	***3***	***2***	***1***	***6***	***5***	***4***
	Total Rank	**14**	**12**	**10**	**34**	**31**	**25**
Friedman's Test	**Mean Rank**	2.333333	2	1.666667	5.666667	5.166667	4.166667
	p–Value	**1.10E-04**					

Table 7
Comparison results of CIMLSHADE-SPA with CEC Competition Winners.

		MOS-2011	MOS-2013	MAGA	SHADE-ILS	MLSHADE-SPA	CIMLSHADE-SPA
D4		0.06103	0.06103	0.0610	0.06103	0.06103	0.06103
D4N		0.05897	0.05897	0.0590	0.05897	0.05897	0.05897
D12		0.00198	0.00194	0.0019	0.00194	0.00194	0.00194
D12N		0.00188	0.00183	0.0018	0.00183	0.00183	0.00183
D19		0.08940	0.00251	0.0025	0.00252	0.00252	0.00252
D19N		0.09180	0.00256	0.0026	0.00256	0.00256	0.00256
Friedman's Test	**Mean Rank**	5.166667	3.083333	2.5	3.416667	3.416667	3.416667
	p–Value	**6.19E-02**					

plex and difficult due to some factors caused by the increase in the number of variables. These factors can be listed as exponential increase in the search area with the increase in size, the complexity and properties of some functions may change, the increase in the number of interdependent variables, etc. For these reasons, algorithms with high performance in normal size may lose their effectiveness as the size grows. Researchers are trying to develop various algorithms and techniques to overcome these problems. IEEE Congress of Evolutionary Computation (CEC) has been organizing competitions and special sessions in this field since 2008 to encourage and motivate researchers. A detailed information about CEC special sessions and competitions of LSGO can be found in [69].

Big-Opt problems which are used as a benchmark set in this paper have a huge number of decision variables, are nonlinear and require real-time handling of EEG signals, so they can be categorized as a LSGO problems which are complex and challenging. Therefore, to show the sufficiency of the proposed method, it will be an effective indicator to compare it with the winners of the LSGO Competitions of CEC.

To ensure a fair comparison, the parameter values and the termination condition of winners are fixed with the same values as in [59].The average results were obtained after CIMLSHADE-SPA was performed 10 times for 1.000.000 fitness evaluations of each task. The results are compared with the results of 5 winners directly taken from [59] and comparison results are shown in Table 7.

MOS-2011 is the winner of SOCO 2011 Special Issue by using a benchmark created especially for this competition [70]. MOS-2013 and its variants are the winners of CEC'2012, CEC'2013, CEC'2015 for large-scale global optimization [71]. In 2015, a Big Data Competition was carried out and a new benchmark which is also used in this paper, for large-scale global optimization was proposed. Even though several methods were proposed to this special session there were not enough to run a real competition. MAGA (A Multi-Agent Genetic Algorithm) [39] is the algorithm that had the best results among them [69]. In 2018, SHADEILS is the winner of the CEC'2018 Competition by using CEC'2013 LSGO benchmark [64]. And MLSHADE-SPA is the runner-up in the CEC'2018 competition with better results than MOS-2013[51].

A real-world problem with a relatively small number of variables might be thought to be easier to solve, but sometimes the algorithm needs a certain number of variables to obtain enough information for identifying the correlation between signals and separating the artifacts from them. Which means that D4(1024) is the least representative of a real environment among problems. Although there are more variables to optimize in the D12 problem (3072 variables), algorithms still get the best results since more information is accessible. Although it is anticipated that the D19 problem which has 4864 variables will produce inferior results than D12 because of the required greatest domain searching, it can still be optimized more effectively than D4 because of the knowledge gleaned from a significant number of variables [39].

The results listed in Table 7 show that the results produced by the different algorithms are quite similar or even identical. This shows that the proposed CIMLSHADE-SPA algorithm has successfully reached the targeted optimum values like other LSGO winner algorithms. The fact that the p value obtained as a result of the Friedman test shown below the Table 7 is greater than the level of significance (0.05) confirms this situation.

5. Conclusion

The large search space in multidimensional optimization problems requires very efficient search strategies. In this study, non-repetitive random number generation features of chaotic maps were used to increase efficiency. Thanks to this feature, the improvement in local search methods has been observed. Especially as the problem size increases, the success of the local search method increases. Studies and comparisons have shown that using the value obtained by chaotic mapping as a coefficient in the step size change part of the algorithm, which is one of the most important factors of the algorithm, is an update that increases the efficiency of the algorithm.

For future studies, the population initialization technique which was introduced by Majdouli et al. can be used in the CIMLSHADE-SPA algorithm and the effect of the technique on performance can be investigated. This technique uses data directly from the original EEG signal rather than randomly initializing parameters of the potential feasible solutions within the specified lower/upper bounds.

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.

Acknowledgments

This work was supported by The Coordinatorship of Scientific Research Projects of Selçuk University [Grant number: 18101012].

References

[1] K. Hussain, M.N. Mohd Salleh, S. Cheng, Y. Shi, Metaheuristic research: a comprehensive survey, Artif. Intell. Rev. 52 (4) (2019) 2191–2233.
[2] J. Stork, A.E. Eiben, T. Bartz-Beielstein, A new taxonomy of global optimization algorithms, Nat. Comput. 21 (2) (2022) 219–242.
[3] Blocho, M., *Heuristics, metaheuristics, and hyperheuristics for rich vehicle routing problems.* Smart Delivery Systems: Solving Complex Vehicle Routing Problems, 2020: p. 101–156.
[4] S. Elsayed, R. Sarker, An adaptive configuration of differential evolution algorithms for big data, IEEE Congr. Evolut. Comput. (Cec) 2015 (2015) 695–702.
[5] S. Elsayed, R. Sarker, Differential evolution framework for big data optimization, Memetic Comput. 8 (1) (2016) 17–33.
[6] Sheikholeslami, R., Kaveh, A., *A survey of chaos embedded meta-heuristic algorithms.* 2013. **3**(4): p. 617-633.
[7] L.J. Yang, T.L. Chen, Application of chaos in genetic algorithms, Commun. Theor. Phys. 38 (2) (2002) 168–172.
[8] Y.Ç. Kuyu, F. Vatansever, The chaos-based approaches for actual metaheuristic algorithms, Uludağ Üniversitesi Mühendislik Fakültesi Dergisi 23 (2018) 103–116.
[9] L. Wang, Y. Zhong, Cuckoo search algorithm with chaotic maps, Math. Probl. Eng. 2015 (2015) 1–14.
[10] V. Jothiprakash, R. Arunkumar, Optimization of hydropower reservoir using evolutionary algorithms coupled with chaos, Water Resour. Manag. 27 (7) (2013) 1963–1979.
[11] Z.Y. Guo et al., Self-adaptive chaos differential evolution, Adv. Nat. Comput., Pt 1 (4221) (2006) 972–975.
[12] P. Snaselova, F. Zboril, Genetic algorithm using theory of chaos, Procedia Comput. Sci. 51 (2015) 316–325.
[13] A.H. Gandomi, X.-S. Yang, S. Talatahari, A.H. Alavi, Firefly algorithm with chaos, Commun. Nonlinear Sci. Numer. Simul. 18 (1) (2013) 89–98.
[14] S. Saremi, S.M. Mirjalili, S. Mirjalili, Chaotic krill herd optimization algorithm, Procedia Technol. 12 (2014) 180–185.
[15] A.H. Gandomi, G.J. Yun, X.-S. Yang, S. Talatahari, Chaos-enhanced accelerated particle swarm optimization, Commun. Nonlinear Sci. Numer. Simul. 18 (2) (2013) 327–340.
[16] S. Arora, S. Singh, An improved butterfly optimization algorithm with chaos, J. Intell. Fuzzy Syst. 32 (1) (2017) 1079–1088.
[17] M. Kohli, S. Arora, Chaotic grey wolf optimization algorithm for constrained optimization problems, J. Comput. Des. Eng. 5 (4) (2018) 458–472.
[18] S. Arora, P. Anand, Chaotic grasshopper optimization algorithm for global optimization, Neural Comput. & Applic. 31 (8) (2019) 4385–4405.
[19] M. Mitić, N. Vuković, M. Petrović, Z. Miljković, Chaotic fruit fly optimization algorithm, Knowl.-Based Syst. 89 (2015) 446–458.
[20] Y. Wang et al., A gravitational search algorithm with hierarchy and distributed framework, Knowl.-Based Syst. 218 (2021) 106877.
[21] Y. Wang, S. Gao, M. Zhou, Y. Yu, A multi-layered gravitational search algorithm for function optimization and real-world problems, IEEE/CAA J. Autom. Sin. 8 (1) (2021) 94–109.
[22] S. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng, M. Zhou, Chaotic local search-based differential evolution algorithms for optimization, IEEE Trans. Syst., Man, Cybernetics: Syst. 51 (6) (2021) 3954–3967.
[23] A.M. Machado et al., A new hybrid matheuristic of GRASP and VNS based on constructive heuristics, set-covering and set-partitioning formulations applied to the capacitated vehicle routing problem, Expert Syst. Appl. 184 (2021) 115556.
[24] D. Jia, G. Zheng, M.K. Khan, An effective memetic differential evolution algorithm based on chaotic local search, Inf. Sci. 181 (15) (2011) 3175–3187.
[25] J.A. Koupaei, S.M.M. Hosseini, F.M. Ghaini, A new optimization algorithm based on chaotic maps and golden section search method, Eng. Appl. Artif. Intel. 50 (2016) 201–214.
[26] Tseng, L.Y., C. Chen, *Multiple Trajectory Search for Large Scale Global Optimization.* 2008 Ieee Congress on Evolutionary Computation, Vols 1-8, 2008: p. 3052-+.
[27] El Majdouli, M.A., et al., *A Fireworks Algorithm for Single Objective Big Optimization of Signals.* 2016 Ieee/Acs 13th International Conference of Computer Systems and Applications (Aiccsa), 2016.
[28] M.A. El Majdouli, I. Rbouh, S. Bougrine, B. El Benani, A.A. El Imrani, Fireworks algorithm framework for Big Data optimization, Memetic Comput. 8 (4) (2016) 333–347.
[29] Loukdache, A., et al., *A Clonal Selection Algorithm for the Electro Encephalography Signals Reconstruction.* Proceedings of 2017 International Conference on Electrical and Information Technologies (Iceit 2017), 2017.
[30] Celil, S., S. Aslan, and S. Demirci, *A Novel Harmony Search Based Method for Noise Minimization on EEG Signals*, in *2021 6th International Conference on Computer Science and Engineering (UBMK)*. 2021. p. 747-750.
[31] N.R. Sabar, J. Abawajy, J. Yearwood, Heterogeneous cooperative co-evolution memetic differential evolution algorithm for big data optimization problems, IEEE Trans. Evol. Comput. 21 (2) (2017) 315–327.
[32] Meselhi, M.A., et al., *Fast Differential Evolution for Big Optimization.* 2017 11th International Conference on Software, Knowledge, Information Management and Applications (Skima), 2017.
[33] Meselhi, M.A., et al., *Parallel Evolutionary Algorithm for EEG Optimization Problems.* 2021 Ieee Congress on Evolutionary Computation (Cec 2021), 2021: p. 2577-2584.
[34] Arslan, S. and S. Aslan, *A new lattice based artificial bee colony algorithm for EEG noise minimization.* Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, 2022.
[35] Aslan, S. and D. Karaboga, *A genetic Artificial Bee Colony algorithm for signal reconstruction based big data optimization.* Applied Soft Computing, 2020. **88**.
[36] S. Aslan, A comparative study between artificial bee colony (ABC) algorithm and its variants on big data optimization, Memetic Comput. 12 (2) (2020) 129–150.
[37] Z.J. Cao et al., A phase based optimization algorithm for big optimization problems, IEEE Congress Evolut. Comput. (Cec) 2016 (2016) 5209–5214.
[38] S. Aslan, S. Demirci, An improved immune plasma algorithm with a regional pandemic restriction, SIViP 16 (8) (2022) 2093–2101.
[39] Y.T. Zhang et al., A multi-agent genetic algorithm for big optimization problems, Ieee Congress on Evolut. Comput. (Cec) 2015 (2015) 703–707.
[40] Turkoglu, B., S.A. Uymaz, and E. Kaya, *Chapter 1 - Chaos theory in metaheuristics*, in *Comprehensive Metaheuristics*, S. Mirjalili and A.H. Gandomi, Editors. 2023, Academic Press. p. 1-20
[41] R. Tang, S. Fong, N. Dey, Metaheuristics and chaos theory, in: K.A.M.A. Naimee (Ed.), Chaos Theory, InTech, 2018.
[42] A.R. Jordehi, Chaotic bat swarm optimisation (CBSO), Appl. Soft Comput. 26 (2015) 523–530.
[43] G. Kaur, S. Arora, Chaotic whale optimization algorithm, J. Comput. Des. Eng. 5 (3) (2018) 275–284.
[44] S.A. Rather, P.S. Bala, *Swarm-based chaotic gravitational search algorithm for solving mechanical engineering design problems*, World J. Eng. 17 (1) (2020) 97–114.
[45] F.B. Demir, T. Tuncer, A.F. Kocamaz, A chaotic optimization method based on logistic-sine map for numerical function optimization, Neural Comput. Appl. 32 (17) (2020) 14227–14239.
[46] G.-G. Wang, L. Guo, A.H. Gandomi, G.-S. Hao, H. Wang, Chaotic krill herd algorithm, Inf. Sci. 274 (2014) 17–34.

[47] S. Saremi, S. Mirjalili, A. Lewis, Biogeography-based optimisation with chaos, Neural Comput. Appl. 25 (5) (2014) 1077–1097.
[48] S. Talatahari, B. Farahmand Azar, R. Sheikholeslami, A.H. Gandomi, Imperialist competitive algorithm combined with chaos for global optimization, Commun. Nonlinear Sci. Numer. Simul. 17 (3) (2012) 1312–1319.
[49] R.M. May, Simple mathematical models with very complicated dynamics, in: B.R. Hunt, T.-Y. Li, J.A. Kennedy, H.E. Nusse (Eds.), The theory of chaotic attractors, Springer New York, New York, NY, 2004, pp. 85–93.
[50] H.G. Kocer, S.A. Uymaz, A novel local search method for LSGO with golden ratio and dynamic search step, Soft. Comput. 25 (3) (2021) 2115–2130.
[51] A.A. Hadi, A.W. Mohamed, K.M. Jambi, LSHADE-SPA memetic framework for solving large-scale optimization problems, Complex & Intellig. Syst. 5 (1) (2019) 25–40.
[52] Mohamed, A.W., et al. *LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems*. In: *2017 IEEE Congress on evolutionary computation (CEC)*. 2017. IEEE.
[53] A.W. Mohamed, Solving large-scale global optimization problems using enhanced adaptive differential evolution algorithm, Complex & Intelligent Syst. 3 (4) (2017) 205–231.
[54] A.W. Mohamed, A.S. Almazyad, Differential evolution with novel mutation and adaptive crossover strategies for solving large scale global optimization problems, Appl. Comput. Intelligence Soft Comput. 2017 (2017) 1–18.
[55] AW, M. *Optimization project*. 2019 5.12.2022]; Available from: https://sites.google.com/view/optimizationproject/files.
[56] S. Yildirim, H.E. Kocer, A.H. Ekmekci, Quantitative analysis of EEG slow wave activity based on minpeakprominence method, Traitement du Signal 38 (3) (2021) 757–773.
[57] S.K. Goh, H.A. Abbass, K.C. Tan, A. Al-Mamun, Decompositional independent component analysis using multi-objective optimization, Soft. Comput. 20 (4) (2016) 1289–1304.
[58] S.K. Goh et al., Evolutionary Big Optimization (BigOpt) of Signals, IEEE Congr. Evolut. Comput. (Cec) 2015 (2015) 3332–3339.
[59] D. Molina, A.R. Nesterenko, A. LaTorre, Comparing large-scale global optimization competition winners in a real-world problem, Ieee Congress Evolut. Comput. (Cec) 2019 (2019) 359–365.
[60] Goh, S.K., et al., *Artifact Removal from EEG Using a Multi-objective Independent Component Analysis Model*. Neural Information Processing (Iconip 2014), Pt I, 2014. **8834**: p. 570-577.
[61] Molina, D. and F. Herrera, *Iterative hybridization of DE with Local Search for the CEC'2015 Special Session on Large Scale Global Optimization*. 2015 Ieee Congress on Evolutionary Computation (Cec), 2015: p. 1974-1978.
[62] LaTorre, A., S. Muelas, and J.M. Pena, *Multiple Offspring Sampling In Large Scale Global Optimization*. 2012 Ieee Congress on Evolutionary Computation (Cec), 2012.
[63] S.Z. Zhao, P.N. Suganthan, S. Das, Self-adaptive differential evolution with multi-trajectory search for large-scale optimization, Soft. Comput. 15 (11) (2011) 2175–2185.
[64] D. Molina, A. LaTorre, F. Herrera, SHADE with iterative local search for large-scale global optimization, Ieee Congr. Evolut. Comput. (Cec) 2018 (2018) 1252–1259.
[65] J. Luo, Q. Wang, X.H. Xiao, A modified artificial bee colony algorithm based on converge-onlookers approach for global optimization, Appl. Math Comput. 219 (20) (2013) 10253–10262.
[66] M.N. Omidvar, X. Li, Y.i. Mei, X. Yao, Cooperative co-evolution with differential grouping for large scale optimization, IEEE Trans. Evol. Comput. 18 (3) (2014) 378–393.
[67] J.Q. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958.
[68] R. Tanabe, A. Fukunaga, Evaluating the performance of SHADE on CEC 2013 benchmark problems, Ieee Congress on Evolutionary Computation (Cec) 2013 (2013) 1952–1959.
[69] D. Molina, A. LaTorre, F. Herrera, An insight into bio-inspired and evolutionary algorithms for global optimization: review, analysis, and lessons learnt over a decade of competitions, Cogn. Comput. 10 (4) (2018) 517–544.
[70] A. LaTorre, S. Muelas, J.M. Pena, A MOS-based dynamic memetic differential evolution algorithm for continuous optimization: a scalability test, Soft. Comput. 15 (11) (2011) 2187–2199.
[71] A. LaTorre, S. Muelas, J.M. Pena, Large scale global optimization: experimental results with MOS-based hybrid algorithms, IEEE Congress Evolut. Comput. (Cec) 2013 (2013) 2742–2749.