Research Article

Differential evolution based global best algorithm: an efficient optimizer for solving constrained and unconstrained optimization problems

Mert Sinan Turgut[1] · Oguz Emrah Turgut[2]

Received: 7 October 2019 / Accepted: 4 March 2020 / Published online: 10 March 2020

Abstract
This study proposes an optimization method called Global Best Algorithm for successful solution of constrained and unconstrained optimization problems. This propounded method uses the manipulation equations of Differential Evolution, dexterously combines them with some of the perturbation schemes of Differential Search algorithm, and takes advantages of the global best solution obtained on the course of the iterations to benefit the productive and feasible in the search span through which the optimum solution can be easily achieved. A set of 16 optimization benchmark functions is then applied on the proposed algorithm as well as some of the cutting edge optimizers. Comparative study between these methods reveals that GBEST has the ability to achieve more competitive results when compared to other algorithms. Effects of algorithm parameters on optimization accuracy have been benchmarked with some high-dimensional unimodal and multimodal optimization test functions. Five real world design problems accompanied with three challenging test functions have been solved and verified against the literature approaches. Optimal solution obtained for economic dispatch problem also proves the applicability of the proposed method on multidimensional constrained problems with having large solution spaces.

Keywords Constrained optimization · Differential evolution · Differential search · Economic dispatch problem · Optimization

1 Introduction

Optimization plays a great role in real world problems as there are many industries and companies deals with finding optimum design of their products. There are plenty of applications of optimization problems in the world ranging from economics to engineering that should be solved in a reasonable amount of time within a plausible precision in order to reliably answer the needs of the society as well as to tackle the complexities that are inherent in any kind of real world design problem. Most of the real world problem involves binding and conflicting design constraints that are formed under the effect of material properties and nonlinearities not only associated with design variables but also some restrictions related with the type of optimization objective such as maximum stress, maximum deflection, geometrical characteristics, etc. [1]. In addition, increasing problem dimension hampers finding optimum solution of the problem which is caused by the increase in the number of search domain in the

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s42452-020-2426-8) contains supplementary material, which is available to authorized users.

✉ Mert Sinan Turgut, sinanturgut@me.com | [1]Department of Mechanical Engineering, Faculty of Engineering, Ege University, 35100 Bornova, Izmir, Turkey. [2]Department of Industrial Engineering, Faculty of Engineering and Architecture, Izmir Bakircay University, Menemen, Izmir, Turkey.

solution space. However, classical optimization techniques fails to give suitable and rational results for these kind of problems due to their deficiency in coping with the local optimum points and indifferentiable paths of the search domain. Therefore, researchers have devoted themselves to search for other alternatives for successful solution. Under the category of the approximate based algorithms, metaheuristics can be said as a promising and favourable alternative for conventional methods.

Metaheuristic algorithms are literally divided into two different groups [2–4]: These are single solution and population based algorithms. In individual (single solution) based algorithms, a set of solution is generated randomly at the initial phase of the algorithm and candidate solutions are iteratively adjusted throughout the optimization process. Tabu Search [5], Hill climbing [6], and Iterated Local Search [7] are good examples of the algorithms that can be grouped in individual based optimization methods. They are all local search based techniques developed with a strong aim to jump over local optimum points in the search domain. Multiple solution based (population based) algorithms have such advantages over single solution ones that they can jump over the local optimum points on the search domain and easily settle on the promising areas through the communication between neighboured candidate solutions [2]. Therefore, it can be concluded they have considerably high exploration ability with having a relatively satisfying convergence speed as compared to individual solution based algorithms, which are better at exploitation [2]. Particle Swarm Optimization (PSO) [8], Gravitational Search Algorithm (GSO) [9], Harmony Search Algorithm (HS) [10], and Ant Colony Optimization (ACO) [11] are the most popular and widely quoted population based algorithms, those of which have been applied on variety of optimization tasks. However, they inherit some serious operational drawbacks that it requires huge number of iterations to acquire the optimum result which increases the computation time as well as imposing a considerable burden in terms of computational cost.

Differential Evolution (DE) [12] is a conventional population based metaheuristic optimization algorithm which has favorable exploratory and exploratory capabilities. The proposed algorithm in this paper is based on the Differential Evolution algorithm and many variants of the Differential Evolution algorithm is proposed in the literature. Li et al. [13] proposed modified versions of JADE and CoDE algorithms applied the algorithms on various unimodal and multimodal test functions. The authors compared the results of these modified algorithms with that of other non-DE algorithms and some other DE-variants. The results showed the superiority of the proposed modified algorithms. Mohamed et al. [14] introduced an enhanced Differential Evolution (EDDE) algorithm by slightly changing the mutation scheme of the algorithm. The authors applied the algorithm on twenty four test problems and five constrained engineering problems and compared the results with that of some other metaheuristic optimizers. The EDDE algorithm came up with more desirable results than the other optimizers. Mohamed and Mohamed [15] proposed an Enhanced Adaptive Guided Differential Evolution (EAGDE) algorithm that reduces the population size over the iterations according to a non-linear function. Moreover, a new rule is added to determine the initial population size of the algorithm which is correlated with the dimension size of the problem. The authors evaluated the effectivity of the algorithm by applying on CEC2013 benchmark problems and comparing the results with that of their non-DE optimizers. The results showed that the proposed algorithm outperformed other algorithms in terms of optimization performance. Polakova et al. [16] introduced a novel adaptive mechanism based on linear reduction of the population size and enables to increase or decrease the size during the search to the DE algorithm. The efficiency of this new method is assessed by comparing the DE variants with and without the adaptive mechanism on CEC2014 test suite. As a result of this paper, the proposed method proved its effectivity especially on more complex multimodal and hybrid problems. As can be seen, most recent papers in the literature either introduces a novel mutation scheme or adopts an adaptive population size changing strategy to increase the optimization capability of the Differential Evolution algorithm and the results proved the effectiveness of these approaches.

Swarm Intelligence (SI) based optimization algorithms are the most prevalent and prominent branch of population based algorithms. These kind of algorithms are occasionally based on the flocking behaviour of fishes, birds and collective movements of some nature colonies. In addition to previously mentioned and renowned ACO, PSO, and GSA optimizers, recently new emerged intelligent swarm based algorithms including Artificial Cooperative Search (ACS) [17], Hunting Search (HUNT) [18] can be a good example of SI based optimizers. These methods take the advantages of utilizing the previous experiences of the populations and benefit the collectivity resulted from the historical knowledge of population evolution. They generally have fewer adjustable algorithm parameters than those of evolutionary algorithms by which probing capacity of the algorithm is restricted to some extent [2].

Physics- based optimizers can also be categorized into metaheuristic algorithms. They usually simulate the nature inspired physical phenomenas to tackle the optimization task at hand. Galaxy based Search Algorithm (GBSA) [19], Charged System Search (CSS) [20], Big Bang–Big Crunch algorithm (BB–BC) [21], Ray optimization (RAY) [22], Colliding Bodies Optimization (CBO) [23] are some of the most applied and popular ones available in the literature. For instance, GBSA is based on the movements of the spiral arms of spiral galaxies to explore their neighbourhoods with using chaotic numbers in order to avoid local optimum points. CSS uses the fundamental equation of Coulomb's law and some preliminaries on Newtonian mechanics in order to obtain optimum solution of the problem. BB-BC predicates its establishment on one of the theories which is related to evolution phases of the universe. RAY is constructed on the Snell's law of refraction and manipulates decision variables of any optimization problem by virtue of this physical law. CBO algorithm is mainly based on the collisions between two moving particles and their corresponding momentum and energy equations.

There has been made plenty of numerical studies on developing hybrid metaheuristics in order to strengthen the robustness and accuracy of the whole hybrid algorithm. Some of the proposed hybrid methods given in the literature are PSO-HS [24], PSO-ABC [25], ACO-PSO [26], GA-GSA [27], KH-BBO [28], GWO-DE [29], KH-AB [30], ABC-DE [31]. Besides, researchers put forward different strategies on the grounds of mathematical operators to improve the search performance of metaheurstic algorithms. Applying local search mechanisms on associated parts of the algorithm has been frequently utilized strategy for hybridization [32–35], through which search efficiency of the algorithm is greatly enhanced. Chaotic maps have also been widely used for improving the stochasticity of the perturbation schemes in metaheuristic algorithms [36–40] by means of the chaotic numbers taking advantage of the ergodic and unpredictible behaviour of the dynamically generated chaotic sequences. Additionally, literature encompass human-interaction bases optimizers including Teaching Learning Based Optimization(TLBO) [41], Human behaviour based optimization (HBBO) [42], League Championship Algorithm [43], Passing Vehicle Search (PVS) [44], Social Based Algorithm (SBA) [45], Group Counseling Optimization (GCO) [46] etc.

A good and practical metaheuristic algorithm has the ability of maintaining a compromising balance between exploration and exploitation [47, 48]. Exploration phase contains perturbation equations responsible for exploring throughly the unvisited regions of the search domain by virtue of the randomized movements made by stochastic operators. Following that, exploitation phase come into practice in which promising areas of the search space are detailly investigated and benefited in order to promote the local search capacity and avoid premature convergence. In this study, we propose a new algorithm called Global Best Algorithm (GBEST) which make use of the global best solution obtained over the course of the iterations to perform stochastic searches around these fruitful regions with a view to explore solution space efficiently. By this way, it is intended to eliminate the tedious exploration phase of the algorithm and focus on exploitation phase to obtain possible optimum solution. Different perturbation schemes taken from Differential Evolution [12] and some equations inspired from Differential Search [49] algorithms are successively utilized and candidate solutions are manipulated based on the current global best solution vector. In addition, the merits of Logistic map based chaotic sequences have been sucessfully used instead of random numbers produced by uniform distribution to sustain stochastic nature of the proposed algorithm. Numerical results obtained from unconstrained and constrained real world optimization problems show that intensification (exploitation) based proposed GBEST can procure plausible and reasonable outcomes with respect to solution accuracy and efficiency. It is also shown that GBEST can cope with the nonlinearities and complexities inherent in the benchmark functions within a small amount of run time. The paper is organised as follows: Sect. 2 gives the detailed description about GBEST algorithm along with its corresponding pseudo-code. Section 3 reports the optimization results of unconstrained benchmark problems and makes discussion plaftorm on how variations of algorithm parameters affect the optimization performance. Section 4 gives the optimization results of the constrained real world design problems obtained from GBEST algorithm as well as literature optimizers. Finally, the paper is concluded with remarkable comments in Sect. 5.

2 Global best algorithm

This study investigates the applicability and accuracy of the Global Best Algorithm over the widely accepted constrained and unconstrained optimization problems. Global Best Optimizer is firstly proposed by the authors in [50] with an aim to optimally design heat pipes and spiral heat exchangers. However, detailed description of the algorithm phases and numerical investigations over widely known constrained and unconstrained problems have not been throughly analyzed yet. In this study, it is aimed to assess the performance of the GBEST algorithm with plenty of unconstrained problems. Moreover, in addition to the previously solved real world single and multi-objective optimization problems with binding constraints in [50], this study demonstrates the solution of a pack of constraint optimization problems those frequently utilized in testing the optimization performance of the literature optimizers. Contrary to the majority of the metaheuristic algorithm available in the literature, this method is based on the intensification of the global best solution reached through the iterations. Metaheuristic algorithms are generally based on two different perturbation phases in which solution vectors are manipulated on the grounds of the current best solution based schemes (promising solutions) or exploration of the unvisited paths of the solution space. These are called exploration (diversification) and exploitation (intensification) phases of the algorithm. Their successful interaction leads to more favourbale and promising results, however there is no clear explanation on how to successfully balance these two phenomenas. Exploration phase generally takes more time than that of the exploitation since more dedication is performed on diversification so as to discover the unexplored regions of the search domain. A naive approach of 50–50 balance between two search components would be a plausible initialization for the upcoming iterations. However, GBEST chooses to utilize local search strategies rather than global search schemes in order to increase the performance of the algorithm. Perturbation schemes inspired from Artificial Bee Colony [51], Differential Evolution [12], and Differential search [49] algorithms have been incorporated into the proposed method to further investigate the different paths of the search space and mostly to probe around the global best solution obtained so far. Another benefited term for the proposed optimization algorithm is chaotic random numbers. In order to maintain stochasticity throughout the iterations, algorithm takes advantages of chaotic sequences generated by Logistic map [52], which was proposed by the renowned biologist Robert May and demonstrates that how an ergodic chaotic behaviour can be constructed by simple dynamical equations, instead of the random numbers defined between 0 and 1 based on normal distribution. Through the Logistic map, very effective and distinctive number sequences can be produced. Sequential chaotic numbers are dynamically formulated by the following equation.

$$z(t+1) = 4 \times z(t) \times (1 - z(t)), \quad z(t) \in (0,1) \tag{1}$$

where $z(0) \notin \{0.0, 0.25, 0.5, 0.75, 1.0\}$. Algorithm is initiailized with formation of the D dimensional N elements, X and X_{old}, with the given procedure below,

Algorithm 1. Initialization procedure of the proposed GBEST algorithm

```
for i = 1 to N
    for j = 1 to D
        X_{i,j} = low_j + (up_j − low_j) × φ_{i,j}
        X_{old,i,j} = low_j + (up_j − low_j) × φ_{old,i,j}
    end
end
```

where X and X_{old} are the N-sized D dimensional matrices in which solution vectors are manipulated by means of the perturbation scheme; low_j and up_j are respectively lower and upper bounds of the jth decision variable; ϕ_1 and ϕ_2 are chaotic random number defined between 0 and 1 produced by the Logistic map. After the initialization process, all matrix elements of X_{old} are evaluated and best solution vector (G_{best}) are determined. At the first part of the algorithm matrix components of X_{old} are adjusted by the manipulation scheme given below

$$V_{i,j} = G_{best,j} + \left(2.0 \times \left(\phi_2 - 0.5\right)\right) \times \left(G_{best,j} - X_{old,i,j}\right) \tag{2}$$

where $V_{i,j}$ are the perturbed matrix elements under the effect of global best solution so far (G_{best}) and matrix components of $X_{old,i,j}$. Then, boundary control mechanism is come into practice in order to restrict the $V_{i,j}$ individuals into the prescribed boundaries based on the following method

Algorithm 2. Boundary control mechanism of the proposed GBEST algorithm

```
for i = 1 to N
    for i = 1 to D
        if ((V_{i,j} < low_j) || (V_{i,j} > up_j))
            V_{i,j} = low_j + (up_j − low_j) × rand(0,1)
        end
    end
end
```

where *rand(0,1)* is a uniform random number generated between 0 and 1. X_{old} solution matrix is updated by the following procedure,

Algorithm 3. Solution matrix update procedure

```
for i = 1 to N
    if (func(V_i) < func(X_{old,i}))
        for j = 1 to D
            X_{old,i,j} = V_{i,j}
        end
    end
end
```

After that, Global best solution is decided based on the new $X_{old,i,j}$ matrix, and this updated G_{best} vector is utilized in the second part of the first phase of the algorithm through the following scheme

$$X_{old2} = randperm(X_{old}) \tag{3}$$

$$V_{i,j} = G_{best,j} + \left(2.0 \times \left(\phi_3 - 0.5\right)\right) \times \left(X_{old,i,j} - X_{old2,i,j}\right) \tag{4}$$

where X_{old2} is built up by the *randperm()* function by which row elements of X_{old} matrix are shuffled. Utilization of X_{old2} matrix comprised of shuffled components in Eq. (4) considerably increases the solution diversity as it was previously benefited in Differential Search algorithm which suffers from slow convergence and incurs highly computational burden to reach the optimum solution. After numerical experiments, it is seen that the integration of the difference between two matrices (X_{old} and X_{old2}) into Eq. (4) enormously increases probing performance of the algorithm around the global best solution. Following that, boundary check and G_{best} solution update mechanisms are consecutively applied and first phase of the algorithm is completed.

In the second phase of the proposed algorithm, basic manipulation equations of the Differential Evolution [12] algorithm will be utilized to further exploit the advantages of the Global best solution. Moreover, advantages of the ensemble learning mechanism will be also benefited. This strategy was previously used in Particle Swarm Optimization [53, 54] and the variants of Differential Evolution algorithm [55–57]. Global best algorithm takes advantages of the variety of mutation schemes and control parameters of the Differential Evolution algorithm. Some mutation strategies are useful and applicable in global search mechanism [58] whereas some remaining others are effective in local search and prompt higher convergence capabilities [59]. Therefore, in order to prevent premature convergence and enhance the search capacity of algorithm, population individuals shall communicate with each other by means of the different mutation schemes of DE which are collectively utilized through ensemble learning. In this study, the most frequently used mutation strategies of *DE/best/1* [60] and *DE/best/2* [60] are combined into a single perturbation scheme with an aim to avoid being trapped in the local optimum points of the search domain. Mentioned mutation schemes can be formulated as

$$\text{DE/best/1}: \quad V_i = G_{best} + F \times \left(X_{r1} - X_{r2}\right) \tag{5}$$

$$\text{DE/best/2}: \quad V_i = G_{best} + F_1 \times \left(X_{r1} - X_{r2}\right) + F_2 \times \left(X_{r3} - X_{r4}\right) \tag{6}$$

Indices those took place in above equations r_1, r_2, r_3, r_4 are distinct integer numbers defined in the range between 1 and N, which are also different from current population index i. F is the positive valued algorithm parameter used for scaling different mutated solution vector and occasionally defined in [0.5,1.0] [60]. Many researchers have proposed different scaling factor values for different types of optimization problems. Initial value of $F=0.5$ or 0.6 would be a favourable choice as proposed in [12, 61] whereas [62] it was suggested that $F=0.9$ is a promising initial value. Apart from that, selection of $F>1$, for instance $F=1.2$, increases the chance of escaping from local optimum points however, a dramatic decrease is seen in convergence speed of the algorithm since larger difference between two trial solutions elongates the run time of the algorithm [62]. It is typical to use F in the range of 0.4 and 0.95 [62]. In this study, after numerical tests on variety of optimization problems which will be discussed in the following sections, F is defined as a uniform random number generated between 0 and 1, contrary to the proof in [63] about transforming F into Gaussian random number does not improve the solution accuracy of a basic DE. Classical DE uses target vector $X_i=\{x_{i,1}, x_{i,2},\ldots, x_{i,D}\}$ along with mutant vector $V_{i,j}=\{v_{i,1}, v_{i,2},\ldots, v_{i,D}\}$ to generate trial vector $U_i=\{u_{i,1}, u_{i,2},\ldots, u_{i,D}\}$ through the utilization of crossover operation. In literature, there are two different available crossover operators those are called binomial (uniform) and exponential operators (two point modulus) [64]. Most frequently used one is binomical operator, expressed by the following scheme

$$u_{i,j} = \begin{cases} v_{i,j} \ \ if \ (rand(0,1) \le CR) || \left(j = j_{rand}\right), & j = 1,2,\ldots,D \\ x_{i,j} \ otherwise & \end{cases} \tag{7}$$

In above equation, CR represents crossover probability that decides the number of design variables to be copied from mutant vector to trail vector; j_{rand} is an integer number defined in the range [1,D]. Algorithm condition ($j=j_{rand}$) makes at least one design variable in trial vector U_i different from the corresponding mutant vector X_i. Successful parameter tuning of CR is very crucial in manipulating the mutant vector. There is no clear compromise on assigning value of CR in literature. Generally, CR is set in the range [0,1]. Some researchers [12, 60, 61] asserted that higher value of CR increase the convergence ability while some others said that initial value of $CR=0.1$ would be a plausible initial value. However, it was seen that diversity of the trial solutions considerably reduces and therefore stagnation occurs when $F=1.0$ used. This value will be decided by trial-and-error according to the statistical results of optimization benchmark functions benefited in this study.

As it was previously mentioned, optimization performance of DE depends on a suitable mutation strategy and its corresponding mutation parameters. However, they are problem-specific and it requires different mutation strategies along with associated algorithm parameters for different kind of optimization problem. Moreover, many variants of DE are proposed in the literature to further improve the exploration and explotation capabilities of the algorithm [14–16, 64, 65].In the context of ensemble learning, the idea comes along that it would be beneficial to use different mutation strategies with different parameter settings to solve optimization problem instead of utilizing a single perturbation scheme as it has been traditionally used in a basic DE. With such an ensemble strategy, it is aimed to obtain fruitful offspring population by using the advantages of different mutation schemes and their related parameter settings those of which show different characteristics for any type of optimization problem. This behavior also leads to enhance the solution diversity of algorithm due to such integration shows distinctive performance on the course of iterations, which is also essential for not to be stuck in local optimas. This promising idea has been previously practiced through combining the available mutation schemes in the literature into a single perturbation scheme by many researchers with a view to ease the selection of appropiate mutation strategy for a specific optimization problem. For instance, Qin et al. [58] proposed Self-Adaptive Differential Evolution algorithm (SADE) in order to overcome the computational cost burden by employed trial-and-error strategy of selecting suitable mutation schemes accompanied with their related parameter settings. By using this strategy, algorithm parameters are iteratively adjusted and self-adapted on the grounds of the past experiences of the generated solutions. Gong et al. [66] put forward two different perturbation schemes of Differential Evolution with adaptive strategy namely called Probability Matching and Adaptive Pursuit. These schemes adaptively choose the most suitable strategy for optimization problem in the light of the impact history retained throughout the optimization process. Jia et al. [67] proposed an improved version of (μ + λ)-constrained differential evolution ((μ + λ)-CDE) to solve constrained optimization problems with an ensemble of mutation strategies comprising a novel archiving based adaptive tradeoff model and a new mutation strategy called "*current-to-rand/best/1*". When compared to these abovementioned optimizers, ensemble strategy proposed in GBEST algorithm seems to be simple, yet effective. This study assigns the mutation strategies to population members with the given procedure,

Algorithm 4. Mutation strategy selection for population members

for $i = 1$ *to* N
    if $(i \% 2 == 0)$
        Employ DE/best/1
    else
        Employ DE/best/2
    end
end

where *N* is the population size, *DE/best/1* and *DE/best/2* are the mutation schemes as respectively formulated in Eqs. (5) and (6). During the iterations, crossover operator (*CR*) and scale factor (*F*) is adjusted by the self-adaptation scheme proposed in Brest et al. [68] with a little modification in the usage of scale factor F. Throughout the optimization process, after exhaustive numerical investigations on literature benchmark problems, scale factor is defined as a uniformly distributed random number generated between 0 and 1. Initial value, produced in the range between 0 and 1, is given to *CR* and *τ* (which will be explicitly discussed in the following section). Then, a random number is generated between 0 and 1. If the produced value is smaller than*τ*, then a new *CR* is reset in the range [0,1] else initialized value of *CR* remains same. Following this phase, boundary control and feasibility check mechanisms will come into practice. Population individuals those exceeding prescribed boundaries are restricted with the evolutionary constraint handling scheme proposed by Gandomi and Yang [69]. The proposed constraint handling scheme also takes advantage of the global best solution and is formulated by the following procedure,

Algorithm 5. Constraint handling procedure of the algorithm

for $j = 1$ *to* D
    if $x_j < low_j$
        $x_j = \phi_1 \times low_j + (1 - \phi_1) \times G_{best,j}$
    else if $x_j > up_j$
        $x_j = \phi_2 \times up_j + (1 - \phi_2) \times G_{best,j}$
    end
end

where G_{best} represents the global best solution obtained on the course of iterations; $\phi_1$and ϕ_2 are chaotic random numbers generated by Logistic map; and low_j and up_j correspondingly denote lower and upper bounds of the search space. This scheme was previously utilized in hybrid Teaching Learning Differential algorithm in [70] and its usage in boundary control considerably improved the probing capacity of the whole method. Numerical investigations on benchmark problems show that utilization of this constraint handling scheme gives better results when it is used in the second phase of the algorithm rather than used in first phase.

3 Results of the numerical tests for unconstrained optimization benchmark functions

Among different alternatives, 16 optimization test functions have been selected comprised of unimodal and multimodal functions to assess the performance of the proposed Global best algorithm as it has been a common practice in algorithm design evaluation [13, 71, 72]. Unimodal functions have only one optimum point and have the ability of evaluating the intensification performance of the algorithm while multimodal functions have more than one optimum points which are useful and applicable in testing both exploration and exploitation capacity of any metaheuristic method. If it is to give some examples of unimodal test function, Sphere and Schwefel 2.22 are the prominent ones used in asssesment. Ackley, Rastrigin, and Griewank functions are good representatives of multimodal functions. Generally, these aforementioned multimodal functions have one optimum point while having plenty of local optima. A successful metaheuristic should pass over these trap points and reach the global optimum with a tangible accuracy. Optimization capability of the propounded GBEST method have been benchmarked against some of the well known optimizers of Moth-Flame Algorithm (MOTH-FLAME) [73], Multi-Verse Optimization (MVO) [4], Bat Algorithm (BAT) [74], Hunting Search (HUNT) [18], Quantum behaved Particle Swarm Optimization (QPSO) [75], Differential Search [49], Bird Mating Optimizer (BMO) [76], Intelligent tuned Harmony Search (ITHS) [77], Big Bang- Big Crunch Optimization (BB-BC) [21], Particle Swarm Optimization (PSO) [8] and Differential Evolution [12].

Initialization of the respective control parameters for each optimization algorithm is given as follows:

1. *Global best algorithm (GBEST)* Population size = 500, Maximum number of iteration = 100, Crossover Rate (CR) = 0.9, $\tau = 0.5$
2. *Moth-flame algorithm (MOTH-FLAME)* Population size = 20, Maximum number of iteration = 2500
3. *Multi-verse optimizer (MVO)* Populaton size = 20, Maximum number of iteration = 2500
4. *Bat Algorithm (BAT)* Population size = 10, Maximum number of iteration = 5000, Loudness(A) = 0.45, Pulse emission rate (r) = 0.5, Maximum frequency (Q_{max}) = 2.0, Minimum frequency (Q_{min}) = 0.0
5. *Hunting search (HUNT)* Hunting Group Size (HGS) = 20, Number of generation = 2500, Maximum movement toward the leader (MML) = 0.2, Iteration per epoch (IE) = 10, Maximum and minimum search radius of the hunter → Ra_{max} = 1e-2 Ra_{min} = 1e-7, Reorganization parameters → α = -0.1 β = 1.0.
6. *Quantum behaved Particle Swarm Optimization (QPSO)* Population size = 10, Number of generation = 5000, Social and cognitive factors → $c_1 = 2.0$ $c_2 = 2.0$, Contraction-expansion coefficient (w) is initialized as 1.0 and iteratively decreased to 0.5.
7. *Differential Search (DS)* Population size = 25, Number of generation = 2500
8. *Bird Mating Optimizer (BMO)* Population size = 200, Number of generation = 250
9. *Intelligent tuned Harmony Search (ITHS)* Harmony memory size = 20, Maximum number of iteration = 2500, Harmony memory consideration rate (HMCR) = 0.95
10. *Big Bang–Big Crunch Algorithm (BB-BC)* Population size = 25, Maximum number of iteration = 2000, Algorithm parameters → $\alpha = 0.4$ $\beta = 0.8$
11. *Differential Evolution (DE)* Population size = 10, Maximum number of generation = 5000, Scale factor (F) = 0.9, Crossover Rate (CR) = 0.5
12. *Particle Swarm Optimization (PSO)* Population size = 10, Maximum number of generation = 5000, Social and cognitive factors → $c_1 = 2.0$ $c_2 = 2.0$

For numerical experiment case in which 25,000 function evaluations have been considered and detailly discussed below, maximum number of iteration (generation) is taken as the half of the corresponding value given above and other parameters remain same.

Due to the stochastic and unpredictible characteristics of these optimizers, 30 algorithm runs along with separately applied 25,000 and 50,000 function evaluations have been considered. Table 1 compares the statistical results of theses algorithms obtained after 25,000 function evaluations. GBEST finds optimum solution of Schaffer, Pathologic, Sphere, Rastrigin, Griewank, Zakharov and Schwefel functions and outperforms the rest of the compared optimization methods (except Rosenbrock function) in terms of statistical results. Moreover, the worst result obtained from GBEST is much better than that of other optimizers for most of cases. It is also worth to mention that BAT performs the worst prediction performance over all optimization functions. Rosenbrock function, consisting of huge narrow parabolic shaped valleys, is a challenging unimodal benchmark problem and its complexity considerably increases with increasing dimensionality. GBEST fails to estimate the optimum solution of this function and is unable to exploit the promising and fruitful areas in the long valley in which optimum point is located. However, it is easy to see that optimization performance of GBEST is quite impressive in finding optimum solutions of unimodeal and multimodal problems. Even though algorithm is only based on exploitation, it can be said that it has both ability to escape from local entrapments and to explore the unvisited paths of the search domain. Table 2 depicts the Wilcoxon signed-rank test with a significance level of 5%. If the results of GBEST are better than that of the compared algorithm than the '+' sign is used, however, if the results are worse than the '-' sign is used. It can be seen from Table 2 that GBEST performs significanly better than the compared algorithms.

Table 1 Numerical results of optimization problems after 25,000 function evaluations (ranked according to obtained mean solution performances)

	GBEST	MOTH-FLAME	MVO	BAT	HUNT	PSO
Levy						
Min	3.18948E−09	4.08923E+00	4.55891E−01	1.90582E+01	2.61181E+01	8.95142E−02
SD	2.40747E−08	3.88619E+00	3.98991E+00	1.84446E+01	8.79653E+00	2.29033E+00
Mean	2.17777E−08	1.18124E+01	4.67925E+00	5.18735E+01	3.97521E+01	3.21198E+00
Max	9.47202E−06	1.80962E+01	1.80937E+01	1.58751E+02	6.29806E+01	1.00202E+01
Rank	1	8	5	12	10	4
	QPSO	DS	BMO	ITHS	BB-BC	DE
Min	6.38762E−01	3.37659E+00	3.08217E+00	3.11572E−05	2.40921E+01	4.12331E−05
SD	3.08278E+00	3.02487E+00	1.39562E+01	1.86232E−01	9.48277E+00	1.62279E−01
Mean	6.09871E+00	8.73962E+00	1.76341E+01	1.19836E−01	4.19762E+01	6.97447E−02
Max	1.91822E+01	1.56392E+01	5.87520E+01	8.74892E−01	6.38251E+01	4.54911E−01
Rank	6	7	9	3	11	2
	GBEST	MOTH-FLAME	MVO	BAT	HUNT	PSO
Step						
Min	1.13858E−12	8.61512E−12	2.08638E−04	1.04682E+01	4.33956E−10	1.22194E−10
SD	2.41982E−11	8.67212E−04	1.18692E−04	1.59171E+01	4.99724E−08	8.29552E−03
Mean	2.27131E−11	2.28209E−04	4.45150E−04	4.76492E+01	4.03716E−08	1.01160E−03
Max	1.49792E−10	4.08528E−03	7.43180E−04	7.38975E+01	2.04827E−07	6.99189E−02
Rank	1	4	5	12	2	6
	QPSO	DS	BMO	ITHS	BB-BC	DE
Min	6.22621E−02	1.33995E+00	5.26571E−02	2.65281E−06	3.78754E−04	1.12829E−07
SD	2.56203E+00	3.26602E+00	1.58927E−01	9.75921E−02	5.78955E−01	4.93800E−06
Mean	1.81342E+00	7.50937E+00	2.39649E−01	5.37927E−02	1.53816E−01	5.30992E−06
Max	9.38905E+00	1.54182E+01	9.62891E−01	4.89267E−01	3.01076E+00	2.86827E−05
Rank	10	11	9	7	8	3
	GBEST	MOTH-FLAME	MVO	BAT	HUNT	PSO
Penalized1						
Min	9.02038E−11	9.04329E−11	4.41484E−06	4.35913E−01	9.35827E−10	9.36157E−11
SD	4.60452E−11	4.27551E−02	7.50536E−06	3.26829E−01	6.08872E−02	1.43048E−06
Mean	4.38997E−11	1.58102E−02	1.40088E−05	1.14504E+00	6.04771E−02	2.84062E−07
Max	2.17224E−10	2.07521E−01	3.94412E−05	1.63511E+00	2.07628E−01	8.60771E−06
Rank	1	6	4	12	8	3
	QPSO	DS	BMO	ITHS	BB-BC	DE
Min	2.57119E−06	5.32712E−02	1.14823E−03	1.03920E−06	1.94826E−01	1.35867E−08
SD	6.99551E−02	9.99028E−02	9.48927E−02	3.00242E−03	4.59812E−01	1.47037E−07
Mean	4.31923E−02	2.67916E−01	6.97652E−02	1.08962E−03	8.08661E−01	1.12233E−07
Max	2.07661E−01	4.98562E−01	3.70753E−02	2.14247E−02	2.26924E+00	9.33494E−07
Rank	7	10	9	5	11	2
	GBEST	MOTH-FLAME	MVO	BAT	HUNT	PSO
Min	0.00000E+00	5.62814E−03	1.41024E−03	5.72027E+01	7.94621E−08	1.62513E−04
SD	9.23083E−06	2.18926E+00	1.29222E−03	6.31184E+06	2.62782E−06	6.48869E−01
Mean	2.18139E−06	9.58261E−01	3.68926E−03	2.15442E+06	2.61827E−06	2.32730E−01
Max	4.99000E−05	9.60972E+00	6.98639E−03	3.30052E+07	1.23882E−05	2.61112E+00
Rank	1	5	3	12	2	4

Table 1 (continued)

	QPSO	DS	BMO	ITHS	BB-BC	DE
Min	6.15708E−01	1.10532E+01	1.60931E+00	3.51972E−03	4.48721E+01	5.06852E+00
SD	9.36990E+00	1.13249E+01	5.20496E+00	3.19735E+00	5.48726E+01	7.66137E+00
Mean	1.08972E+01	3.33027E+01	9.07215E+00	1.95826E+00	1.17829E+02	1.83057E+01
Max	6.28263E+01	6.70862E+01	2.57522E+01	2.26365E+01	3.15698E+02	4.55251E+01
Rank	8	10	7	6	11	9
	GBEST	MOTH-FLAME	MVO	BAT	HUNT	PSO
Ackley						
Min	4.44089E−16	6.82462E−10	3.68927E−02	1.66282E+00	1.66613E−05	2.68294E−10
SD	1.02434E−15	4.08921E−03	9.16832E−03	9.86241E−01	1.38726E−04	1.71220E−04
Mean	6.98461E−15	7.31342E−04	5.51827E−02	1.97972E+01	1.42442E−04	4.45187E−05
Max	4.50826E−14	2.42241E−02	8.73921E−02	2.09913E+01	6.59821E−04	1.10259E−03
Rank	1	5	6	12	4	3
	QPSO	DS	BMO	ITHS	BB-BC	DE
Min	6.50281E−02	5.94246E+00	2.19271E−01	1.87492E−07	7.17929E−02	1.20350E−07
SD	6.80261E−01	1.17551E+00	3.39282E−01	2.38361E−01	9.69826E−03	6.53952E−07
Mean	6.80719E−01	8.62031E+00	6.37957E−01	2.07691E−01	8.79281E−02	7.75991E−07
Max	3.57927E+00	1.13832E+01	1.46703E+00	1.00178E+00	1.12682E−01	2.64013E−06
Rank	10	11	9	8	7	2
	GBEST	MOTH-FLAME	MVO	BAT	HUNT	PSO
Min	0.00000E+00	2.66930E−12	1.06421E−04	1.03826E+00	1.20104E−10	1.41248E−10
SD	1.04923E−08	6.50721E−02	7.68261E−03	2.50692E−02	8.31927E−03	8.26684E−03
Mean	5.81942E−09	1.98264E−02	8.43826E−03	1.09759E+00	7.69271E−03	7.56599E−03
Max	3.99548E−08	3.49777E−01	2.75281E−02	1.14927E+00	3.03892E−02	4.92628E−02
Rank	1	6	5	12	4	3
	QPSO	DS	BMO	ITHS	BB-BC	DE
Min	7.71592E−03	4.12153E−01	1.92242E−02	2.34321E−06	1.61489E−04	4.51502E−07
SD	5.69271E−02	1.16002E−01	3.33713E−02	5.59271E−02	3.30217E−02	4.65397E−03
Mean	7.11192E−02	7.33304E−01	7.33208E−02	2.49995E−02	3.02823E−02	3.17156E−03
Max	1.84927E−01	9.10726E−01	1.68263E−01	2.98541E−01	1.40671E−01	1.48457E−02
Rank	9	11	10	7	8	2
	GBEST	MOTH-FLAME	MVO	BAT	HUNT	PSO
Rastrigin						
Min	0.00000E+00	4.17892E+01	4.48261E+01	7.26432E+01	8.67212E+01	2.31306E+01
SD	2.56755E−05	1.94897E+01	1.96382E+01	4.12619E+01	1.90281E+01	9.72560E+00
Mean	1.14846E−05	7.97251E+01	8.18262E+01	1.99275E+02	1.28736E+02	4.47602E+01
Max	1.04725E−04	1.35927E+02	1.23864E+02	2.78201E+02	1.62937E+02	7.86057E+01
Rank	1	5	6	11	8	2
	QPSO	DS	BMO	ITHS	BB-BC	DE
Min	2.42836E+01	1.24826E+02	8.12836E+01	1.56792E+00	1.34826E+02	6.42917E+01
SD	1.68263E+01	1.38729E+01	2.58361E+01	2.69831E+01	4.18261E+01	2.11569E+01
Mean	5.42262E+01	1.56927E+02	1.31836E+02	6.68268E+01	1.99825E+02	1.12742E+02
Max	9.49689E+01	1.88982E+02	1.90738E+02	1.27937E+02	2.82708E+02	1.51792E+02
Rank	3	10	9	4	12	7

Table 1 (continued)

	GBEST	MOTH-FLAME	MVO	BAT	HUNT	PSO
Rosenbrock						
Min	2.63398E+01	1.81534E+01	2.35637E+01	1.69172E+01	1.69283E+01	2.05962E+01
SD	3.55412E−01	2.40753E+01	1.70542E+00	3.01985E+01	3.39278E+00	1.82634E+01
Mean	2.71827E+01	4.27659E+01	2.75581E+01	6.02899E+01	2.38751E+01	3.32698E+01
Max	2.76698E+01	8.48261E+01	2.98630E+01	1.40271E+02	2.92038E+01	8.17595E+01
Rank	3	6	4	8	1	5
	QPSO	DS	BMO	ITHS	BB-BC	DE
Min	3.19273E+01	1.00273E+02	3.04086E+01	8.27903E+00	2.64022E+01	2.03280E+01
SD	3.02769E+01	3.67302E+01	4.77262E+01	4.92732E+01	4.80271E+01	2.50203E+00
Mean	8.19964E+01	1.66927E+02	9.58398E+01	5.93722E+01	8.07728E+01	2.64448E+01
Max	1.57927E+02	2.76945E+02	2.56927E+02	1.84998E+02	2.22086E+02	3.89021E+01
Rank	10	11	12	7	9	2
	GBEST	MOTH-FLAME	MVO	BAT	HUNT	PSO
Sphere						
Min	0.00000E+00	2.07134E−11	2.05613E−04	1.40827E+01	4.40672E−10	1.65197E−10
SD	5.86922E−10	9.52902E−04	1.39629E−04	1.49485E+01	8.33789E−08	2.96927E−03
Mean	2.53989E−11	2.39828E−04	4.84872E−04	5.56046E+01	5.47921E−08	5.05793E−04
Max	3.36978E−09	5.31975E−03	8.48028E−04	8.38749E+01	4.02975E−07	2.19092E−02
Rank	1	4	5	12	2	6
	QPSO	DS	BMO	ITHS	BB-BC	DE
Min	1.99262E−02	3.57542E+00	6.04821E−02	1.09482E−07	3.65172E−04	1.19928E−06
SD	5.48721E−02	2.28231E+00	2.05954E−02	8.07658E−02	1.27291E−01	6.76450E−06
Mean	4.38628E−01	7.79008E+00	2.80965E−01	5.30281E−02	2.65989E−02	6.32970E−06
Max	2.27885E+01	1.47269E+01	1.07335E+00	5.98886E−01	7.04372E−01	4.09158E−05
Rank	10	11	9	8	7	3
	GBEST	MOTH-FLAME	MVO	BAT	HUNT	PSO
Alpine						
Min	1.72645E−12	1.04454E−07	9.40972E−01	7.70886E+00	1.20487E+00	1.38255E−04
SD	4.79083E−05	1.16682E−01	1.43798E+00	6.45542E+00	1.03770E+00	1.94799E−01
Mean	2.09326E−05	3.35826E−02	2.38965E+00	2.28008E+01	2.85975E+00	7.33890E−02
Max	2.43352E−04	5.00068E−01	6.86891E+00	3.70954E+01	5.91628E+00	1.37372E+00
Rank	1	2	7	12	8	3
	QPSO	DS	BMO	ITHS	BB-BC	DE
Min	1.46892E−02	7.65322E+00	2.10582E+00	5.69921E−05	1.92677E+00	3.72118E−02
SD	4.13889E−01	1.89341E+00	4.51989E+00	1.56692E+00	3.00284E+00	3.73415E+00
Mean	3.32997E−01	1.26543E+01	8.37854E+00	7.43871E−01	8.80635E+00	4.68349E+00
Max	2.04672E+00	1.65799E+01	2.39871E+01	7.90338E+00	1.44027E+01	1.19891E+01
Rank	5	4	10	6	11	9
	GBEST	MOTH-FLAME	MVO	BAT	HUNT	PSO
Salomon						
Min	5.36421E−11	6.99821E−02	2.99982E−01	8.99828E−01	1.19987E+00	3.99875E−01
SD	4.32817E−02	1.06725E−01	5.69821E−02	1.55123E−01	9.49826E−02	7.62623E−02
Mean	2.49725E−02	9.02628E−01	3.57886E−01	1.48921E+00	1.41893E+00	4.70704E−01
Max	9.98261E−02	1.19982E+00	4.99871E−01	1.79826E+00	1.59826E+00	6.99873E−01
Rank	1	9	3	11	10	7

Table 1 (continued)

	QPSO	DS	BMO	ITHS	BB-BC	DE
Min	3.99827E−01	2.99817E−01	2.07261E−01	9.98827E−02	1.09987E+00	2.99873E−01
Std.dev.	8.65292E−02	4.97261E−02	5.04821E−02	1.49826E−01	2.79271E−01	5.32270E−02
Mean	5.69291E−01	3.79271E−01	3.79762E−01	2.82635E−02	1.76954E+00	3.99564E−01
Max	7.99827E−01	4.99817E−01	4.99933E−01	8.98827E−01	2.49987E+00	4.99948E−01
Rank	8	4	5	2	12	6
	GBEST	MOTH-FLAME	MVO	BAT	HUNT	PSO
Pathologic						
Min	0.00000E+00	1.38235E−01	1.99827E−01	5.96261E−01	3.14826E+00	5.24951E−02
SD	2.54398E−04	2.13827E−01	2.02281E−01	2.68977E+00	6.29271E−01	3.96342E−01
Mean	7.36792E−05	4.39295E−01	5.57829E−01	3.34261E+00	4.10282E+00	5.39146E−01
Max	9.54678E−04	9.29984E−01	1.00968E+01	7.01082E+00	5.17261E+00	1.63070E+00
Rank	1	2	4	11	12	3
	QPSO	DS	BMO	ITHS	BB-BC	DE
Min	5.70927E−01	2.00158E+00	3.14771E+00	6.01622E−01	6.29162E−01	1.89982E+00
SD	8.28971E−01	4.12592E−01	4.62811E−01	3.89217E−01	5.33896E−01	1.87499E−01
Mean	2.19864E+00	3.00017E+00	4.22370E+00	1.50721E+00	1.29989E+00	2.23690E+00
Max	3.75248E+00	3.63827E+00	5.04821E+00	2.38825E+00	2.76318E+00	2.57867E+00
Rank	7	9	10	6	5	8
	GBEST	MOTH-FLAME	MVO	BAT	HUNT	PSO
Mishra01						
Min	2.59128E+00	9.83975E+00	3.94162E+00	2.98166E+00	4.58943E+00	7.42999E+00
SD	5.04145E−01	6.80104E+02	1.36152E+01	2.21542E+10	2.62891E+08	4.44698E+04
Mean	3.24188E+00	2.82903E+02	1.25359E+01	4.47628E+09	1.76013E+08	2.84309E+04
Max	6.05283E+00	3.88415E+03	1.07562E+02	1.15245E+11	2.88892E+10	2.96207E+05
Rank	1	6	5	12	11	7
	QPSO	DS	BMO	ITHS	BB-BC	DE
Min	2.96341E+00	3.19342E+04	6.72612E+02	3.65876E+00	1.62428E+03	4.17927E+00
SD	2.68746E+00	4.59555E+07	7.99953E+05	3.33278E+00	8.43620E+08	1.29725E+00
Mean	4.05179E+00	1.66040E+07	8.59265E+06	8.26900E+00	1.25871E+08	5.89581E+00
Max	1.19207E+01	3.61471E+08	1.29064E+08	2.04722E+01	6.90001E+09	3.49462E+01
Rank	2	9	8	4	10	3
	GBEST	MOTH-FLAME	MVO	BAT	HUNT	PSO
Schfewel 04						
Min	6.86901E−08	1.07226E−02	7.88692E−02	3.55914E+03	8.56520E−04	2.98377E+00
SD	2.23139E−02	1.31832E−01	4.45835E−02	4.39962E+03	3.37864E−01	3.78135E+01
Mean	2.10863E−02	1.04468E−01	1.35726E−01	1.00262E+04	1.57534E−01	3.63861E+01
Max	1.16792E−01	7.32916E−01	2.62512E−01	3.61497E+04	1.25408E+00	2.17791E+02
Rank	1	2	3	12	4	6
	QPSO	DS	BMO	ITHS	BB-BC	DE
Min	2.12543E+01	3.02317E+03	3.30156E+00	4.73496E+00	7.70132E+01	1.79261E+01
SD	1.11032E+02	6.47307E+02	2.67352E+01	1.25854E+01	8.93631E+02	1.22965E+02
Mean	1.14747E+02	4.19862E+03	4.16729E+01	2.27765E+01	1.31332E+03	1.45430E+02
Max	6.80196E+02	5.61173E+03	1.00892E+02	5.54726E+01	4.25911E+03	8.85479E+02
Rank	8	11	7	5	10	9

Table 2 Wilcoxon signed ranked test based on mean objective function error values for 25,000 function evaluations with a significance level of α = 0.05

Problem	Pairwise comparison with GBEST algorithm										
	MOTH-FLAME	MVO	BAT	HUNT	PSO	QPSO	DS	BMO	ITHS	BB-BC	DE
f_1 Levy	+	+	+	+	+	+	+	+	+	+	+
f_2 Step	+	+	+	+	+	+	+	+	+	+	+
f_3 Penalized1	+	+	+	+	+	+	+	+	+	+	+
f_4 Zakharov	+	+	+	+	+	+	+	+	+	+	+
f_5 Ackley	+	+	+	+	+	+	+	+	+	+	+
f_6 Griewank	+	+	+	+	+	+	+	+	+	+	+
f_7 Rastrigin	+	+	+	+	+	+	+	+	+	+	+
f_8 Rosenbrock	+	+	+	−	+	+	+	+	+	+	−
f_9 Sphere	+	+	+	+	+	+	+	+	+	+	+
f_{10} Alpine	+	+	+	+	+	+	+	+	+	+	+
f_{11} Salomon	+	+	+	+	+	+	+	+	+	+	+
f_{12} Pathologic	+	+	+	+	+	+	+	+	+	+	+
f_{13} Mishra01	+	+	+	+	+	+	+	+	+	+	+
f_{14} Schwefel04	+	+	+	+	+	+	+	+	+	+	+

4 Numerical benchmark on constrained optimization problems

In this section, minimization (maximization) performance of the proposed Global Best Algorithm will be tested on harshly constrained optimization problems. In order to validate and assess the performance of the proposed optimization method, five real world engineering design problems along with three strictly constrained widely known test functions will be solved by GBEST and optimization results will be compared with literature studies in terms of statistical analysis. Exact mathematical formulations of these design problems, which are not given in here due to the space restrictions, can be found in [78, 79]. Due to the stochastic nature of the proposed optimizer, 50 algorithms runs along with 100,000 function evaluations are made for each test function. Maximization problems are directly converted into minimization problems by *−f(x)*. All equality constraints are transformed into inequality constraints with mathematical representation of $|h(x) - \delta| \leq 0$ using the degree of violation $\delta = 1.0\mathrm{E}-10$. Constrained handling is maintained by the basic penalty function which converts constrained optimization problem into unconstrained one. In order to account for the equality (φ) and inequality (ψ) constraints taking place in the constrained test problems, following mathematical expression is utilized

$$F(\alpha, \beta, x) = f(x) + \sum_{i=1}^{M} \alpha_i \varphi_i^2(x) + \sum_{j=1}^{N} \beta_j \psi_j^2(x) \tag{8}$$

where $1 \leq \beta$ and $0 \leq \alpha$. The terms α and β are penalty coefficients which penalize unfeasible solutions obtained on the course of iterations.

4.1 Constrained problem 9: multidimensional economic dispatch

Economic Load Dispatch (ELD) problem is not only the one of the most applied optimization benchmark case in multidimensional optimization, but also an important issue in power system control engineering. The problem itself deals with obtaining optimum power generation rates in order to acquire possible minimum total cost whilst satisfying a bunch of system related equality and inequality constraints. Taking into account of the canonical mathematical formulation of the basic economical load dispatch problem, generation costs of power units can be represented by quadratic functions those of which can be characteristically solved by traditional optimization techniques such as gradient based method [80] and the lambda iteration method [81]. In spite of being a simple form in mathematical aspects, real world applications of these kind of problems inherits modelling of non-smooth and non convex curves those are occured due the valve point effects and some discontinuities caused by the prohibited operating zones [82]. These complications make successful solving of economic dispatch problem harder than ever for the gradient based and mathematical programming methods since these methods were constructed for smooth and continuous type optimization objectives. Among the majority of the solution options for these problems, metaheuristc algorithms comes one step forward due to their success in dealing with the complexities and difficulties which are posed by the non-convex, non-differentiable, and non-continuos form of the objective function of ELD problem. Plenty of metaheuristic algorithms, considerable amount of which are recently emerged optimizer including Bat algorithm [83], Firefly algorithm [84], Differential Search [85], Social Spider Algorithm [86] have been applied on ELD problems and it is seen that these kind of algorithms can successfully cope with the extreme non linearities occured by the abovementioned restrictions. Here in this study, the proposed Global Best Algorithm is utilized to retain the optimum scheduling of power generation units, and this method is meticilously benchmarked against some of the literature optimizers which were previously tested on ELD problems. Mathematical formulation of the problem, expressed on the grounds of a quadratic approximation of the output power obtained from generating units, is represented by the following equation [87]

$$F = \sum_{i=1}^{N} a_i P_i^2 + b_i P_i + c_i \tag{9}$$

where F is the total generation cost in MW; P_i is the output power of ith generation unit in MW; N is the total number of generation units to be optimally scheduled; and a_i, b_i, and c_i are the coefficients related to the cost function. Ripple like effect occured in the valve opening process of multivalve steam turbines generates highly nonlinear heat rate curve. This complexity can be refined through the utilization of the periodic sine function. That is to say, valve point effects are included in total quadratic cost function by means of the sinusoidal terms which are formulated by the equation given below.

$$f_i(P_i) = a_i P_i^2 + b_i P_i + c_i + \left| e_i \sin\left(f_i\left(P_i^{\min} - P_i \right) \right) \right| \tag{10}$$

where e_i and f_i are the coefficients concerning the valve point loading effect. Problem at hand is restricted by some of the system imposed constraints which will be described in the following sections.

4.1.1 Power generation constraints

This system constraint considers the total power load demand as well as the losses occured through the network transmission process. In order to compute the transmission network losses, B coefficient method [88], widely known and highly appreciated by the power industry, is applied. Power balance constraint is expressed as:

$$\sum_{i=1}^{N} P_i = P_{Demand} + P_{Loss} \tag{11}$$

where P_{Demand} stands for total power demand and P_{Loss} represents transmisson losses calculated by the aforementioned B coefficient method as formulated below

$$P_{Loss} = \sum_{i=1}^{N} \sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1}^{N} B_{0i} P_i + B_{00} \tag{12}$$

In addition, it should be emphasized that power generation rates shall be restricted within predetermined limits such as:

$$P_i^{\min} \le P_i \le P_i^{\max} \tag{13}$$

Ramp rate limits are taken into consideration not to exceed prescribed output charges, determined and restricted by the physical characteristics of the generation units. Power generation rates may be varied according to the ramp rate limits. That is, if power generation rates are inclined to increase, then it becomes

$$P_i - P_i^{prev} \le P^{UR_i} \tag{14}$$

Else if power generation rates are in tendency to fall, it becomes

$$P_i^{prev} - P_i \le P^{DR_i} \tag{15}$$

where the term P_i^{prev} symbolizes previous power charge of i^{th} unit and P^{URi} and P^{LRi} are respectively ramp up and ramp down limits. The contribution of these terms modifies the constraint handling process of generation units presented in Eq. (12) to the expression that is given below

$$\max\left(P_i^{\min}, P_i^{prev} - P^{DR_i} \right) \le P_i \le \min\left(P_i^{\max}, P_i^{prev} + P^{UR_i} \right) \tag{16}$$

Prohibited operating zones (POZ) constraints consider some restrictions brought about by the physical limitations of the system components including steam valves and shaft bearings [82]. As a result of these restrictions, some discontinuities emerge on the cost curves which result in prohibited operating zones. Therefore, it is a must to avoid these operational areas in order to produce economical outputs. All in all, feasible operation range of unit i can be formulated by the following terms

$$P_i \in \begin{cases} P_i^{\min} \le P_i \le P_{i,1}^{low} \\ P_{i,k-1}^{up} \le P_i \le P_{i,k}^{low} \\ P_{i,m}^{up} \le P_i \le P_i^{\max} \end{cases} \quad k = 2, 3, .., m, \quad i = 1, 2, \dots, N \tag{17}$$

where m represents the total number of the prohibited zones of unit i; $P_{j,k}^{low}$ and $P_{j,k}^{up}$ are correspondingly the upper and lower output limits of the kth prohibited zone of the ith generator.

In this study, to test the convergence ability and the solution accuracy of the proposed GBEST algorithm, two different case studies with having highly correlated equations have been solved. Results obtained from the GBEST have been benchmarked against the outcomes of the literature optimizers. Due to the stochastic nature and unpredictible search characteristics of the Global Best Algorithm, 50 consecutive algorithm runs have been performed along with varying function evaluations which depends on the multidimensionality of the test case. Algorithm has been run on Java on a personal computer having 3.0 GHz processor with 4.0 GB RAM. In the light of the past experience on the assessment of the solution accuracy of any metahuristic method, economic load dispatch problem with having 38 and 40 generation units have been selected as a benchmark case and algorithm performance have been evaluated through the statistical results obtained after sequential algorithm runs.

4.1.2 Case 1: 38 generation units

A system with 38 generation unit with 6000 MW total power demand is selected as a test case for this study. Cost curve characteristics and data input are taken from Sinha et al. [89]. On the grounds od the complex solution space, this case can evaluate the solution accuracy and the corresponding robustness of the algorithm. Table 3 compares the statistical results of the literature algorithms accompanied with the outcomes of the GBEST methods and abovementioned algırithms in terms of minimum, maximum, and average costs. GBEST algorithm is so consistent that it finds the same results on the course of 50 algorithm runs with an optimum cost of 9,417,209.004 ($/h). Figure 1 depicts the convergence behaviour the GBEST algorithm for 38 units test case concerning the best, worst and mean solution. It can be clearly observed that after a steep decrease at the initial phase of the iterations, algorithm converges to its optimum solution after 99,823 function evaluations.

Table 3 Statistical results for 38 generation systems

	Minimum cost ($/h)	Maximum cost ($/h)	Average cost ($/h)
SPSO [90]	9,543,984.77	N/A	N/A
PSO-Crazy [90]	9,520,024.60	N/A	N/A
NPSO [90]	9,516,448.31	N/A	N/A
PSO-TVAC [90]	9,500,448.30	N/A	N/A
DE-BBO [91]	9,417,235.78	N/A	N/A
BB-BC	9,596,038.65	9,954,873.50	9,735,223.85
DS	9,709,432.88	10,077,919.06	9,861,247.70
ITHS	9,492,756.85	9,860,606.40	9,721,820.17
QPSO	9,450,725.34	9,984,268.57	9,668,388,89
MVO	9,449,770.68	9,504,343.70	9,477,619.56
MOTH-FLAME	9,547,870.37	9,922,020.33	9,654,828.85
GBEST	9,417,209.00	9,417,209.00	9,417,209.00

N/A not available

4.1.3 Case 2: 40 generation units

The economic dispatch problem with forty generation units is selected as another benchmark case for this stıdy. Problem itself involves the effect of valve point loading which increase the total number of local optimum points in the solution space. Total power load demand for this case 10,500 MW. System input data are obtained from Sinha et al. [89]. Table 4 compares the robustness of the GBEST algorithm with the state-of-art optimization algorithms. It is seen that the consistency and the accuracy of the GBEST is much better than those of the compared algorithms as it attains the same optimum result through 50 algorithm evaluations. Figure 2 show the evolution history of the GBEST algorithm for 40 unit system. Followed by the sharp decrease at the early phase of the iterations, algorithm reaches the optimum solution after 20,000 function evaluations which is very efficient and effective when compared to the evolution characteristics of the literature optimizers.

5 Conclusion

This study proposes Global Best Algorithm (GBEST) for solving constrained and unconstrained optimization problems. GBEST uses the merits of the perturbation schemes of the Differential Evolution algorithm, blends these equations with some of the search equations inspired by Differential Search algorithm, and benefits the so far obtained global best solution in order to exploit the productive and fertile areas in the search span. Influences of algorithm parameters on the optimization performance have been successfully benchmarked against some of the widely known and applied optimization test functions. Based on the results, suitable parameter ranges have been suggested. A set of 16 test functions have been applied on GBEST as well as relatively new emerged literature optimizers. Results reveal that optimization outcomes of the proposed GBEST are those of the compared state-of-art algorithms. In order to get a further information on the constrained cases, five real world design problems along with three binding constrained problems have been solved. GBEST is very competitive and much superior over the literature optimizers. Finally, GBEST is applied on multidimensional economic dispatch problem with

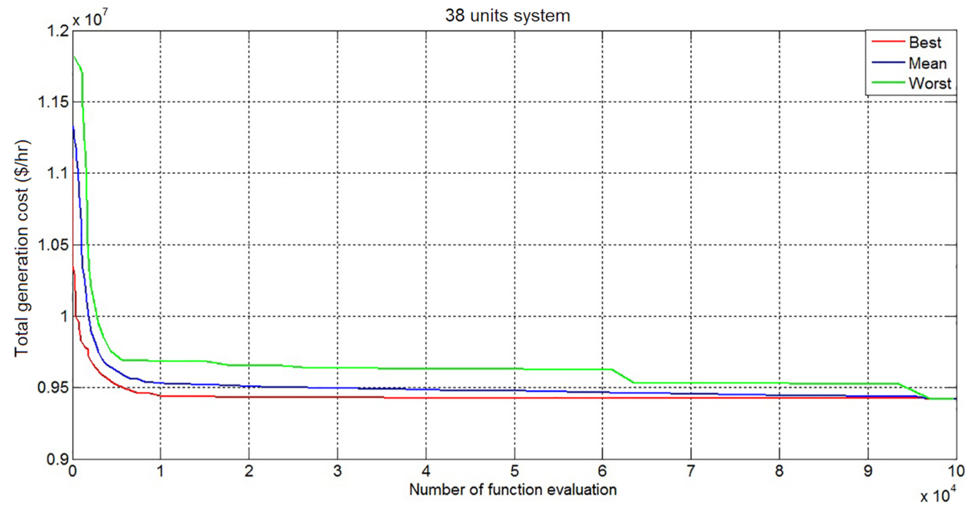

Fig. 1 Evolution of the objective function for 38 units system

Table 4 Comparison of the literature studies in terms of statistical results for 40-generator system

	Minimum cost ($/h)	Avarage cost ($/h)	Maximum cost ($/h)
TDE [92]	121,552.3516	121,708,0739	122,056.6991
SOH_PSO [93]	121,501.140	121,853.570	122,446.300
TSARGA [94]	121,463.070	122,928.310	124,296.540
IDE [95]	121,442.2682	121,448.8196	121,457.2746
IA_EDP [96]	121,436.9729	121,492.7018	121,648.4401
BBO [97]	121,426.9530	121,508.0325	121,688.6674
RCGA [98]	121,418.7224	121,685.9971	121,921.6589
CSOMA [99]	121,414.6978	121,415.0479	121,417.8045
DEPSO [100]	121,412.56	121,419.31	121,468.25
GBEST	121,412.5354	121,412.5354	121,412.5354

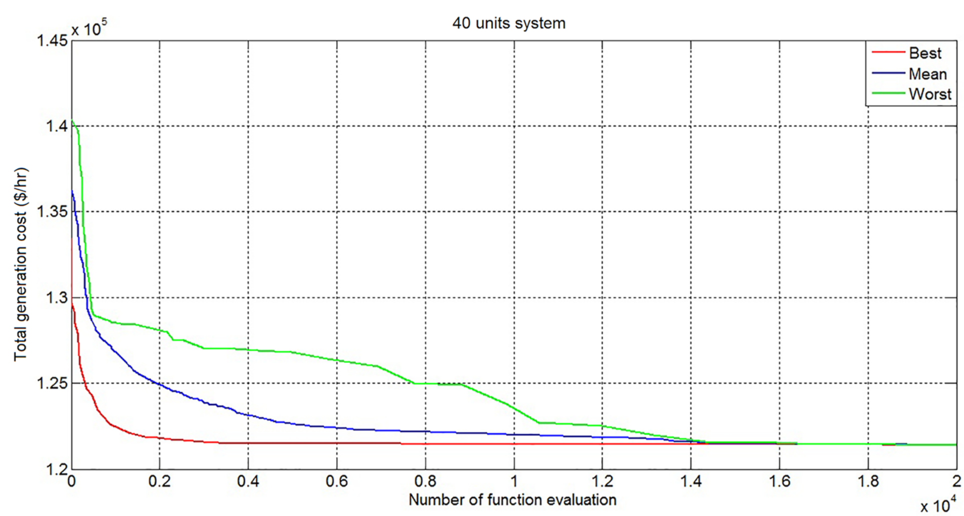

Fig. 2 Convergence characteristics of the objective function for 40 units system

38- and 40 generation units. For future works, utilization of GBEST method will be verified on multi-objective optimization problems. In addition, extraction of the unknown parameters of the solar cell models by GBEST algorithm is currently an active issue and will be discussed in an upcoming paper.

Compliance with ethical standards

Conflict of interest The authors declare that there is no potential conflict of interest in this work.

References

1. Salimi H (2015) Stochasic fractal search: a powerful metaheuristic algorithm. Knowl Based Syst 75:1–18. https://doi.org/10.1016/j.knosys.2014.07.025
2. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:45–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
3. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl Based Syst 96:120–133. https://doi.org/10.1016/j.knosys.2015.12.022
4. Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27:495–513. https://doi.org/10.1007/s00521-015-1870-7
5. Glover F (1989) Tabu search: part I. ORSA J Comput 1:190–206. https://doi.org/10.1287/ijoc.1.3.190
6. Davis L (1991) Bit-climbing, representational bias and test suite design. In: Proccedings of the fourth ınternational conference on genetic algorithm, San Francisco CA, pp18–23
7. Lourenço HR, Martin O, Stützle T (2010) Iterated local search, handbook of metaheuristics. In: Kluwer Academic Publishers, international series in operations research and management science, pp 363–397
8. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE ınternational conference on neural networks, pp 1942–1948
9. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179:2232–2248. https://doi.org/10.1016/j.ins.2009.03.004
10. Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76:60–68. https://doi.org/10.1177/003754970107600201
11. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell 1:28–39. https://doi.org/10.1109/MCI.2006.329691
12. Storn R, Price K (1997) Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359. https://doi.org/10.1023/A:1008202821328
13. Li G, Lin Q, Cui L, Du Z, Liang Z, Chen J, Lu N, Ming Z (2016) A novel hybrid differential evolution algorithm with modified CoDE and JADE. Appl Soft Comput 47:577–599
14. Mohamed AW, Mohamed AK, Elfeky EZ, Saleh M (2019) Enhanced directed differential evolution algorithm for solving constrained engineering optimization problems. Int J Appl Metaheuristic Comput (IJAMC) 10:1–28
15. Mohamed AK, Mohamed AW (2019) Real-parameter unconstrained optimization based on enhanced AGDE algorithm. In: Hassenien A (ed) Machine learning paradigms: theory and application, studies in computational intelligence, vol 801
16. Polakova R, Tvrdik Butok P (2019) Differential evolution with adaptive mechanism of population size according to current population diversity. Swarm Evol Comput 50:100519
17. Civicioglu P (2013) Artificial cooperative search algorithm for numerical optimization problems. Inf Sci 229:58–76. https://doi.org/10.1016/j.ins.2012.11.013
18. Oftadeh R, Mahjoob MJ, Shariatpanahi M (2010) A novel metaheuristic optimization algorithm inspired by group hunting of animals: hunting search. Comput Math Appl 60:2087–2098. https://doi.org/10.1016/j.camwa.2010.07.049
19. Shah-Hosseini H (2011) Principal components analysis by the galaxy based search algorithm: a novel metaheuristic for continuous optimization. Int J Comput Sci Eng 6:132–140. https://doi.org/10.1504/IJCSE.2011.041221
20. Kaveh A, Talathari S (2010) A novel heuristic method: charged system search. Acta Mech 213:267–289. https://doi.org/10.1007/s00707-009-0270-4
21. Erol OK, Eksin I (2006) A new optimization method: big bang – big crunch. Adv Eng Softw 179:106–111. https://doi.org/10.1016/j.advengsoft.2005.04.005
22. Kaveh A, Khayarazad M (2012) A new metaheuristic method: ray optimization. Comput Struct 112:283–294. https://doi.org/10.1016/j.compstruc.2012.09.003
23. Kaveh A, Mahdavi VR (2013) Colliding bodies optimization: a novel meta-heuristic method. Comput Struct 139:18–27. https://doi.org/10.1016/j.compstruc.2014.04.005
24. Quyang H, Gao L, Kong X, Li S, Zou D (2016) Hybrid harmony search particle swarm optimization with global dimension selection. Inf Sci 346:318–337. https://doi.org/10.1016/j.ins.2016.02.007
25. Li Z, Wang W, Yan Y, Li Z (2015) PS-ABC: a hybrid algorithm based on particle swarm artificial bee colony for high dimensional optimization problems. Expert Syst Appl 42:8881–8895. https://doi.org/10.1016/j.eswa.2015.07.043
26. Mandloi M, Bhatia V (2016) A low complexity hybrid algorithm based on particle swarm and ant colony optimization for large-MIMO detection. Expert Syst Appl 50:66–74. https://doi.org/10.1016/j.eswa.2015.12.008
27. Sun G, Zhang A, Yao Y, Wang Z (2016) A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding. Appl Soft Comput 46:703–730. https://doi.org/10.1016/j.asoc.2016.01.054
28. Wang GG, Gandomi AH, Alavi AH (2014) An effective krill herd algorithm with migration operator in biogeography-based optimization. Appl Math Model 38:2454–2462. https://doi.org/10.1016/j.apm.2013.10.052
29. Jayabarathi T, Raghunathan T, Adarsh BR, Suganthan PN (2016) Economic dispatch using hybrid grey wolf optimizer. Energy 111:630–641. https://doi.org/10.1016/j.energy.2016.05.105
30. Gao W, Liu S, Huang L (2012) A global best artificial bee colony algorithm for global optimization. J Comput Appl Math 236:2741–2753. https://doi.org/10.1016/j.cam.2012.01.013
31. Zorarpaci E, Özel SA (2016) A hybrid approach of differential evolution and artificial bee colony for feature selection. Expert Syst Appl 62:91–103. https://doi.org/10.1016/j.eswa.2016.06.004
32. Gao K, Zhang Y, Sadollah A, Rong S (2016) Optimizing urban traffic light scheduing problem using harmony search with ensemble local search. Appl Soft Comput 48:359–372. https://doi.org/10.1016/j.asoc.2016.07.029
33. Toksari MD (2016) A hybrid algorithm of ant colony optimization (ACO) and ıterated local search (ILS) for estimating electricity domestic consumption: case of Turkey. Int J Electr Power 78:776–782. https://doi.org/10.1016/j.ijepes.2015.12.032
34. Moradi P, Gholampour M (2016) A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. Appl Soft Comput 43:117–130. https://doi.org/10.1016/j.asoc.2016.01.044
35. Qin J, Xu X, Wu Q, Cheng TCE (2016) Hybridization of tabu search with feasible and infeasible local searches for

quadratic multiple knapsack problem. Comput Oper Res 66:199–214. https://doi.org/10.1016/j.cor.2015.08.002
36. Gandomi AH, Yang XS (2014) Chaotic bat algorithm. J Comput Sci 5:224–232. https://doi.org/10.1016/j.jocs.2013.10.002
37. Gao F, Gao H, Li Z, Tong H, Lee J (2009) Detecting unstable periodic orbits of nonlinear mappings by a novel quantum-behaved particle swarm optimization non-Lyapunov way. Chaos Solitons Fract 42:2450–2463. https://doi.org/10.1016/j.chaos.2009.03.119
38. Pan QK, Wang L, Gao L (2011) A chaotic harmony search algorithm for the flow shop scheduling problem with limited buffers. Appl Soft Comput 11:5270–5280. https://doi.org/10.1016/j.asoc.2011.05.033
39. Mukherjee A, Mukherjee V (2016) Chaotic krill herd algorithm for optimal reactive power dispatch considering FACTS devices. Appl Soft Comput 44:163–190. https://doi.org/10.1016/j.asoc.2016.03.008
40. Gokhale SS, Kale VS (2016) An application of a tent map initiated chaotic firefly algorthm for optimal overcurrent relay coordination. Int J Electr Power 78:336–342. https://doi.org/10.1016/j.ijepes.2015.11.087
41. Rao RV, Savsani VJ, Vakhana DP (2011) Teaching-learning based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43:303–315. https://doi.org/10.1016/j.cad.2010.12.015
42. Ahmadi SA (2016) Human behaviour-based optimization: a novel metaheuristic approach to solve complex optimization problems. Neural Comput Appl. https://doi.org/10.1007/s00521-016-2334-4
43. Kashan AH (2014) League championship algorithm (LCA): an algorithm for global optimization inspired by sport championships. Appl Soft Comput 16:171–200. https://doi.org/10.1016/j.asoc.2013.12.005
44. Savsani P, Savsani V (2016) Passing vehicle search (PVS): a novel metaheuristic algorithm. Appl Math Model 40:3951–3978. https://doi.org/10.1016/j.apm.2015.10.040
45. Ramezani F, Lotfi S (2013) Social based algorithm (SBA). Appl Soft Comput 13:2837–2856. https://doi.org/10.1016/j.asoc.2012.05.018
46. Eita MA, Fahmy MM (2014) Group counseling optimization. Appl Soft Comput 22:585–604. https://doi.org/10.1016/j.asoc.2014.03.043
47. Olorunda O, Engelbracht AP (2008) Measuring exploration/exploitation in particle swarms using swarm diversity. In: Proceedings of the 2008 IEEE congress on evolutionary computation, CEC (IEEE world congress on computational intelligence), pp 1128–1134
48. Alba E, Dorronsoro B (2005) The exploration/exploitation tradeoff in dynamic cellular genetic algorithm. IEEE Trans Evol Comput 9:126–142. https://doi.org/10.1109/TEVC.2005.843751
49. Civicioglu P (2012) Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm. Comput Geosci 46:229–247. https://doi.org/10.1016/j.cageo.2011.12.011
50. Turgut OE, Coban MT (2016) Thermal design of spiral heat exchangers and heat pipes through global best algorithm. Heat Mass Transf. https://doi.org/10.1007/s00231-016-1861-y
51. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical report, Computer Engineering Department, Engineering Faculty, Erciyes University
52. May R (1976) Simple mathematical models with very complicated dynamics. Nature 261:459–467. https://doi.org/10.1038/261459a0
53. Zhao SZ, Suganthan PN, Pan QK, Tasgetiren MF (2011) Dynamic multi-swarm particle swarm optimizer with harmony search. Expert Syst Appl 38:3735–3742. https://doi.org/10.1016/j.eswa.2010.09.032
54. Zhang J, Ding X (2011) A multi swarm self-adaptive and cooperative particle swarm optimization. Eng Appl Artif Intell 24:958–967. https://doi.org/10.1016/j.engappai.2011.05.010
55. Yu WJ, Zhang J (2011) Multipopulation differential evolution with adaptive parameter control for global optimization. In: Gecco-2011: proceedings of the 13 th annual genetic and evolutionary computation conference, pp1093–1098
56. Novoa-Hernandez P, Cruz Corona C, Pelta DA (2013) Self-adaptive, multipopulation differential evolution in dynamic environments. Soft Comput 17:1861–1881. https://doi.org/10.1007/s00500-013-1022-x
57. Liang J, Suganthan PN (2005) Dynamic multi-swarm particle swarm optimizer. In:Proceedings 2005 IEEE swarm ıntelligence symposium, IEEE, pp 124–129
58. Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evol Comput 13:398–417. https://doi.org/10.1109/TEVC.2008.927706
59. Rahnamayan S, Tizhoosh HR, Salama MMA (2008) Oppositon-based differential evolution. IEEE Trans Evol Comput 12:64–79. https://doi.org/10.1109/TEVC.2007.894200
60. Storn R (1996) On the usage of differential evolution for function optimization. In: Biennial conference of the north american fuzzy ınformation processing society (NAFIPS), IEEE, Berkeley, pp 519–523
61. Gamperle R, Müller SD, Koumoutsakos P (2002) A parameter study for differential evolution. In: Advances in ıntelligent systems, fuzzy systems, evolutionary computation, WSEAS Press, Interlaken, Switzerland, pp 293–298
62. Rönkkönen K, Kukkonen S, Price KV (2005) Real-parameter optimization with differential evolution, In: IEEE congresson evolutionary computation, pp 506–513
63. Rönkkönen J, Lampinen J (2003) On using mormally distributed mutation step length for differential evolution algorithm. In: Proceedings of MENDEL 2003, ninth ınternational MENDEL, conference on soft computing, Brno, Chech Republic, pp 11–18
64. Zhu W, Tang Y, Fang J, Zhang W (2013) Adaptive population tuning scheme for differential evolution. Inf Sci 223:164–191. https://doi.org/10.1016/j.ins.2012.09.019
65. Stanovov V, Akhmedova S, Semenkin E (2019) Selective pressure strategy in differential evolution: exploitation improvement in solving global optimization problems. Swarm Evol Comput 50:100463
66. Gong W, Fialho A, Cai Z, Li H (2011) Adaptive strategy selection in differential evolution for numerical optimization: an empirical study. Inf Sci 181:5364–5386. https://doi.org/10.1016/j.ins.2011.07.049
67. Jia G, Wang Y, Cai Z, Jin Y (2013) An improved (μ+λ): constraind differential evolution for constrained optimization. Inf Sci 222:302–322. https://doi.org/10.1016/j.ins.2012.01.017
68. Brest J, Greiner S, Boscovic B, Mernik M, Zumer V (2006) Self adaptive control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans Evol Comput 10:646–657. https://doi.org/10.1109/TEVC.2006.872133
69. Gandomi AH, Yang XS (2012) Evolutionary boundary constraint handling scheme. Neural Comput Appl 21:1449–1462. https://doi.org/10.1007/s00521-012-1069-0
70. Turgut OE, Coban MT (2016) Optimal proton exchange membrane fuel cell modelling based hybrid teaching learning based optimization–differential evolution algorithm. Ain Shams Eng J 7:347–360. https://doi.org/10.1016/j.asej.2015.05.003
71. Mirjalili S (2015) The ant lion optimizer. Adv Eng Soft 83:80–98. https://doi.org/10.1016/j.advengsoft.2015.01.010

72. Kashan AH (2015) A new metaheuristic for optimization: optics inspired optimization (OIO). Comput Oper Res 55:99–125. https://doi.org/10.1016/j.cor.2014.10.011
73. Mirjalili S (2015) Moth-flame optimziation algorithm: a novel nature-inspired heuristic paradigm. Knowl- Based Syst 89:228–249. https://doi.org/10.1016/j.knosys.2015.07.006
74. Gandomi AH, Yang XS, Alavi AH, Talatahari S (2013) Bat algorithm for constrained optimization tasks. Neural Comput Appl 22:1239–1255. https://doi.org/10.1007/s00521-012-1028-9
75. Sun J, Feng B, Xu WB (2004) Particle swarm optimization with particles having quantume behavior. IEEE Proceedings of congress on evolutionary computation, pp 325–331
76. Askerzadeh A (2014) Bird mating optimizer: an optimization algorithm inspired by bird mating strategies. Commun Nonlinear Sci 19:1213–1228. https://doi.org/10.1016/j.cnsns.2013.08.027
77. Yadav P, Kumar R, Panda SK, Chang CS (2012) An ıntelligent tuned harmony search algorithm for optimization. Inf Sci 196:47–72. https://doi.org/10.1016/j.ins.2011.12.035
78. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm: a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 110–111:151–166. https://doi.org/10.1016/j.compstruc.2012.07.010
79. Gandomi AH, Yang XS, Alavi AH (2011) Mixed variable structural optimization using firefly algorithm. Comput Struct 89:2325–2336. https://doi.org/10.1016/j.compstruc.2011.08.002
80. Dodu J, Martin P, Merlin A, Pouget J (1972) An optimal formulation and solution of short range operating problems for a power system with flow constraints. Proc IEEE 60:54–63. https://doi.org/10.1109/PROC.1972.8557
81. Chen CI, Wang S (1993) Branch-and-bound scheduling for thermal generating units. IEEE Trans Energy Convers 8:184–189. https://doi.org/10.1109/60.222703
82. Adarsh BR, Raghunathan T, Jayabarathi T, Yang XS (2016) Economic dispatch using chaotic bat algorithm. Energy 96:666–675. https://doi.org/10.1016/j.energy.2015.12.096
83. Biswal S, Barisal AK, Behere A, Prakash T (2013) Optimal power dispatch using BAT algorithm. In: International conference on energy efficient technologies for sustainability (ICEETS), Nagercoil, pp 1018–1023
84. Yang XS, Hosseini SSS, Gandomi AH (2012) Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. Appl Soft Comput 12:1180–1186. https://doi.org/10.1016/j.asoc.2011.09.017
85. Sulaiman MH, Zakaria ZN, Mohd-Rashid MI, Rahim SRA (2013) A new swarm intelligence technique for solving economic dispatch problem. In: Power engineering and optimization IEEE conference, pp 199–202
86. Yu JJQ, Li VOK (2016) A social spider algorithm for solving the non-convex economic load dispatch problem. Neurocomputing 171:955–965. https://doi.org/10.1016/j.neucom.2015.07.037
87. Vlachogiannis JG, Lee KY (2009) Economic load dispatch: a comparative study on heuristic optimization techniques with an improved coordinated aggregation-based PSO. IEEE Trans Power Syst 24:991–1001. https://doi.org/10.1109/TPWRS.2009.2016524
88. Chen PH, Chang HC (1995) Large-scale economic dispatch by genetic algorithm. IEEE Trans Power Syst 10:1919–1926. https://doi.org/10.1109/59.476058
89. Sinha N, Chakabarti R, Chattopadhyay PK (2003) Evolutionary programming techniques for economic load dispatch. Evol Comput IEEE Trans 7:83–94. https://doi.org/10.1109/TEVC.2002.806788
90. Chaturvedi KT, Pandit M, Srivastava L (2009) Particle swarm optimization with time varying acceleration coefficients for non-convex economic power dispatch. Int J Electr Power 31:249–257. https://doi.org/10.1016/j.ijepes.2009.01.010
91. Bhattacharya A, Chattopadhyay PK (2010) Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch. IEEE Trans Power Syst 25:1955–1964. https://doi.org/10.1016/j.eswa.2011.04.208
92. Coelho LdS, Souza RCT, Mariani VC (2009) Improved differential evolution approach based on cultural algorithm and diversity measure applied to solve economic load dispatch problems. Math Comput Simul 79:3136–3147. https://doi.org/10.1016/j.matcom.2009.03.005
93. Chatuverdi KT, Pandit M, Srivastava L (2008) Self-organizing hierarchical particle swarm optimization for non-convex economic dispatch. IEEE Trans Power Syst 23:1079–1087. https://doi.org/10.1109/TPWRS.2008.926455
94. Subbaraj P, Rengaraj R, Salivahanan S (2011) Enhancement of self-adaptive real-coded genetic algorithm using Taguchi method for economic dispatch problem. Appl Soft Comput 11:83–92. https://doi.org/10.1016/j.asoc.2009.10.019
95. Basu M (2014) Improved differential evolution for economic dispatch. Electr Power Energy Syst 63:855–861. https://doi.org/10.1016/j.ijepes.2014.07.003
96. Aragon VS, Esquivel SC, Coello Coello CA (2015) An immune algorithm with power redistribution for solving economic dispatch problems. Inf Sci 295:609–632. https://doi.org/10.1016/j.ins.2014.10.026
97. Bhattacharya A, Chattopadhyay PK (2010) Biogeography-based optimization for different economic load dispatch problems. IEEE Trans Power Syst 25:273–283. https://doi.org/10.1109/TPWRS.2009.2034525
98. Amjady N, Nasiri-Rad H (2009) Economic dispatch using an efficient real coded genetic algorithm. IET Gener Transm Distrib 3:266–278. https://doi.org/10.1049/iet-gtd:20080469
99. Coelho LdS, Mariani VC (2010) An efficient cultural self organizing migrating strategy for economic dispatch optimization with valve-point effect. Energy Convers Manag 51:2580–2587. https://doi.org/10.1016/j.enconman.2010.05.022
100. Sayah S, Hamouda A (2013) A hybrid differential evolution algorithm based on particle swarm optimization for non-convex economic dispatch problems. Appl Soft Comput 13:1608–1619. https://doi.org/10.1016/j.asoc.2012.12.014