



A novel local search method for LSGO with golden ratio and dynamic search step

Havva Gül Koçer¹ · Sait Ali Uymaz²

Published online: 28 August 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

Depending on the developing technology, large-scale problems have emerged in many areas such as business, science, and engineering. Therefore, large-scale optimization problems and solution techniques have become an important research field. One of the most effective methods used in this research field is memetic algorithm which is the combination of evolutionary algorithms and local search methods. The local search method is an important part that greatly affects the memetic algorithm's performance. In this paper, a novel local search method which can be used in memetic algorithms is proposed. This local search method is named as golden ratio guided local search with dynamic step size (GRGLS). To evaluate the performance of proposed local search method, two different performance evaluations were performed. In the first evaluation, memetic success history-based adaptive differential evolution with linear population size reduction and semi-parameter adaptation (MLSHADE-SPA) was chosen as the main framework and comparison is made between three local search methods which are GRGLS, multiple trajectory search local search (MTS-LS1) and modified multiple trajectory search. In the second evaluation, the improved MLSHADE-SPA (IMLSHADE-SPA) framework which is a combination of MLSHADE-SPA framework and proposed local search method (GRGLS) was compared with some recently proposed nine algorithms. Both of the experiments were performed using CEC'2013 benchmark set designed for large-scale global optimization. In general terms, the proposed method achieves good results in all functions, but it performs superior on overlapping and non-separable functions.

Keywords Large-scale global optimization · Local search · Golden ratio · Memetic algorithm · CEC'2013 LSGO benchmark

1 Introduction

Optimization is a problem-solving technique and its purpose is to calculate the parameters that will produce the optimum result of a function. In other words, the goal in optimization is to find a global optimum for an objective function defined on a specific search space and under some constraints. Most problems in the real world such as scheduling, vehicle routing, bio-computing and

engineering problems, etc., are concerned with finding the maximum or minimum value of a function efficiently. So such real-world problems can be called optimization problems. Large number of decision variables included optimization problems are known as large-scale optimization problems. Large-scale global optimization (LSGO) term is used for this particular category of global optimization problems (Molina et al. 2018b).

Optimization problems should be modeled mathematically in order to be solved by computers. When the deal is large-scale problems, both the modeling and the solution process can become quite complicated depending on the number of variables and data types. Some factors make large-scale problems extremely difficult. Depending on size increase, search space also increases exponentially. That changes in both complexity and properties of some functions and increases the number of dependent variables (Li et al. 2013). Due to all these factors, the solution

Communicated by V. Loia.

✉ Sait Ali Uymaz
sauymaz@ktun.edu.tr

¹ Distance Education Application and Research Center, Selçuk University, Konya, Turkey

² Department of Computer Engineering, Konya Technical University, Konya, Turkey

requires a very large computational cost. Therefore, more effective and efficient new search techniques are needed to explore all promising regions in the search space that expands exponentially with the number of problem dimensions within a given time. As a result, researchers have proposed various algorithms and techniques to overcome these problems. These studies can be roughly divided into three classes; standard evolutionary algorithms, cooperative coevolution (CC)-based evolutionary algorithms and memetic algorithms (Singh and Jana 2017). The proposed method in this paper is a novel local search method which can be classified as a memetic algorithm.

Depending on the developing technology, large-scale problems have emerged in many areas such as business, science and engineering. Therefore, large-scale optimization problems and solution techniques have become an important research area and are getting more and more attention in last decades. International competitions and private sessions attract researchers' attention to focus on LSGO and play a key role in designing powerful, scalable and efficient algorithms in this area. The first special session about LSGO was held back in 2008 at the IEEE Congress of Evolutionary Computation. Since then, the CEC has held a special session on LSGO every year, except 2009 and 2011. A detailed review and analysis about CEC special sessions and competitions of LSGO can be found in Molina et al. (2018a).

Objectives of this work can be summarized as follows:

1. To design a local search method which can be an effective option for LSGO.
2. To analyze the effect of local search methods on a main framework.
3. To analyze the performance of a memetic framework which uses proposed local search method by comparing it with recently proposed algorithms.

In line with these research objectives, the remainder of the paper is organized as follows. In Sect. 2, related work about memetic algorithms and local search methods are given. Section 3 presents a novel local search method in detail. Numerical experiments and performance evaluation are given in Sect. 4. Finally, in Sect. 5, conclusions are made.

2 Background and related works

The optimization of large-scale problems is considered a challenging task. Because the increased complexity depending on size increase makes it impossible to search for every possible solution or combination when searching for solution of a problem. For this reason, it is preferable to use heuristic methods that can find acceptable quality

solutions at reasonable time. Classic heuristic algorithms have difficulty in dealing with the problems caused by size increase and thus lose their advantages and effectiveness in design. For this case, "curse of dimensionality" term which refers to the difficulties of finding optimum in high-dimensional space using extensive search is used (Chen et al. 2015). Motivated by the purpose, many effective and robust metaheuristic algorithms have been proposed to obtain high-quality solutions with low calculation costs.

One of the most effective methods used for LSGO problems is memetic algorithm (MA) which is the combination of evolutionary algorithms (EAs) with local search (Moscato 1989). MAs are inspired from combination of natural adaptation and lifetime learning of the individuals in a population. MA is a kind of EA that apply separate processes to exploit search space, for example, using extra local search methods to improve their fitness (Krasnogor and Smith 2005).

When considering related papers and the competitions held in the special sessions, it is seen that memetic algorithms have an important and effective place in the LSGO research area. The first winner of the LSGO competitions was multiple trajectory search (MTS) (Tseng and Chen 2008) which is a combination of three local searches presented in CEC'2008. Results of MTS have led many researchers to realize that local searches can significantly improve the results of EAs. Then, researchers head toward to embed local search technics to existing optimization algorithms. The algorithms that adopt memetic approach and have success in CEC competitions are briefly mentioned in chronologically in following lines. MA-SW-Chains (Molina et al. 2010) is a memetic algorithm which is combination of genetic algorithm with Solis and Wets algorithm (Solis and Wets 1981). Likewise, self-adaptive differential evolution with multi-trajectory search (SaDE-MMTS) (Zhao et al. 2011) is combined JADE with modified version of MTS-LS1 which is the first one of the local searches of the MTS algorithm. Multiple offspring sampling (MOS) (LaTorre et al. 2012) is a hybrid approach combining the Solis & Wets Algorithm and MTS-LS1 local search techniques. In the improved version of MOS (LaTorre et al. 2013), MTS-LS1-Reduced which is a modified version of MTS-LS1 and Solis and Wets are used as local search methods. The improved version of MOS has been also the winner and unrivaled algorithm until 2018 in CEC LSGO Competitions. Iterative Hybridization of DE with local search (IHDELS) (Molina and Herrera 2015) is a memetic algorithm that combines differential evolution algorithm with two local search methods [MTS-LS1 and L-BFGS-B (Morales and Nocedal 2011)] in an iterated way. Coral reefs optimization with substrate layers (CRO-SL) (Salcedo-Sanz et al. 2016) is a memetic algorithm which is used MTS-LS1 as local search method.

MLSHADE-SPA (Hadi et al. 2019) is a memetic framework that combines three population-based algorithms with a modified version of MTS-LS1 as local search. SHADE-ILS (Molina et al. 2018b) is combined the exploration power of SHADE algorithm with the exploitation ability of several local searches (MTS-LS1 and L-BFGS-B).

When the memetic algorithms are examined which are used in the LSGO field, it can be seen that three of the methods used as local search come into prominence which are MTS-LS1, Solis and Wets, L-BFGS-B.

MTS (Tseng and Chen 2008) uses three different local search methods for looking for candidate solutions that are neighbor to the current solution. Among these local search methods in this algorithm, MTS-LS1 is the most effective one for LSGO. MTS-LS1 evaluates each dimension one by one from the first dimension to the last dimension. MTS-LS1 is a local search method which is suitable for separable problems but sensitive to rotations (Molina et al. 2018b). MTS-LS1 and its variants were used as local search methods in algorithms and frameworks like SHADE-ILS (Molina et al. 2018b), MTS (Tseng and Chen 2008), SaDE-MMTS (Zhao et al. 2011), MOS (LaTorre et al. 2012, 2013), IHDELS (Molina and Herrera 2015), CRO-SL (Salcedo-Sanz et al. 2016), MLSHADE-SPA (Hadi et al. 2019).

The Solis and Wets Algorithm is a local search method which adapts the step size according to consecutive failures or successes of the movement from another solution. This method was used in MA-SW-Chains (Molina et al. 2010) and both 2012 (LaTorre et al. 2012) and 2013 (LaTorre et al. 2013) are versions of MOS which are pioneering algorithms in LSGO field. More detailed information about Solis and Wets search algorithm can be found in Solis and Wets (1981).

L-BFGS-B (Morales and Nocedal 2011) is used as a local search method that uses an approximation of the gradient to improve the search and sensitive to rotation (Molina et al. 2018b). This method is used in IHDELS (Molina and Herrera 2015) and SHADE-ILS (Molina et al. 2018b).

As can be seen in these examples (Fig. 1), the combination of a population-based method and a local search has been used in many of the best-performing methods. This

technique which provides a good balance between exploration and exploitation, enables design of effective and efficient algorithms to overcome the challenges of LSGO.

Golden ratio (GR) is an irrational number with fractal properties and is equal to value of 1.61803398874989 and inverse golden ratio (IGR) is 0.6180339887. GR and IGR have the same digits after the decimal point. It defines an especial proportion which can be experienced in several natural phenomena like the crystals geometry, the placement of stems in plants, the body parts proportions, and in the proportions of human. Nature, biology, computational science, stock market, mathematics, engineering, industrial design, architecture, art, etc., can be given as examples of fields where golden ratio is used (Ciucurel et al. 2018; Dabbour 2012; Lu 2003; Shekhawat 2015). More detail information about Golden Ratio can be obtained from (Hemenway 2005; Iosa et al. 2018; Livio 2008; Lutge and Souza 2019).

Golden ratio is also used in optimization algorithms as a coefficient or search technique or selection method.

Golden ratio is used as a coefficient in golden ratio particle swarm optimization (GRPSO) (Manikantan et al. 2012). In this algorithm, c_1 and c_2 parameters in the basic PSO velocity equation are replaced by inverse golden ratio (IGR) and gold ratio (GR), respectively.

Golden section search (GSS) is widely used as a local search method in optimization for unimodal functions. In this method, the golden ratio is used to determine the new lower and upper bounds for next step in order to divide the search space. GSS is constantly narrows the search space to seek the optimum solution (Aurasopon and Khamsen 2019; Bansal et al. 2013; Koupaei et al. 2016; Kumar et al. 2014; Neri and Tirronen 2009; Zhang 2019).

In the use of the golden ratio as a selection method, the population is divided into subgroups determined by the golden ratio. Each subgroup contains a certain number of individuals which have similar quality. The probability of each individual being chosen depends on the group to which it belongs. This approach provides a better balance between elitism and diversity (Cuevas et al. 2018).

In the GRGLS, the method proposed in this study, the golden ratio was used as a coefficient to adjust the adaptive step size. During the search process, the step size is

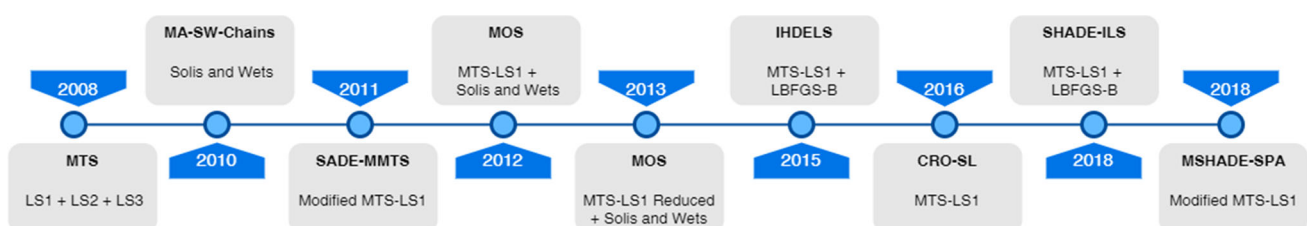


Fig. 1 Graphical representation of related works

increased or decreased according to the improvement in individual’s fitness value. This ensures a better balance between exploration and exploitation. So that GRGLS does not have any limitations according to its problem characteristics (unimodal, multimodal, etc.).

3 Proposed method

3.1 Improved MSHADE-SPA (IMSHADE-SPA)

MLSHADE-SPA (Hadi et al. 2019) which is a memetic framework chosen as the main framework on solving LSGO problems effectively. This memetic framework is a combination of population-based algorithms (LSHADE-SPA (Mohamed et al. 2017), EADE (Mohamed 2017), and ANDE (Mohamed and Almazayad 2017)) and a local search method (MMTS).

MLSHADE-SPA uses different strategies in the search process, such as sharing computational resources, divide and conquer, and population reduction. While sharing computational resources, \max_{FES} is divided into two main parts. EAs consume half part of \max_{FES} , remains is consumed by local search method. EAs use their computational resources in two stages. In the first stage, LSHADE-SPA works on all dimensions. In the second stage, all EAs work together by sharing dimensions (divide and conquer) and computational resources allocated for each EAs according to improvement ratio. Distribution of computational resources is shown in Fig. 2.

In this framework, linear population size reduction (LPSR) is applied until population size reaches the minimum number of population. The population size to be reduced (Pop_{Reduce}) is calculated by Eq. (1). After Pop_{Reduce} is calculated, first, individuals are sorted according to their fitness values. Individuals which have the worst fitness values are removed from the current population until the population size reaches Pop_{Reduce} . With

the effect of LPSR, population size will reach the min pop size within the first half of \max_{FES} .

$$Pop_{Reduce} = \text{round} \left[\left(\frac{Pop_{init} - Pop_{min}}{0.5 * \max_{FES}} \right) * c_{FES} + Pop_{init} \right] \tag{1}$$

where c_{FES} is the current fitness evaluation number, \max_{FES} is the maximum fitness evaluation number, Pop_{init} is the initial population size and Pop_{min} is the minimum population size.

MLSHADE-SPA is improved by replacing its local search method (MMTS) with proposed GRGLS local search method and it is named as Improved MSHADE-SPA (IMSHADE-SPA). Figure 3 illustrates the structure of the Improved MSHADE-SPA.

3.2 A novel local search method: golden ratio guided local search with dynamic step size (GRGLS)

Evolutionary algorithms (EAs) usually use the combination of nature-inspired rules and randomness to achieve the optimal solution of optimization problems. EAs perform two fundamental search phases which are exploration and exploitation. Visiting completely new points in a search area is called exploration, and fine-tuning those points for the purpose of improving within the neighborhood of previously visited locations is called exploitation. The success of an evolutionary algorithm depends on the capacity to find a good balance between exploration and exploitation. The purpose of this balance is to prevent the stuck in local solutions that are not optimum or to detect sensitive solutions that may be closer to the target solution near the available solution. In EAs, the search process usually begins with exploration, and then gradually turns into exploitation (Črepinšek et al. 2013).

Memetic algorithms are the combination of population-based global search and the heuristic local search. In exploration phase, population-based algorithm guides the

Fig. 2 Distribution of computational resources of the MLSHADE-SPA

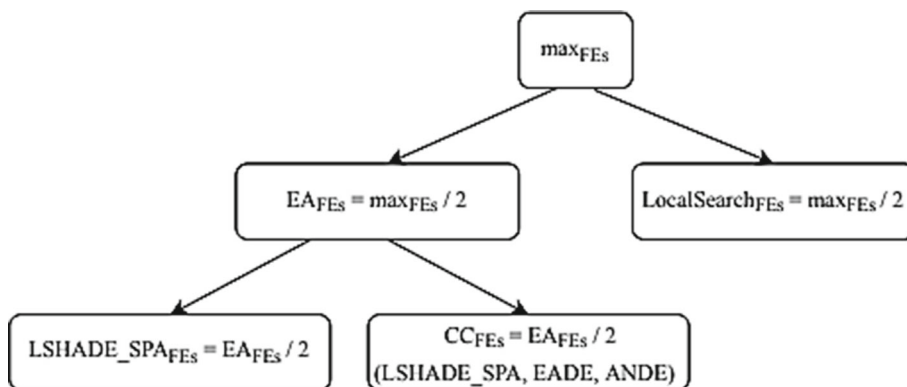


Fig. 3 Structure of improved MSHADE-SPA

Memetic Framework (Improved MSHADE-SPA)	Initialization	initialize framework parameters initialize LSHADE-SPA, EADE, ANDE, MMTS parameters initialize population randomly $flag = 0$; $iteration_{FES} = maxFES / iteration_number$;
	Population-Based Algorithms (EAs)	$EA_{FES} = iteration_{FES} / 2$; $SPA_{FES} = EA_{FES} / 2$; [Pop , Fit] = SPA (SPA_{FES} , Pop , Fit) ; // LSHADE-SPA works on all dimensions until it consumes its FEs $CC_{FES} = EA_{FES} / 2$; if $flag = 0$ $SPA_{CC_{FES}} = CC_{FES} / 3$; $EADE_{CC_{FES}} = CC_{FES} / 3$; $ANDE_{CC_{FES}} = CC_{FES} / 3$; $flag = 1$; else calculate CC_{FES} per EA according to the improvement ratio of each algorithm $CC_{ind} = Divide_Dimension (3 , Dim)$ // three randomly divided mutually exclusive group [Pop , Fit , imp(1)] = SPA_CC ($SPA_{CC_{FES}}$, Pop , Fit , $CC_{ind}(1)$) ; // LSHADE-SPA works on one-third of all dimensions [Pop , Fit , imp(2)] = EADE_CC ($EADE_{CC_{FES}}$, Pop , Fit , $CC_{ind}(2)$) ; // EADE works on one-third of all dimensions [Pop , Fit , imp(3)] = ANDE_CC ($ANDE_{CC_{FES}}$, Pop , Fit , $CC_{ind}(3)$) ; // ANDE works on one-third of all dimensions
	Local Search	$GRGLS_{FES} = iteration_{FES} / 2$; [Pop , Fit] = GRGLS ($GRGLS_{FES}$, Pop , Fit) ; // GRGLS works on each dimension of best solution one by one
	Finalization	Calculate the improvement ratio of each algorithm Apply Linear Population Size Reduction(LPSR)

search and find candidate solutions by exploring the solution space. Then the exploitation phase is triggered by calling the local search method. Local search usually starts from a candidate solution and iteratively moves to a neighbor solution thus can further improve a candidate solution in the search space by exploiting its neighborhood. This collaboration between global search and local search enables the solution to be searched in solution space much more thoroughly and makes the search efficiency of MAs outperform EAs. As a result, memetic algorithms provide higher performance than using either stand-alone population-based global search or stand-alone local search (Lin and Chen 2011).

In this paper, a novel local search method named as golden ratio guided local search with dynamic step size (GRGLS) is proposed. This method can be used in memetic algorithms for the purpose of developing better solutions for large-scale optimization problems. In proposed local search method, golden ratio is used for changing the step size dynamically.

When the local search methods which used in memetic algorithms are considered, it is common that searching along each dimension from the first to the last is an effective method in the LSGO field. MTS-LS1 and its variants that use this technique are preferred by effective

and efficient algorithms (Hadi et al. 2019; LaTorre et al. 2012, 2013; Molina and Herrera 2015; Molina et al. 2018b; Salcedo-Sanz et al. 2016; Tseng and Chen 2008; Zhao et al. 2011). Another effective operation used in both solis and wets algorithm and MTS-LS1 is to look at the points in the opposite directions of the current dimension. The proposed local search method GRGLS incorporates both techniques by researching solution at opposite directions of current solution with changing the step size under the guidance of the golden ratio.

In golden ratio guided local search with dynamic step size (GRGLS), each dimension of problem has its own step size. After results are evaluated of the experiments made by giving different initial values to the step size, the step size optimum initial value has been determined as 0.1 for each dimension. Local search is applied to the best individual which has the highest fitness value in the population along each dimension from the first to the last. In all following equations, x_i represents current position, x_{i+1} represents next position, dss_i represents step size of corresponding dimension, GoldenRatio represents 1.618, InverseGoldenRatio represents 0.618.

Firstly, the point which is forward of the best individual as far as its own step size is evaluated for purpose to investigate a better solution by Eq. (2).

$$x_i^{t+1} = x_i^t + dss_i^t \quad (2)$$

New candidate solution which is formed by changing one of its parameter is evaluated by object function. If the new solution is better than the best solution, new solution assigns as best individual and step size of changed dimension is increased by golden ratio. If new solution is worse than best solution, the point which is backward of the best individual as far as its own step size is evaluated for purpose to investigate a better solution by Eq. (3).

$$x_i^{t+1} = x_i^t - dss_i^t \quad (3)$$

New candidate solution which is formed by changing one of its parameter is evaluated by object function. If the new solution is better than the best solution ($f(x_{best}^t)$), new solution assigns as best individual and step size of changed dimension is increased by golden ratio, otherwise step size of changed dimension is decreased by inverse golden ratio Eq. (4).

$$dss_i^{t+1} = \begin{cases} dss_i^t * \text{GoldenRatio}, & f(x_i^{t+1}) \leq f(x_{best}^t) \\ dss_i^t * \text{InverseGoldenRatio}, & f(x_i^{t+1}) > f(x_{best}^t) \end{cases} \quad (4)$$

All these steps are repeated by moving to the next dimension until it reaches the last dimension of problem. Flowchart and pseudo-code of GRGLS are given in Figs. 4 and 5.

4 Performance evaluation

4.1 Experimental environment

Experiments were performed on the computer platform which is based on an Intel(R) Core(TM) i7-7700HQ 2.80 GHz processor, 16.0 GB of RAM, and the Microsoft Windows 10 Enterprise operating system.

All experimental works were conducted by using MATLAB. The source code is available on Uymaz (2020). Code of MLSHADE-SPA, used as main framework in our experiments, is downloaded from authors' website (Mohamed 2019). The proposed method GRGLS and MTS-LS1 were also coded in MATLAB.

4.2 Benchmark set

Performance analysis experiments were carried out with the IEEE CEC 2013 benchmark set (Li et al. 2013). It is the latest defined and most recent benchmark set for LSGO. This benchmark set is designed to be as similar to the features of real-world problems as possible. To accomplish this, the characteristics included in the functions are “Non-

uniform subcomponent sizes, imbalance in the contribution of subcomponents, functions with overlapping subcomponents and new transformations to the base functions” (Li et al. 2013). This benchmark set includes 15 continuous optimization functions which have different separability degrees. All functions in benchmark set are the minimization problems and the optimum function values of them are zero. The dimensions of all functions in benchmark set are 1000 except for F13 and F14 where D is 905. Main properties are given in Table 1 but detailed information about the benchmark set can be found in Li et al. (2013).

4.3 Experimental results

In this study, two performance analysis experiments were performed to verify the efficiency of the proposed local search method GRGLS. Firstly, GRGLS's performance has been compared with MTS-LS1 and MMTS local search methods each by placing it in the same main framework. Secondly, the IMLSHADE-SPA algorithm which is formed by replacing the local search method of MLSHADE-SPA algorithm with GRGLS has been compared with nine recently proposed algorithms.

4.3.1 Effect of proposed local search method on main framework's performance

In the first phase of experiments, the main framework was chosen and results obtained by using only one from GRGLS, MTS-LS1, MMTS at a time in the same main framework were evaluated.

In MTS-LS1, the initial step values for each dimension are determined as the difference between the upper bound and the lower bound. Local search is only applied to all dimensions of the best individual. Step size of each dimension can be different. Local search is applied by adding or subtracting step size to the corresponding dimension. Firstly, step size is subtracted from solution. If it doesn't reach a better result, half of the step size added to the original value of current dimension. If a good result is not still achieved as a result of these two moves, it decreases the step size by half and checks whether the new step size below $1.00E-15$. If it falls below this value, the step size is updated by multiplying the difference between upper and lower by 0.4. More detailed information about this method can be obtained from Tseng and Chen (2008).

In MMTS which is used in MLSHADE-SPA, the initial step values for each dimension are determined by multiplying random number in range $[0, 1]$ to the difference between the current minimum and maximum values of the dimension. One of these two values whichever is smaller is assigned as the initial values for each dimension. When

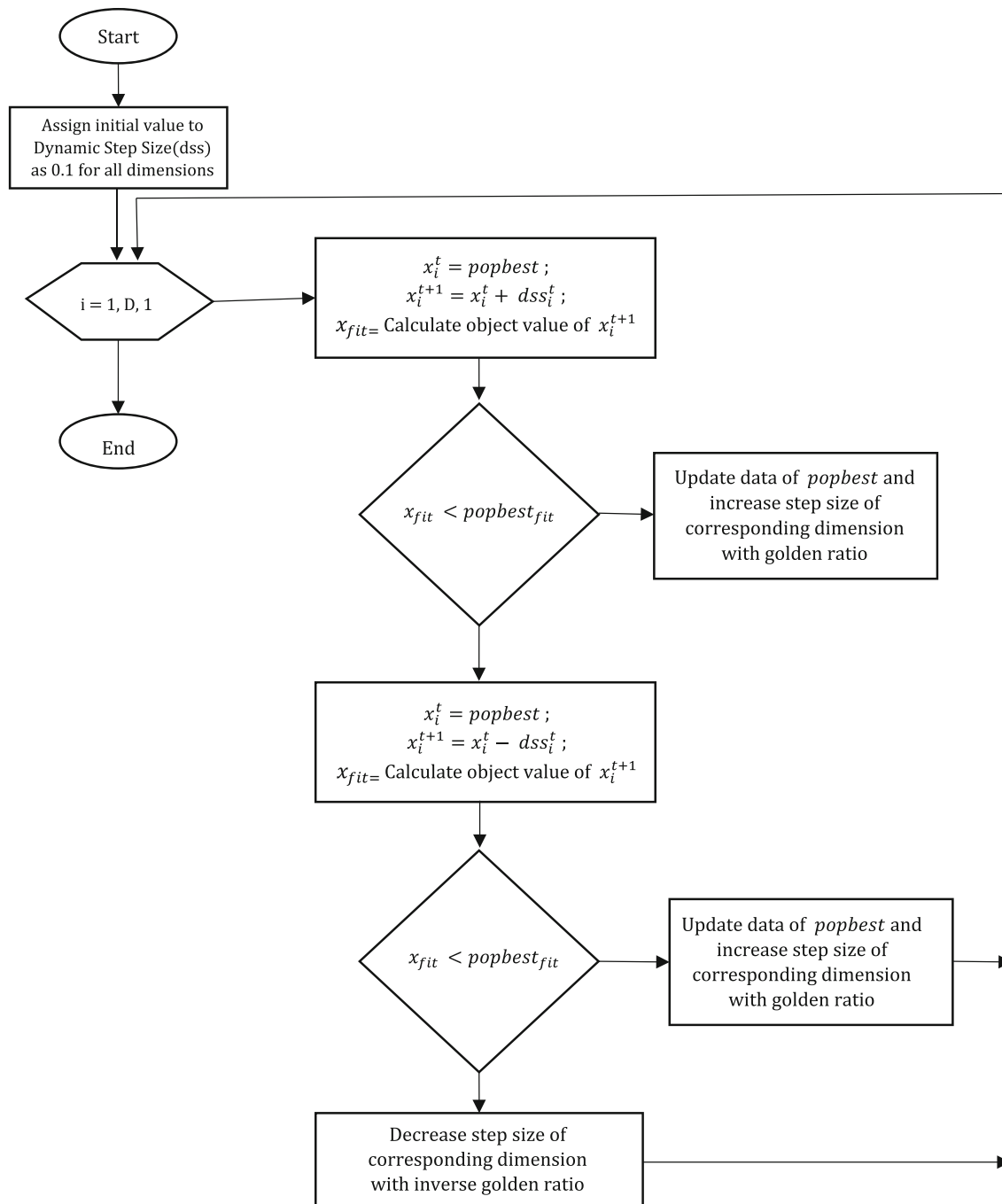


Fig. 4 Flowchart of the GRGLS

searching for the corresponding dimension, firstly, the point which is forward (+) of the best individual as far as its own step size is evaluated for the purpose to investigate a better solution. If it reaches a better result, the search will continue with this direction by adding another step size until it fails to generate a better result or reach the upper bound. In other words, the step size increases linearly as long as the result is successful. Then it runs the same procedure backward (−) from the point at which it reached

its last successful result. More detailed information about this method can be obtained from Hadi et al. (2019).

Briefly, GRGLS, MTS-LS1, MMTS are basically three similar local search methods with differences.

The similarities between these methods are;

- Applying local search to the best individual which has the highest fitness value in the population,

```

D : dimension size
popbest : highest fitness valued individual of population
step : an array which stores movement size for each dimension
1 : assign initial value to step as 0.1 for all dimensions
2 : for i=1 to D
3 :     assign popbest to candidate solution
4 :     investigate a better solution by Eq. (2)
5 :     check boundary violation and fix
6 :     calculate object value of current candidate solution
7 :     If candidate solution fitness better than popbest fitness
8 :         assign candidate solution position and fitness to popbest
9 :         step size of corresponding dimension is increased by golden ratio Eq.(4)
10 :    else if
11 :        assign popbest to candidate solution
12 :        investigate a better solution by Eq. (3)
13 :        check boundary violation and fix
14 :        calculate object value of current candidate solution
15 :        If candidate solution fitness better than popbest fitness
16 :            assign candidate solution position and fitness to popbest
17 :            step size of corresponding dimension is increased by golden ratio Eq.(4)
18 :        else if
19 :            step size of corresponding dimension is decreased by inverse golden ratio Eq. (4)
20 :        End
21 :    End
22 : End

```

Fig. 5 Pseudo-code of the GRGLS

- Searching along one dimension from the first dimension to the last dimension,
- Looking at the opposite points of the current dimension while creating a new solution.

The differences between these methods are;

- Each method follows a different path when assigning initial values to the array that holds the step sizes for each dimension.
- Each method uses different coefficients when changing the step size.
- Proposed method increases step size exponentially when finds a better solution while others keeps step size unchanged or increases linearly, in the same situation.

Experiments are carried out by codes which obtained from MLSHADE-SPA authors' website. Firstly, original codes which include MMTS as a local search method were run and results were recorded. Then the same experiments were performed in equal conditions with replacing MMTS

with MTS-LS1 (Tseng and Chen 2008) and results has been stored. Finally, the same experiments were performed in equal conditions with replacing MMTS with GRGLS. The initial population size, minimum population size and iteration number parameters of the framework were used as 250, 20 and 50, respectively. In all experiments, the suggested instructions for CEC special sessions and competitions have been followed. All reported results are the averages of 25 independent runs for each test problem. Maximum fitness evaluation (\max_{FEs}) was set to $3.0e6$. In all these experiments, results are recorded when $FEs = 1.2e5$, $6.0e5$, and $3.0e6$ according to CEC competition guidelines (Li et al. 2013) and results of $FEs = 1.2e5$ are listed in Table 2, results of $FEs = 6.0e5$ are listed in Table 3 and results of $FEs = 3.0e6$ are listed in Table 4.

In addition, the Wilcoxon signed-rank test was conducted for statistical analysis using the best values recorded at the end of each run. The results of these analyses between GRGLS and other both local search methods are demonstrated for $1.2e5$, $6.0e5$, and $3.0e6$ FEs in Tables 5, 6 and 7, respectively.

Table 1 Functions list

Function name	Range	Properties	Group
F1 Elliptic function	[− 100, 100]	Unimodal, Shifted	Fully separable functions
F2 Rastrigin function	[− 5, 5]	Multimodal, Shifted	
F3 Ackley function	[− 32, 32]	Multimodal, Shifted	
F4 Elliptic function	[− 100, 100]	Unimodal, Shifted	Functions with a separable subcomponent
F5 Rastrigin function	[− 5, 5]	Multimodal, Shifted	
F6 Ackley function	[− 32, 32]	Multimodal, Shifted	
F7 Schwefel's Problem 1.2	[− 100, 100]	Multimodal, Shifted	
F8 Elliptic function	[− 100, 100]	Unimodal, Shifted	Functions with no separable subcomponents
F9 Rastrigin function	[− 5, 5]	Multimodal, Shifted	
F10 Ackley function	[− 32, 32]	Multimodal, Shifted	
F11 Schwefel's Problem 1.2	[− 100, 100]	Unimodal, Shifted	
F12 Rosenbrock's function	[− 100, 100]	Multimodal, Shifted	Overlapping functions
F13 Schwefel's function with conforming overlapping subcomponents	[− 100, 100]	Unimodal, Shifted	
F14 Schwefel's function with conflicting overlapping subcomponents	[− 100, 100]	Unimodal, Shifted	
F15 Schwefel's Problem 1.2	[− 100, 100]	Unimodal, Shifted	Non-separable functions

Considering the results of $1.2e5$ fitness evaluation it is seen that GRGLS is the method that obtains the best result in 8 of 15 functions. It is in second place in the general ranking. In $1.2e5$ fitness evaluation, the proposed method's performance is lower than other methods in full separable functions.

Considering the results of $6.0e5$ fitness evaluation it is seen that GRGLS is the method that obtains the best result in 9 of 15 functions. It is observed that the proposed method improves itself in all functions when compared with the results of $1.2e5$ fitness evaluation. The development of F10 and F12 functions which are multimodal and member of two different function groups is remarkable. GRGLS is ranked first in the general ranking in $6.0e5$ fitness evaluation.

Considering the results of $3.0e6$ fitness evaluation it is seen that GRGLS is the method that obtains the best result in 10 of 15 functions. When compared with the results of $6.0e5$ fitness evaluation, only rank of F5 and F9 is changed for the proposed method. After $6.0e5$ fitness evaluation, the

proposed method is still keep the first position in general ranking.

Averages of the rankings specified in Tables 2, 3, and 4 are listed in Fig. 6. The lowest value in the ranking averages represents the most successful algorithm. The results in Fig. 6 show that the proposed method is more successful than the others. Although MTS-LS1 is the best method in $1.2e5$ fitness evaluation, after $6.0e5$ fitness evaluation GRGLS has taken the first place and has maintained its success by obtaining the best result for 10 of 15 functions in $3.0e6$ fitness evaluation.

The convergence curves of GRGLS, MTS-LS1, MMTS for the selected six problems: F1, F5, F7, F11, F13, F15 are presented in Fig. 7. Each of these functions has been selected from different function groups. Y axes represent mean values of 25 independent runs of related function. X axes represent a recorded point in each 2000 fitness evaluations. For example, 300 value at x axis corresponds to $6e5$ FEs.

Table 2 Results (mean values) for the 1.2e5 FEs milestone

	Local search methods			Ranks		
	MTS-LS1	MMTS	GRGLS	MTS-LS1	MMTS	GRGLS
F1	2.50E+03	5.72E+07	7.05E+07	1	2	3
F2	1.34E+03	1.81E+03	2.14E+04	1	2	3
F3	8.85E−03	3.59E+00	1.97E+01	1	2	3
F4	3.33E+11	6.77E+11	2.27E+11	2	3	1
F5	1.04E+07	1.06E+07	1.12E+07	1	2	3
F6	9.69E+05	9.90E+05	9.69E+05	1.5	3	1.5
F7	5.23E+09	7.51E+09	4.99E+09	2	3	1
F8	1.12E+16	3.14E+16	8.48E+15	2	3	1
F9	7.82E+08	8.51E+08	9.27E+08	1	2	3
F10	2.00E+07	3.47E+07	5.21E+07	1	2	3
F11	1.02E+12	9.91E+11	4.72E+11	3	2	1
F12	4.27E+03	1.47E+07	4.85E+07	1	2	3
F13	4.33E+10	4.16E+10	1.87E+10	3	2	1
F14	8.45E+11	1.01E+12	5.66E+11	2	3	1
F15	2.57E+09	6.38E+09	4.66E+08	2	3	1
			Avg =>	1.63	2.40	1.96

Table 3 Results (mean values) for the 6.0e5 FEs milestone

	Local search methods			Ranks		
	MTS-LS1	MMTS	GRGLS	MTS-LS1	MMTS	GRGLS
F1	5.11E+02	3.36E+00	1.09E+01	3	1	2
F2	1.32E+03	2.26E+02	7.68E+03	2	1	3
F3	3.45E−03	1.52E−03	6.56E+00	2	1	3
F4	2.80E+10	2.54E+10	2.41E+10	3	2	1
F5	2.73E+06	4.50E+06	3.46E+06	1	3	2
F6	2.47E+03	4.48E+03	5.56E+03	1	2	3
F7	6.21E+08	8.08E+08	3.63E+08	2	3	1
F8	2.78E+14	2.98E+14	1.75E+14	2	3	1
F9	2.36E+08	3.81E+08	3.23E+08	1	3	2
F10	5.66E+03	7.04E+03	4.20E+03	2	3	1
F11	8.70E+09	1.17E+10	4.49E+09	2	3	1
F12	2.93E+03	3.16E+03	2.41E+03	2	3	1
F13	1.57E+10	1.76E+10	4.79E+09	2	3	1
F14	1.09E+11	1.69E+11	9.78E+10	2	3	1
F15	2.53E+07	7.20E+07	1.56E+07	2	3	1
			Avg =	1.93	2.47	1.60

Convergence curves confirm that GRGLS is more successful compared to other local search methods as in the results shown in Tables 2, 3 and 4. The three methods produce similar results except full separable functions up to approximately 1.5e6 fitness evaluations. At the point performance of the methods are differs from each other. The proposed method draws a robust convergence characteristic from beginning to the end. So, it can be seen that GRGLS is more successful compared to other local search methods.

To analyze the problem-solving performance of the local search methods the Wilcoxon signed-rank test was used in addition to averages best values tables, averages of the rankings tables and convergence curves. 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. In Tables 5, 6 and 7, statistical comparisons by Wilcoxon signed-rank test are demonstrated between GRGLS and other local search methods. The last rows of Tables 5, 6 and 7 show the total counts in the (+/

Table 4 Results (mean values) for the 3.0e6 FEs milestone

	Local search methods			Ranks		
	MTS-LS1	MMTS	GRGLS	MTS-LS1	MMTS	GRGLS
F1	3.24E-07	2.60E-22	4.97E-11	3	1	2
F2	1.18E+03	8.39E+01	4.65E+03	2	1	3
F3	2.30E-08	9.96E-14	2.28E+00	2	1	3
F4	8.88E+08	7.74E+08	3.78E+08	3	2	1
F5	1.93E+06	1.91E+06	1.75E+06	3	2	1
F6	1.53E+03	2.59E+03	3.67E+03	1	2	3
F7	4.05E+05	4.76E+04	4.90E+03	3	2	1
F8	1.21E+13	1.23E+13	5.40E+12	2	3	1
F9	1.48E+08	1.51E+08	1.63E+08	1	2	3
F10	5.23E+02	6.16E+02	4.95E+02	2	3	1
F11	4.36E+07	3.51E+07	2.00E+06	3	2	1
F12	3.18E+02	1.41E+02	1.08E+02	3	2	1
F13	3.76E+08	8.29E+07	3.05E+06	3	2	1
F14	1.42E+07	1.44E+07	5.44E+06	2	3	1
F15	5.31E+06	2.62E+07	1.23E+06	2	3	1
			Avg=	2.33	2.07	1.60

Table 5 The results of two-sided Wilcoxon signed-rank test for the 1.2e5 FEs milestone ($\alpha = 0.05$)

Func.	GRGLS versus MTS-LS1			GRGLS versus MMTS		
	<i>p</i> Value	<i>T</i>	<i>W</i>	<i>p</i> Value	<i>T</i>	<i>W</i>
F1	1.23E-05	325	-	6.96E-01	177	=
F2	1.23E-05	325	-	1.23E-05	325	-
F3	1.23E-05	325	-	1.23E-05	325	-
F4	3.96E-02	86	+	2.26E-05	5	+
F5	1.35E-01	218	=	1.22E-01	220	=
F6	6.57E-01	146	=	3.51E-03	54	+
F7	1.92E-01	114	=	1.38E-02	71	+
F8	3.47E-02	84	+	1.57E-05	2	+
F9	3.51E-03	271	-	6.53E-02	231	=
F10	1.23E-05	325	-	4.07E-05	315	-
F11	3.82E-03	55	+	1.89E-03	47	+
F12	1.23E-05	325	-	1.23E-05	325	-
F13	1.23E-05	0	+	1.23E-05	0	+
F14	4.93E-04	33	+	1.13E-04	19	+
F15	1.23E-05	0	+	1.39E-05	1	+
	+/-/-	6/3/6		+/-/-	8/3/4	

Table 6 The results of two-sided Wilcoxon signed-rank test for the 6.0e5 FEs milestone ($\alpha = 0.05$)

Func.	GRGLS versus MTS-LS1			GRGLS versus MMTS		
	<i>p</i> Value	<i>T</i>	<i>W</i>	<i>p</i> Value	<i>T</i>	<i>W</i>
F1	1.23E-05	0	+	6.85E-03	62	-
F2	1.23E-05	325	-	1.23E-05	325	-
F3	1.23E-05	325	-	1.23E-05	325	-
F4	1.04E-01	102	=	4.59E-01	135	=
F5	8.71E-03	260	-	2.26E-03	49	+
F6	1.57E-03	280	-	2.21E-01	208	=
F7	2.70E-03	51	+	4.57E-05	11	+
F8	6.02E-04	35	+	2.66E-04	27	+
F9	1.77E-05	322	-	1.73E-02	74	+
F10	1.58E-01	110	=	2.14E-02	77	+
F11	3.22E-03	53	+	4.07E-05	10	+
F12	9.68E-01	161	=	2.70E-03	51	+
F13	1.23E-05	0	+	1.23E-05	0	+
F14	3.39E-01	127	=	3.28E-04	29	+
F15	4.03E-04	31	+	1.23E-05	0	+
	+/-/-	6/4/5		+/-/-	10/2/3	

=/-) format for the three statistical significance cases in the pairwise comparison. Tables 5, 6 and 7 show that GRGLS can achieve statistically better results than the comparison algorithms, with a level of significance $\alpha = 0.05$. Especially, advantage of the GRGLS seems better as fitness evaluation number increases.

4.3.2 Comparison with other LSGO algorithms

MMTS has been used as local search method in MLSHADE-SPA’s original version. It is a modified version of MTS-LS1. In this work, MMTS has been replaced with proposed method GRGLS in MLSHADE-SPA’s framework and thus MLSHADE-SPA’s performance has

Table 7 The results of two-sided Wilcoxon signed-rank test for the 3.0e6 FEs milestone ($\alpha = 0.05$)

Func.	GRGLS versus MTS-LS1			GRGLS versus MMTS		
	<i>p</i> Value	<i>T</i>	<i>W</i>	<i>p</i> Value	<i>T</i>	<i>W</i>
F1	1.23E−05	0	+	1.23E−05	325	−
F2	1.23E−05	325	−	1.23E−05	325	−
F3	1.23E−05	325	−	1.23E−05	325	−
F4	4.46E−04	32	+	2.66E−04	27	+
F5	7.36E−02	96	=	8.27E−02	98	=
F6	1.19E−02	256	−	7.80E−02	228	=
F7	1.23E−05	0	+	1.23E−05	0	+
F8	2.96E−04	28	+	7.33E−04	37	+
F9	1.60E−02	252	−	3.47E−02	241	−
F10	7.16E−01	176	=	2.11E−01	116	=
F11	1.39E−05	1	+	1.39E−05	1	+
F12	7.33E−04	37	+	3.67E−01	129	=
F13	1.23E−05	0	+	1.23E−05	0	+
F14	1.23E−05	0	+	1.23E−05	0	+
F15	1.23E−05	0	+	1.23E−05	0	+
	+/=−		9/2/4	+/=−		7/4/4

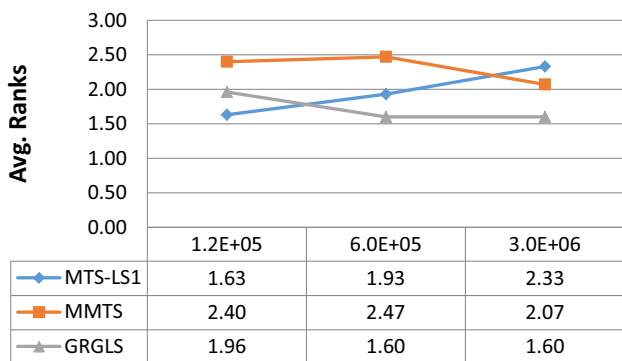


Fig. 6 General performances of methods

been improved. This improved version of MLSHADE-SPA was named as IMLSHADE-SPA.

In the second part of experiments, the IMLSHADE-SPA algorithm was compared with some state-of-the-art algorithms. Briefly information about the algorithms used in the comparison given in Table 8.

Table 9 shows the comparison results of proposed IMLSHADE-SPA and other algorithms using CEC 2013 LSGO benchmark set. Comparisons of the results were obtained from the mean of 25 independent runs at 3.0e+6 FEs. Results of LSGO algorithms were directly taken from related references in Table 8. Best results of the functions were marked bold in Table 9.

When considering the best results of functions in Table 9 and ranking average in Table 10, the following results were achieved:

- In fully separable functions (F1, F2, F3), the best results were obtained from MOS, VGDE, LSGOjDE, VGDE algorithms, respectively. When average rankings in Table 10 evaluated as a function group, VGDE is the most successful algorithm in fully separable functions.
- In functions with a separable subcomponent (F4, F5, F6, F7) and functions with no separable subcomponents (F8, F9, F10, F11), different algorithms achieved best results for each function but when average ranking evaluated, results shows that IMLSHADE-SPA is the most successful algorithm in both function groups.
- All best results of overlapping functions (F12, F13, F14) and non-separable function (F15) were obtained by IMLSHADE-SPA and therefore IMLSHADE-SPA is the most successful algorithm for these function groups.
- As a result of this comparison between all algorithms in 15 functions, the proposed method IMLSHADE-SPA takes the first place while the second algorithm is MOS and the third algorithm is VGDE.

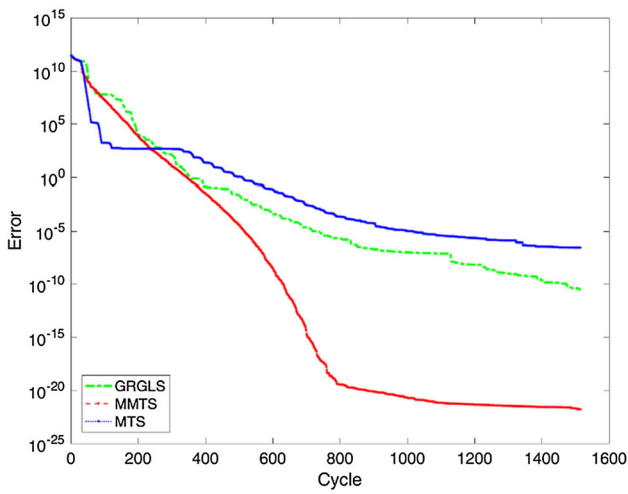
The success order of each algorithm among all algorithms for the corresponding function is listed in Table 10.

Scores of algorithms in two different metrics which are Ranking and Formula One Score on the CEC 2013 benchmark functions are listed in Table 11. For each metric, algorithms are placed in order from best to worst. Average ranks of each algorithm in Table 11 prove that IMLSHADE-SPA is the best algorithm among ten algorithms.

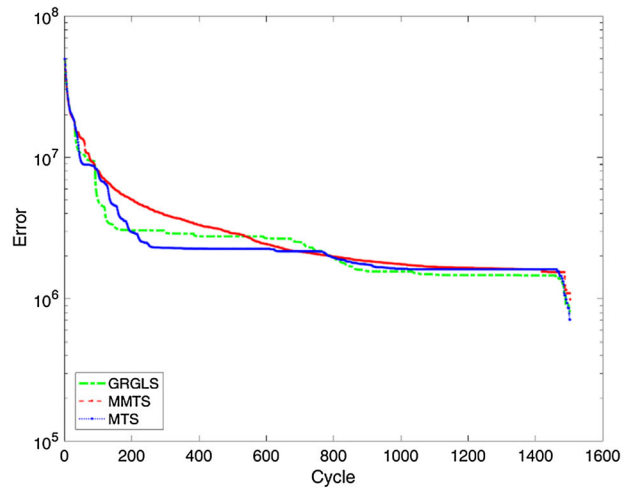
Formula One Score is a metric that inspired from formula one racing scoring system. According to the method, algorithms are sorted according to the performance from best to worst. Then the best 10 algorithm takes 25, 18, 15, 12, 10, 8, 6, 4, 2, and 1 point, respectively. The rest of the algorithms take zero point. The algorithm’s formula one score is calculated by summing the gained points for each function. After all scores are calculated, the most point gainer algorithm is selected as the best algorithm. This metric also is used for comparing the performance of LSGO algorithms (Hadi et al. 2019; Maučec and Brest 2019). According to Formula One Score, IMLSHADE-SPA is the best algorithm with 268 points.

5 Conclusion

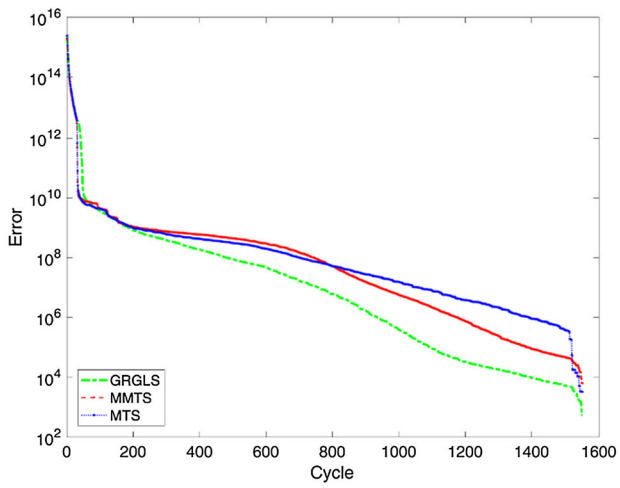
In this work, a local search method named as Golden Ratio Guided Local Search with Dynamic Step Size (GRGLS) is introduced.



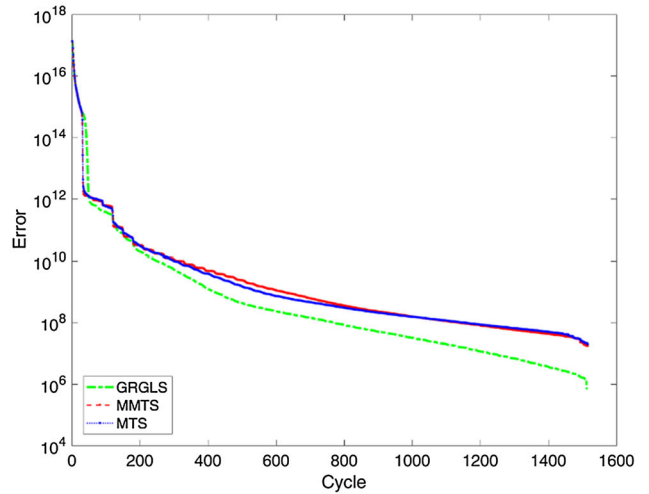
(a) F1



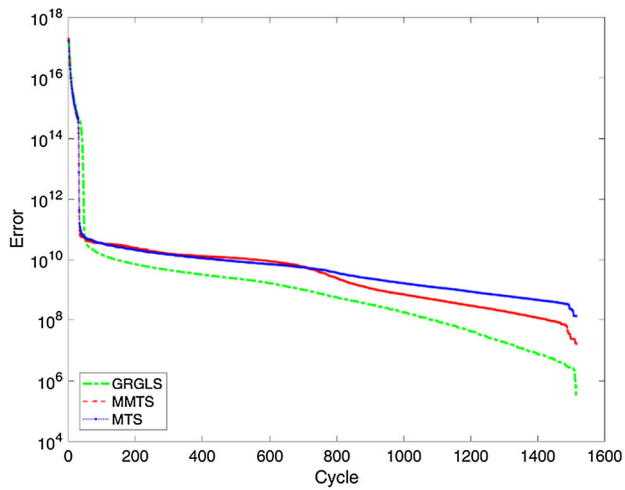
(b) F5



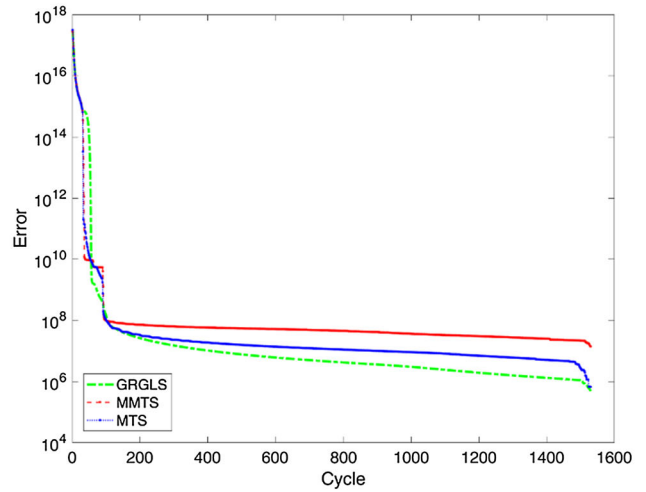
(c) F7



(d) F11



(e) F13



(f) F15

Fig. 7 Convergence curves of sample functions

Table 8 Summary Information of Compared Algorithms

Algorithm	Description	Published Year
LSGOjDE (Maučec et al. 2018)	Self-adapted jDE algorithm with population size reduction for LSGO problems	2018
CBCC3-DG2 (Omidvar et al. 2017)	Contribution-based cooperative co-evolutionary algorithm with improved variant of the differential grouping	2017
CRO-LS (Salcedo-Sanz et al. 2016)	A coral reefs optimization algorithm with substrate layers and local search	2016
CCFR-IDG2 (Yang et al. 2016)	Cooperative coevolution framework using improved differential grouping for efficient resource allocation	2016
IHDELS (Molina and Herrera 2015)	Iterative hybridization of DE with local search	2015
VMO-DE (López et al. 2015)	A hybrid algorithm which includes Variable Mesh optimization(VMO) and differential evolution (DE)	2015
VGDE (Wei et al. 2014)	Variable grouping-based differential evolution using an auxiliary function	2014
MOS (LaTorre et al. 2013)	Multiple offspring sampling is a framework which allows the combination of different metaheuristics and local searches following a HRH (high-level relay hybrid) approach.	2013
MA-SW-Chains (Molina et al. 2010)	MA-SW-chains is a memetic algorithm which is combination of genetic algorithm with solis and wets algorithm as local search	2010

Table 9 Comparison mean values of IMLSHADE-SPA with other LSGO algorithms

	MA-SW-Chains	MOS	VGDE	VMO-DE	IHDELS	CRO-LS	CBCC3-DG2	CCFR-IDG2	LSGOjDE	IMLSHADE-SPA
F1	1.14E-12	0.00E+00	0.00E+00	1.29E-03	4.34E-28	1.84E+06	8.65E+05	2.00E-05	5.64E-03	4.97E-11
F2	1.18E+03	8.32E+02	4.56E+01	5.53E+03	1.32E+03	9.84E+02	1.41E+04	3.60E+02	5.77E+00	4.65E+03
F3	6.78E-13	9.17E-13	3.98E-13	3.70E-04	2.01E+01	2.01E+01	2.06E+01	2.10E+01	2.00E+01	2.28E+00
F4	3.80E+09	1.74E+08	5.96E+08	9.13E+09	3.04E+08	1.55E+10	3.39E+07	9.60E+07	6.71E+08	3.78E+08
F5	2.26E+06	6.94E+06	3.00E+06	7.28E+14	9.59E+06	2.38E+07	2.14E+06	2.80E+06	1.72E+06	1.75E+06
F6	1.07E+04	1.48E+05	1.31E+05	2.15E+05	1.03E+06	1.06E+06	1.05E+06	1.10E+06	1.05E+06	3.67E+03
F7	3.78E+06	1.62E+04	1.85E+03	3.43E+06	3.46E+04	2.78E+08	2.95E+07	2.00E+07	6.18E+05	4.90E+03
F8	4.63E+13	8.00E+12	7.00E+14	6.94E+13	1.36E+12	4.56E+14	6.74E+10	7.00E+10	3.00E+12	5.40E+12
F9	1.14E+08	3.83E+08	2.31E+08	7.68E+08	6.74E+08	5.27E+08	1.70E+08	1.90E+08	3.14E+08	1.63E+08
F10	3.66E+04	9.02E+05	1.57E+02	9.10E+06	9.16E+07	9.44E+07	9.28E+07	9.50E+07	9.24E+07	4.95E+02
F11	2.10E+08	5.22E+07	7.52E+07	1.66E+08	1.07E+07	2.91E+10	7.70E+08	4.00E+08	9.27E+11	2.00E+06
F12	1.23E+03	2.47E+02	2.52E+03	4.45E+03	3.77E+02	3.69E+03	5.81E+07	1.60E+09	1.88E+03	1.08E+02
F13	1.98E+07	3.40E+06	1.36E+09	2.46E+07	3.80E+06	5.33E+09	6.03E+08	1.20E+09	1.67E+07	3.05E+06
F14	1.45E+08	2.56E+07	2.29E+10	9.54E+07	1.58E+07	6.08E+10	1.11E+09	3.40E+09	4.37E+07	5.44E+06
F15	5.90E+06	2.35E+06	3.44E+06	1.10E+07	2.81E+06	1.88E+07	7.11E+06	9.80E+06	2.87E+06	1.23E+06

CEC 2013 benchmark set is used for evaluating the performance of proposed local search method. According to the experimental results, the proposed novel local search method GRGLS achieves well results in all functions in general and a memetic framework which uses GRGLS as local search method significantly outperforms better than many recently proposed algorithms. In addition, the

superior performance of GRGLS on overlapping and non-separable functions is remarkable.

As a result, the analyses indicated that the proposed GRGLS can be used as an effective and efficient local search method for LSGO research area.

For future works, the performance of GRGLS in full separable functions can be improved by applying different techniques or hybridizing with another powerful local

Table 10 Ranks of algorithms on the CEC 2013 functions

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
MA-SW-Chains	4	6	2	8	4	2	7	7	1	3	6	4	5	6	6
MOS	1.5	4	3	3	7	4	3	6	7	4	3	2	2	3	2
VGDE	1.5	2	1	6	6	3	1	10	5	1	4	6	9	9	5
VMO-DE	7	9	4	9	10	5	6	8	10	5	5	8	6	5	9
IHDELS	3	7	7.5	4	8	6	4	3	9	6	2	3	3	2	3
CRO-LS	10	5	7.5	10	9	9	10	9	8	9	9	7	10	10	10
CBCC3-DG2	9	10	9	1	3	7.5	9	1	3	8	8	9	7	7	7
CCFR-IDG2	6	3	10	2	5	10	8	2	4	10	7	10	8	8	8
LSGOjDE	8	1	6	7	1	7.5	5	4	6	7	10	5	4	4	4
IMLSHADE-SPA	5	8	5	5	2	1	2	5	2	2	1	1	1	1	1

Table 11 Scores of algorithms using ranking and formula one score

Algorithm	AvgRank	Place	Formula one score	Place
MA-SW-Chains	4.73	5	173	4
MOS	3.63	2	210	2
VGDE	4.63	3	194	3
VMO-DE	7.07	9	92	9
IHDELS	4.70	4	172	5
CRO-LS	8.83	10	45	10
CBCC3-DG2	6.57	7	125	7
CCFR-IDG2	6.73	8	111	8
LSGOjDE	5.30	6	159	6
IMLSHADE-SPA	2.80	1	268	1

search methods. In addition, GRGLS can be used as a local search technique in different memetic frameworks.

Acknowledgement This study was financially supported by the Coordinatorship of Scientific Research Projects of Selçuk University [Grant no: 18101012].

Compliance with ethical standards

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

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Aurasopon A, Khamsen W (2019) An improved local search involving bee colony optimization using lambda iteration combined with a golden section search method to solve an economic dispatch problem. *Prz Elektrotechniczn* 95:202–208. <https://doi.org/10.15199/48.2019.01.49>

Bansal JC, Sharma H, Arya KV, Nagar A (2013) Memetic search in artificial bee colony algorithm. *Soft Comput* 17:1911–1928. <https://doi.org/10.1007/s00500-013-1032-8>

Chen S, Montgomery J, Bolufé-Röhler A (2015) Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution. *Appl Intell* 42:514–526

Ciucurel C, Georgescu L, Iconaru EI (2018) ECG response to submaximal exercise from the perspective of golden ratio harmonic rhythm. *Biomed Signal Process Control* 40:156–162

Črepinšek M, Liu S-H, Mernik M (2013) Exploration and exploitation in evolutionary algorithms: a survey. *ACM Comput Surv (CSUR)* 45:1–33

Cuevas E, Enríquez L, Zaldívar D, Pérez-Cisneros M (2018) A selection method for evolutionary algorithms based on the golden section. *Expert Syst Appl* 106:183–196

Dabbour LM (2012) Geometric proportions: the underlying structure of design process for Islamic geometric patterns. *Front Archit Res* 1:380–391

Hadi AA, Mohamed AW, Jambi KM (2019) LSHADE-SPA memetic framework for solving large-scale optimization problems. *Complex Intell Syst* 5:25–40

Hemenway P (2005) *Divine proportion: phi in art, nature, and science*. Sterling Publishing Company Inc., New York

Iosa M, Morone G, Paolucci S (2018) Phi in physiology, psychology and biomechanics: the golden ratio between myth and science. *Biosystems* 165:31–39

Koupaei JA, Hosseini SMM, Ghaini FM (2016) A new optimization algorithm based on chaotic maps and golden section search method. *Eng Appl Artif Intell* 50:201–214

Krasnogor N, Smith J (2005) A tutorial for competent memetic algorithms: model, taxonomy, and design issues. *IEEE Trans Evolut Comput* 9:474–488

Kumar S, Sharma VK, Kumari R (2014) Memetic search in differential evolution algorithm. [arXiv preprint arXiv:14080101](https://arxiv.org/abs/14080101)

LaTorre A, Muelas S, Peña J-M (2012) Multiple offspring sampling in large scale global optimization. In: 2012 IEEE congress on evolutionary computation. IEEE, pp 1–8

LaTorre A, Muelas S, Peña J-M (2013) Large scale global optimization: Experimental results with mos-based hybrid algorithms. In: 2013 IEEE congress on evolutionary computation. IEEE, pp 2742–2749

Li X, Tang K, Omidvar MN, Yang Z, Qin K (2013) Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization. *Gene* 7:8

Lin J-Y, Chen Y-P (2011) Analysis on the collaboration between global search and local search in memetic computation. *IEEE Trans Evolut Comput* 15:608–623

Livio M (2008) *The golden ratio: the story of phi, the world’s most astonishing number*. Broadway Books, New York

López ED, Puris A, Bello RR (2015) VMODE: a hybrid metaheuristic for the solution of large scale optimization problems. *Investig Oper* 36:232–239

- Lu Y (2003) A golden section approach to optimization of automotive friction materials. *J Mater Sci* 38:1081–1085. <https://doi.org/10.1023/A:1022362217043>
- Luttge U, Souza GM (2019) The golden section and beauty in nature: the perfection of symmetry and the charm of asymmetry. *Prog Biophys Mol Biol* 146:98–103. <https://doi.org/10.1016/j.pbiomolbio.2018.12.008>
- Manikantan K, Arun B, Yaradoni DKS (2012) Optimal multilevel thresholds based on Tsallis entropy method using golden ratio particle swarm optimization for improved image segmentation. *Proc Eng* 30:364–371
- Maučec MS, Brest J (2019) A review of the recent use of differential evolution for large-scale global optimization: an analysis of selected algorithms on the CEC 2013 LSGO benchmark suite. *Swarm Evolut Comput* 50:100428
- Maučec MS, Brest J, Bošković B (2018) Improved differential evolution for large-scale black-box optimization. *IEEE Access* 6:29516–29531
- Mohamed AW (2017) Solving large-scale global optimization problems using enhanced adaptive differential evolution algorithm. *Complex Intell Syst* 3:205–231
- Mohamed AW (2019). <https://sites.google.com/view/optimization-project/files>. Optimization project. Accessed 23 May 2020
- Mohamed AW, Almazayad AS (2017) Differential evolution with novel mutation and adaptive crossover strategies for solving large scale global optimization problems. *Appl Comput Intell Soft Comput*. <https://doi.org/10.1155/2017/7974218>
- 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), pp 145–152
- Molina D (2015) Herrera F iterative hybridization of DE with local search for the CEC'2015 special session on large scale global optimization. In: 2015 IEEE congress on evolutionary computation (CEC). IEEE, pp 1974–1978
- Molina D, Lozano M, Herrera F (2010) MA-SW-chains: memetic algorithm based on local search chains for large scale continuous global optimization. In: IEEE congress on evolutionary computation. IEEE, pp 1–8
- Molina D, LaTorre A, Herrera F (2018a) An insight into bio-inspired and evolutionary algorithms for global optimization: review, analysis, and lessons learnt over a decade of competitions. *Cognit Comput* 10:517–544
- Molina D, LaTorre A, Herrera F (2018b) SHADE with iterative local search for large-scale global optimization. *IEEE Congress Evol Comput*. <https://doi.org/10.1109/cec.2018.8477755>
- Morales JL, Nocedal J (2011) Remark on “Algorithm 778: L-BFGS-B: fortran subroutines for large-scale bound constrained optimization”. *ACM Trans Math Softw* 38:1–4
- Moscato P (1989) On evolution, search, optimization, GAs and martial arts: toward memetic algorithms. California Inst. Technol., Pasadena, CA, Tech. Rep. Caltech Concurrent Comput. Prog. Rep. 826
- Neri F, Tirronen V (2009) Scale factor local search in differential evolution. *Memet Comput* 1:153–171
- Omidvar MN, Yang M, Mei Y, Li X, Yao X (2017) DG2: a faster and more accurate differential grouping for large-scale black-box optimization. *IEEE Trans Evolut Comput* 21:929–942
- Salcedo-Sanz S, Camacho-Gómez C, Molina D, Herrera F (2016) A coral reefs optimization algorithm with substrate layers and local search for large scale global optimization. In: 2016 IEEE congress on evolutionary computation (CEC). IEEE, pp 3574–3581
- Shekhawat K (2015) Why golden rectangle is used so often by architects: a mathematical approach. *Alex Eng J* 54:213–222. <https://doi.org/10.1016/j.aej.2015.03.012>
- Singh A, Jana ND (2017) A survey on metaheuristics for solving large scale optimization problems. *Int J Comput Appl* 170:1–7
- Solis FJ, Wets RJB (1981) Minimization by random search techniques. *Math Oper Res* 6:19–30
- Tseng LY, Chen C (2008) Multiple trajectory search for large scale global optimization. In: 2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence). IEEE, pp 3052–3059
- Uymaz SA (2020). <http://sauymaz.pythonanywhere.com/>. Accessed 23 May 2020
- Wei F, Wang Y, Zong T (2014) Variable grouping based differential evolution using an auxiliary function for large scale global optimization. In: 2014 IEEE congress on evolutionary computation (CEC). IEEE, pp 1293–1298
- Yang M, Omidvar MN, Li C, Li X, Cai Z, Kazimipour B, Yao X (2016) Efficient resource allocation in cooperative co-evolution for large-scale global optimization. *IEEE Trans Evolut Comput* 21:493–505
- Zhang FZ (2019) High-accuracy method for calculating correlated color temperature with a lookup table based on golden section search. *Optik*. <https://doi.org/10.1016/j.ijleo.2019.163018>
- Zhao S-Z, Suganthan PN, Das S (2011) Self-adaptive differential evolution with multi-trajectory search for large-scale optimization. *Soft Comput* 15:2175–2185

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