An International Journal of Optimization and Control: Theories & Applications
ISSN: 2146-0957 eISSN: 2146-5703
Vol.11, No.2, pp.158-177 (2021)
https://doi.org/10.11121/ijocta.01.2021.001077

RESEARCH ARTICLE

Differential gradient evolution plus algorithm for constraint optimization problems: A hybrid approach

Muhammad Farhan Tabassum[a], Sana Akram[a], Saadia Hassan[b], Rabia Karim[b], Parvaiz Ahmad Naik[c], Muhammad Farman[d], Mehmet Yavuz[e,*], Mehraj-ud-din Naik[f], Hijaz Ahmad[g,h]

[a] *Department of Mathematics, University of Management and Technology, Lahore, 54000, Pakistan.*
[b] *Department of Sports Sciences, Faculty of Allied Health Science, University of Lahore, Lahore, 54000, Pakistan.*
[c] *School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, Shaanix. People's Rebublic of China.*
[d] *Department of Mathematics and Statistics, University of Lahore, Lahore, 54000, Pakistan*
[e] *Department of Mathematics and Computer Sciences, Faculty of Science, Necmettin Erbakan University, 42090 Konya, Turkey*
[f] *Department of Chemical Engineering, College of Engineering, Jazan University, Jazan 45142, Saudi Arabia*
[g] *Section of Mathematics, International Telematic University Uninettuno, Corso Vittorio Emanuele II, 39, 00186 Roma, Italy*
[h] *Department of Basic Sciences, University of Engineering and Technology, Peshawar, Pakistan*

ARTICLE INFO

Article history:
Received: 19 January 2021
Accepted: 25 March 2021
Available Online: 2 May 2021

Keywords:
Meta-heuristic algorithms
Hybridization
Differential evolution
Gradient evolution
Constraint optimization problems

AMS Classification 2010:
90C15, 90C26, 90C30, 90C59, 90C90

ABSTRACT

Optimization for all disciplines is very important and applicable. Optimization has played a key role in practical engineering problems. A novel hybrid meta-heuristic optimization algorithm that is based on Differential Evolution (DE), Gradient Evolution (GE) and Jumping Technique named Differential Gradient Evolution Plus (DGE+) are presented in this paper. The proposed algorithm hybridizes the above-mentioned algorithms with the help of an improvised dynamic probability distribution, additionally provides a new shake off method to avoid premature convergence towards local minima. To evaluate the efficiency, robustness, and reliability of DGE+ it has been applied on seven benchmark constraint problems, the results of comparison revealed that the proposed algorithm can provide very compact, competitive and promising performance.

1. Introduction

Optimization is the best fit solution for all possible solutions to a given problem. Many modern optimization approaches fail to solve complex problems. Several researchers then started proposing new approaches to solve complex optimization problems in reasonable time and cost. There are two groups for optimizing methods: deterministic algorithms and stochastic algorithms [2]. If the same initial values are used, Deterministic methods may obtain the same results. Such algorithms have good efficacy for certain problems, but for all forms of optimization problems, it is difficult to generalize them [3]. One disadvantage of these search algorithms, they can simply be trapped in the local optimum [4]. For their strategies, stochastic algorithms usually use some randomness and avoid striking at a local optimum. Although they can have high-quality solutions in a reasonable amount of time for hard optimization problems, they do not ensure that the best solution will be found always.

The complexity of real-world problems has risen over the last few decades. To resolve these problems, a new meta-heuristic technique needs to be developed that is used to achieve optimal solutions with a low computational cost. Meta-heuristics are broadly divided into three categories: algorithms based on evolution theory, physical phenomena and swarm intelligence. A population-based meta-heuristic, inspired by the biological evolution based on mutation, reproduction, selection, and recombination.

*Corresponding author

Algorithms derived from physical phenomena are the second category. In these algorithms, search agents will move around the search space according to the rules of gravity, inertia, and electromagnetism. The final class is swarming intelligence algorithms that are based on social creatures' collective behavior. There are also other meta-heuristic approaches influenced by human behavior. Modern meta-heuristic algorithms having two main components, exploration and exploitation [5, 6]. Exploration makes sure the algorithm hits various promising search space regions while exploitation concentrating on the local area's search [7]. To achieve optimal solutions, both components must be optimized. Schematic view of the classification of the meta-heuristic algorithms is as follows:

Evolutionary Algorithms: Biogeography Based Optimizer [8], Differential Evolution [9], Evolution Strategy [10], Genetic Algorithms [11], Genetic Programming [12].

Physics-Based Algorithms: Artificial Chemical Reaction Optimization Algorithm [13], Big-Bang Big Crunch [14], Gravitational Search Algorithm [15], Ray Optimization Algorithm [16], Simulated Annealing [17], Small-World Optimization Algorithm [18], Nonlinear Optimization Algorithm [19,20], Constrained Optimization Problem [21], Fractional Gradient Based Algorithm [22], Optimization Problems Based on Hyperbolic Penalty Dynamic Framework [23], Jaya Optimization Algorithm [24,25], Feedback Controller Algorithm [26].

Swarm-Based Algorithms: Ant Colony Optimization [27], Bat-Inspired Algorithm [28], Bee Collecting Pollen Algorithm [29], Cuckoo Search [30], Particle Swarm Optimization [31].

Human Behaviors-Based Algorithms: Colliding Bodies Optimization [32], Mine Blast Algorithm [33], Seeker Optimization Algorithm [34], Soccer League Competition Algorithm [35], Social-Based Algorithm [36].

Differential Evolution (*DE*) is one of Price and Storn's most suitable and commonly used evolutionary algorithms [9]. Several methodologies were suggested and used to solve the various optimization problems in literature with the classic *DE* algorithm, such as Adaptive Chaotic *DE* [37], Adaptive Hybrid *DE* [38], *DE* with Ant Colony Optimization [39], *DE* with Firefly Algorithm [40], Modified Teaching–Learning Algorithm [41], Hybrid differential evolution with biogeography-based optimization [42].

The system for gradient evolution uses a series of vectors and consists of three main steps: updating, jumping and refreshing the search space. The major rule for gradient evolution is vector updating. Using the Newton–Raphson method search direction has been determined. The jumping and refreshing vector system allows local optima to be avoided [43]. This concept is based on gradient-based methods of search, such as the newton method, the conjugate direction and the Quasi-Newton method [44].

This paper introduces a new metaheuristic algorithm to optimize unconstrained and chemical design problems. The main characteristic of this paper are as follows: 1) A novel hybrid meta-heuristic optimization algorithm based on local and global search. This algorithm is the best combination of exploration and exploitation. 2) The proposed hybridized algorithm works with the help of an improvised dynamic probability distribution. 3) Additionally, it provides a novel shake off method to avoid premature convergence towards local minima. 4) It has been applied on several benchmark unconstraint problems and four complex practical engineering problems to evaluate the efficiency of proposed algorithm. The remaining of this paper is organized as follows: in section 2, the comprehensive detail of Differential Evolution and Gradient Evolution. In section 3, the proposed DGE+ and the concepts behind it are introduced in details. In section 4, the performance of the proposed optimizer is validated on different constrained optimization problems. Finally, conclusions and future directions are given in section 5.

2. Conventional algorithms

2.1. Differential evolution algorithm

Differential evolution is a relatively efficient meta-heuristic technique designed to optimize existing problems. Through applying mutation, crossover and selection operators, the population is successively improved over generations to achieve an optimal solution [45, 46]. The comprehensive detail of *DE* is present in [9, 47] and the main steps of the *DE* algorithm are given below in the form of a self-explanatory flow diagram shown in Figure 1.

2.2. Gradient evolution algorithm

Gradient evolution (*GE*) is an optimization algorithm based on the concept of gradients. The vector updating operator was driven from the Tylor series expansion and transforms the updating law for population-based search. The vector jumping operator prevents local optima and the vector refreshing operator is implemented in multiple iterations when a vector cannot move to a different location. The detail of this idea and the mathematical formulation of the *GE* algorithm is in [43, 48] the main steps of the *GE* algorithm are given below in the form of the self-explanatory flow diagram shown in Figure 2.

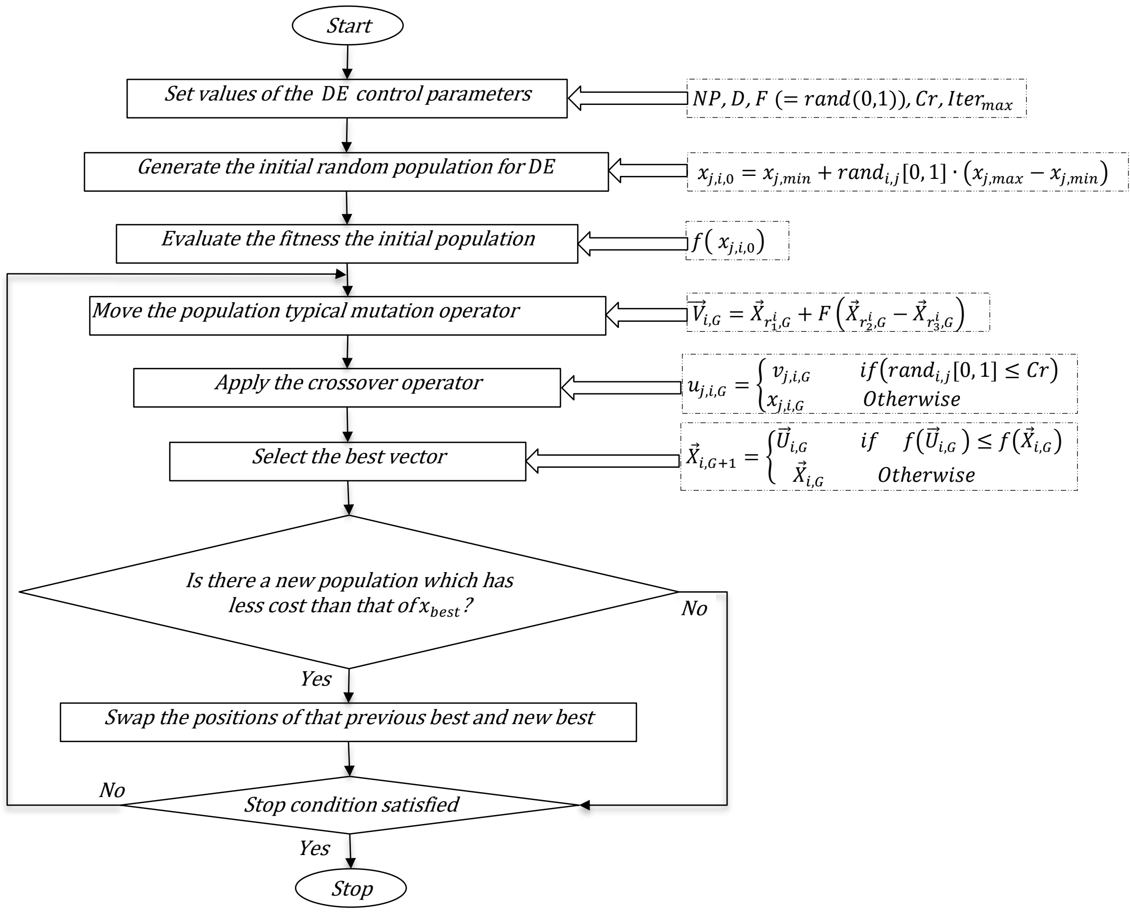

Figure 1. Flowchart for differential evolution

3. Differential gradient evolution plus

Differential Evolution is a powerful search technique to solve optimization problems with non-discreet variables. Differential Evolution is known for its excellent coverage of global search space and its tendency to find optimum solutions in higher dimensional optimization problems. On the other hand Gradient Evolution (GE) is a well-known technique that converges towards local minima by the use of instantaneous gradient information. In this way, GE is an effective method to explore local search space. The proposed algorithm hybridizes the above-mentioned algorithms with the help of an improvised dynamic probability distribution. The proposed algorithm additionally provides a new shake off' method to avoid premature convergence towards local minima. In this proposed method, the best solution of the last generation is maintained as a solution vector Y, this vector Y is used in the differential algorithm to generate new solutions. The proposed algorithm constantly monitors the best solution produced in each completed generation and if no significant improvement against best solution Y of previous generations is observed over a specified number of generations then a shake-off sequence is initiated which slightly changes the position of Y in solution space. In this way, the search direction of all individual members of the population is changed which results in an increased probability of escaping local minima and finding the optimum solution. During the search, best solution found in any iteration is preserved and reported after the search. Combination of these three above mentioned techniques resulted in a novel algorithm, named DGE+ ($DE = Differential\ Evolution, GE = Gradient\ Evolution$ and $+ = Jumping\ Technique$), to solve unconstrained and constrained problems of any size and complexity. Each solution is represented with the symbol ${}^{t}X_i$, where $t = 1, 2, 3, \ldots, G_N$ and $i = 1, 2, 3, \ldots, P_s$ denotes generation and iteration respectively. Here G_N and P_s are user parameters which specify the total number of generation to be run and population size respectively.

$${}^{t}X_i = x_m, \text{ where } m = 1, 2, 3, \ldots, D. \tag{1}$$

In the above equation, x represents values of variables and D is the dimensions of search space and it is equal to the numbers of independent variables of the problem to be solved. The proposed algorithm starts with the initialization of the population with random values of independent variables. Each solution vector is initialized randomly by using the following formula;

$$X = \{LB + random(0 \cdots 1) \times (UB - LB)\}, \quad (2)$$

where LB and UB are lower and upper bounds of the particular variable in specified problem and $random$ number is generated between 0 and 1. This formula ensures uniform distribution of initial values of variables within upper and lower bounds which results in no need for any repair strategy.

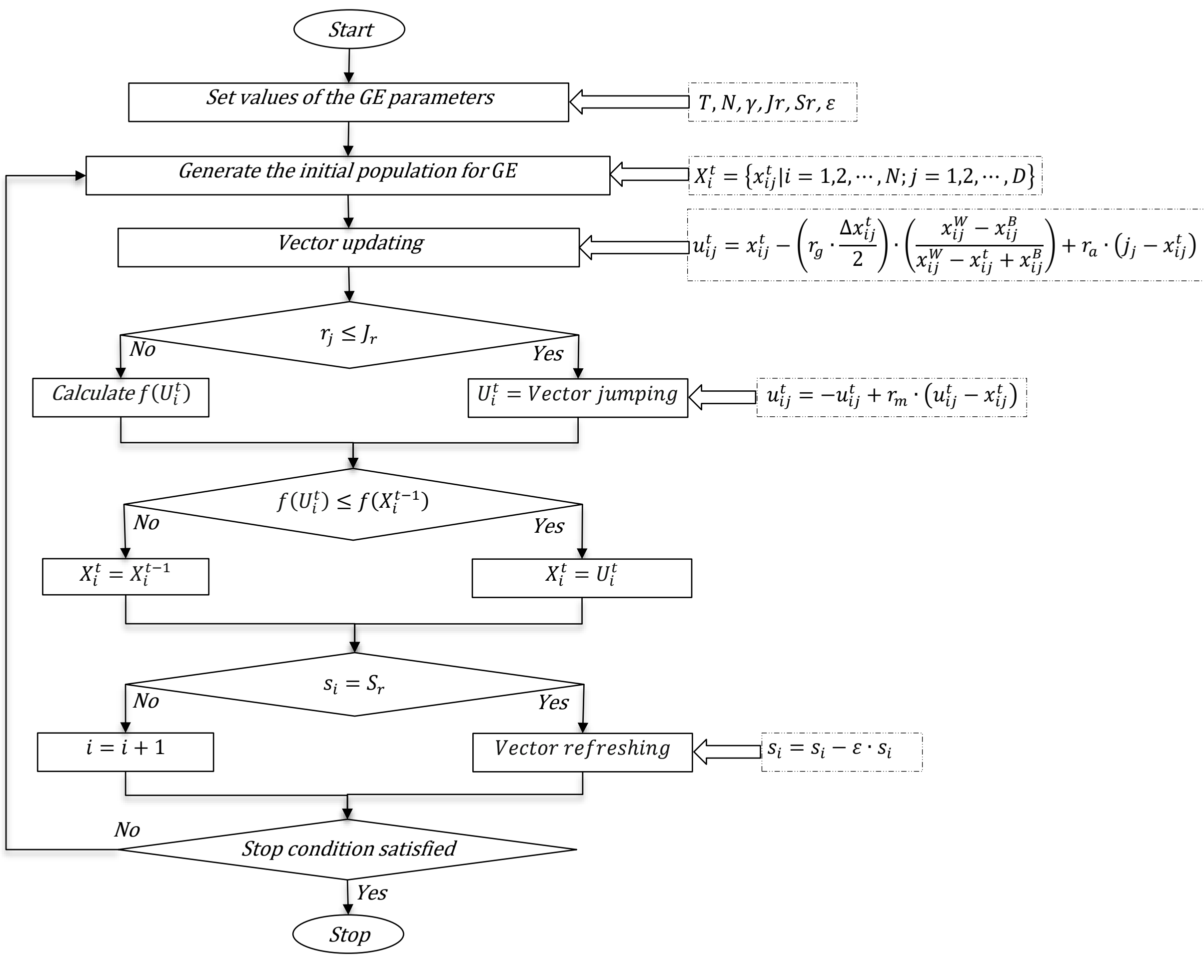

Figure 2. Flowchart for gradient evolution

After initialization, the complete population is evaluated for objective and constraints functions. At this stage, a solution vector Y is selected which is currently the best solution of this initial population. This initial population is then fed to the main body of the search loops. The new solutions are built using DE or GE, the selection of the algorithm to be used is dependent upon the following formula given in Eq. (3). In the following equation ${}^tU_i$ the new solution generated by the application of DE or GE at i^{th} iteration of t^{th} generation.

$${}^tU_i = \begin{cases} DE({}^tP), & if\ random\ (0 \cdots 1) > \frac{S_P}{G_N} \times t \\ GE({}^tP), & else \end{cases} \quad (3)$$

Algorithm selection probability of user parameter is represented by S_P. If differential evolution is to be used for the generation of new solutions then the following formula is used:

$${}^tU_i = DE({}^tP) = Y + S_F(X_{r_1} - X_{r_2}) + S_F(X_{r_3} - X_{r_4}), \quad (4)$$

where S_F is scaling factor and r_1, r_2, r_3 & r_4 are random integer numbers and their values range between 1 to P_s, such that $r_1 \neq r_2 \neq r_3 \neq r_4$. In case when a new solution is to be generated by the use of gradient evolution following formula is used:

$$\delta_x = \frac{\gamma + \left| {}^tX_i - {}^tX_{i-1} \right|}{2} \tag{5}$$

$$b = {}^tX_i - \delta_x \tag{6}$$

$$w = {}^tX_i + \delta_x \tag{7}$$

$$ {}^tU_i = \begin{cases} {}^tX_i - \frac{(rand \times \delta_x)}{2}\left(\frac{{}^tX_{i+1} - b}{{}^tX_{i+1} - {}^tX_i + b}\right), & if\ i = 1 \\ {}^tX_i - \frac{(rand \times \delta_x)}{2}\left(\frac{w - {}^tX_{i-1}}{w - {}^tX_i + {}^tX_{i-1}}\right), & if\ i = P_S \\ {}^tX_i - \frac{(rand \times \delta_x)}{2}\left(\frac{{}^tX_{i+1} - {}^tX_{i-1}}{{}^tX_{i+1} - {}^tX_i + {}^tX_{i-1}}\right), & otherwise \end{cases} \tag{8}$$

In the above expressions, gamma γ is a gradient evolution user parameter. The newly generated solution ${}^tU_i$ is compared with the available solution at i^{th} location of the current population, if this solution is found better then this solution is inserted into the population at i^{th} location. Additionally, this algorithm allows acceptance of solutions with poorer performance into the main population to maintain diversity. This insertion probability of poorer solution is depended on a user control parameter A_R. A random number is generated between 0 and 1 if this number is less than A_R then the poorer solution is accepted in the main population.

To maintain diversity in population, fresh vectors are regularly inserted into the main population. The rate of insertion of a new random vector in population is dependent upon a parameter R_R. After scanning all the members of the population, existing solution vector Y is compared with the best solution of the current population, if this new best solution is better than Y then this new solution is selected as Y and a variable which track changes in Y is reset to 0. For every failed attempt to update Y, this variable is incremented by 1 and if its count becomes equal to user control parameter S_T then the value of current Y is shaken off randomly as per following equations;

$$d = Min\left(\left| {}^tX_i - UB\right|, \left| {}^tX_i - LB\right|\right), \tag{9}$$

$$Y = Y + d \times rand(-1 \cdots +1) \times \frac{G_N - t}{G_N}. \tag{10}$$

The above-mentioned cycles are repeated continuously for all generations and in the end, the best solution, which is preserved during the whole search, is reported as the solution to the given optimization problem.

3.1. Parameter selection

A wrong selection of algorithm parameters may result in a higher tendency to diverge, pre-mature convergence to a local minimum value, or undesired solutions. Therefore, the following considerations should be taken into account to fine-tune the algorithm parameters.

3.1.1. Population size P_s

Optimization problems of low to medium complexity may require a population size of 30 to 50 individual solutions which are sufficient enough to solve the problem optimally. For the problem with a higher number of dimensions more individual members may be required to maintain diversity and room to explore global solution space. But on the other hand, larger population size results in higher computation time and increased number function evaluations. The benchmark problem set, selected for this study, of constrained and unconstrained problems contain optimization problems from low to high complexity. The experiments on the proposed algorithm show that $P_s = 50$ is sufficient enough to solve the entire problem set with excellent solution quality and in reasonable computational time.

3.1.2. Number of generations G_N

The number of generations required to solve a problem optimally is directly proportional to the number of independent variables of the optimization problem. A lower value of the G_N produces non-optimal solutions and an unrealistically high value of G_N results in unnecessary high computational cost. The experiments on the proposed algorithm show that for unconstraint problems with up to 10 variables $G_N = 6000$, up to 20 variables $G_N = 12000$ and up to 30 variables $G_N = 20000$ is sufficient to produce optimal results. For constrained problems $G_N = 600$ is sufficient to solve all the selected Problems with excellent optimal values of objective functions.

3.1.3. Gradient evolution parameter gamma γ

This parameter is used to control the performance of the gradient evolution part of the proposed algorithm. This number ensures that the value of change in any variable is non-zero; a zero value may lead to stagnation at the same point in solution space. The experiments on the proposed algorithm show that the complexity of the problem does not affect the value of this variable and for the chosen set of constrained and unconstrained problems $\gamma = 0.4$ has produced optimal results.

3.1.4. Differential evolution parameter scale factor S_F

This parameter acts as a control of acceleration of convergence and has the most prominent effect on the performance of the differential evolution algorithm. The value of this parameter is dependent on the complexity of objective and constraint functions, a lower value of S_F, may result in non-optimal solutions due to the slower rate of convergence and conversely a higher value of S_F may cause DE to jump over optimal solutions in search space. The experiments with the proposed algorithm suggest that for constrained problems $S_F = 0.5$ and for unconstrained problems $S_F = 0.48\ to\ 0.62$ has produced optimal results for all selected benchmark problems.

3.1.5. Differential evolution parameter crossover rate C_R

This parameter controls how much change, produced by DE should be passed on to the next generations. If the value of this parameter is set to a lower value then the convergence rate of the algorithm drops and vice versa. The value of this parameter should be set at a higher value to pass on the effect of DE to the next generations. The experiments on the proposed algorithm show that a value of $C_R = 0.91$ is good enough to produce optimal results for all selected benchmark constrained and unconstrained optimization problems.

3.1.6. Selection probability S_P

The proposed algorithm uses a differential evolution algorithm to explore (global search) and gradient evolution to exploit (local search) the given search space of the optimization problem. The decision when to use DE or GE is made by a dynamic probability function. At the start of the search, the probability of usage of DE is maximum and as the generations go the probability of DE usage drops and the probability of GE usage increases. In other words, in the beginning, more resources are utilized to perform a global search and in the end, relatively more computation is performed for local search. This dynamic probability distribution is controlled by the parameter S_P. A un-optimized low value of S_P usually causes less exploitation of local search space which results in poorer solution quality and a un-optimized higher value of S_P causes less exploration of global search space which in turn results in premature convergence to local minima. As both of these scenarios are undesirable therefore the value of this variable should be chosen carefully. The experiments conducted on all the constrained and unconstrained problems shows that $S_P = 0.2$ is good value to solve the entire set of benchmark problems optimally. This value $S_P = 0.2$ results in usage probability GE to increase from 0 to 20%, and consequentially the usage of DE drops from 100% to 80% during execution.

3.1.7. Sub-optimal solution acceptance rate A_R

All the new solutions which are produced either by DE or GE are tested for fitness against the corresponding member of the current population. If this new solution is better than the existing solution in the current population then this member of the population is killed and replaced by the newly generated solution. The proposed algorithm additionally allows for the acceptance of poorer solutions with a probability of A_R. This additional feature of the proposed algorithm maintains diversity in future populations and increases the probability of escaping local minima. The value of this parameter should be chosen carefully, in the case when the value of this parameter is set too high then the quality of search degrade and algorithm does not converge to the optimal values. The experiments on the proposed algorithm suggest that A_R between 0.01 and 0.05 is a good value to produce statistically better results in comparison to $A_R = 0$ for the given set of constrained and unconstrained problems.

3.1.8. Refresh rate R_R

For all population base algorithms regular supply of new individual solutions is essential to preserve diversity which in turn results in better solution quality. This fresh supply of new random solutions is controlled by R_R. A lower value of this parameter R_R causes the loss of diversity and poorer solution quality and a higher value of this parameter results in loss of better solutions and divergence of the optimization algorithm. The experiments with the proposed algorithm demonstrate that $R_R = 0.02$ is a decent value to solve the entire benchmark set of constrained and unconstrained problems.

3.1.9. Shake off threshold S_T

As an attempt to escape from local minima this proposed algorithm provides a shake off technique. The algorithm keeps monitoring the best solution of every subsequent generation and if no new improvement is observed then a counter is incremented by one. If the value of this variable becomes equal to shake off threshold S_T then shake off is initiated. A un-optimized high value of this threshold S_T will make this shake off ineffective and in contrast a low value of this parameter will result in poorer solution quality. The experiments conducted on our proposed algorithm indicates that the value of $S_T = 500$ and $S_T = 60$ for all unconstrained and constrained problems respectively can provide optimal results.

3.2. Constraint handling

Constraint handling of the problem is done as per rules given by Mottos & Coello [49]. The following four rules are used:

3.2.1. Rule 1

Whatever the value of the objective function is any feasible solution will always be preferred over infeasible solutions.

3.2.2. Rule 2

Infeasible solutions having a slight violation of 0.001 are considered as feasible solutions.

3.2.3. Rule 3

If two solutions are feasible then the one with better objective function value will be preferred.

3.2.4. Rule 4

If two solutions are infeasible then the one with less violation of feasibility will be preferred.

By incorporating first and fourth rules, the search is guided towards feasible regions rather than wasting resources by exploring infeasible regions of search space, the third rule forces the algorithm to both keep the search within the feasible regions and attempt to find a solution with a better value of objective function [49]. If the optimal solution lies near the boundary of the feasible region then the second rule facilitates the search of boundaries of the feasible region [50]. The algorithm of *DGE+* is as follows:

Algorithm: Differential Gradient Evolution Plus
Step 1: Initialize population
Step 2: Calculate objective and constraint functions
Step 3: Select Y which is the best solution in the current population
Step 4: Check the current generation is equal to G_N if yes then go to step 11. Otherwise, go to step 5
Step 5: Check current iteration is equal to P_s if yes then go to step 9. Otherwise, go to step 6
Step 6: Calculate U by using equations 3-8
Step 7: Evaluate U if it is acceptable then replace current solution of the population with this new solution U
Step 8: Go to step 5
Step 9: Check for shake off conditions, if true then change Y as per equations 9 and10
Step 10: Go to step 4
Step 11: Report the best solution and stop

The detail of the idea and the mathematical formulation of the *DGE+* algorithm is in the last section, the main steps of the *DGE+* algorithm are given below in the form of the self-explanatory flow diagram shown in Figure 3.

4. Experiments on constrained optimization problems

The comparison of the results produced by each constraint problem has been reported and listed in Table 1 which provided the comparative methods with references.

4.1. Experimental setup

The performance of the proposed novel and dynamic algorithm (*DGE+*) is exhibited by solving several optimization problems that are widely used to test optimization methods and considered as the benchmark problems in the literature. These test cases consist of seven benchmark constraint test problems [33]. All analyses are implemented in Matlab® environment on the computer equipped with the Intel CORE i5 @ 1.8 GHz CPU and 4 GB of RAM. The parameter settings of the proposed algorithm are:

Number of runs are 30, population size is 50, generations are 600, gamma value is 0.4, scale factor is 0.5 and cross over is 0.91. In the following subsections, *DGE+* is implemented on seven benchmark constraint problems and eight complex practical engineering problems.

4.2. Constrained optimization problems

4.2.1. Constrained problem 1

Braken and McCormick [84] originally introduced this problem which is a relatively simple constrained problem of minimization, having two variables and two constraints, one is equality constraint and the other is inequality constraint.

$$\min\ f(x) = (x_1-2)^2 + (x_2-1)^2$$

$$subject\ to \begin{cases} h_1(x) = x_1 - 2x_2 + 1 = 0 \\ h_2(x) = -\left(\dfrac{{x_1}^2}{4}\right) - {x_2}^2 + 1 \geq 0 \end{cases}$$

$$-10 \leq x_{1,}x_2 \leq 10$$

Table 2 demonstrates the comparison of the best solution among the different optimizers and the corresponding design variables. The results obtained by *DGE+* are compared with 4 state-of-the-art algorithms that are abbreviated and listed in Table 1.

Evolutionary programming violets both the constraints and remaining methods violet first constraint for the final solution but *DGE+* satisfies all constraints for the final solution. It is evident from Table 2 that the proposed *DGE+* algorithm performed better and superior to all the state-of-the-art methods without any violation.

The convergence curve shows the function values versus the number of generations for the constrained problem 1. The 30 trials of the best solution obtained from the *DGE+* algorithm are given in Figure 4.

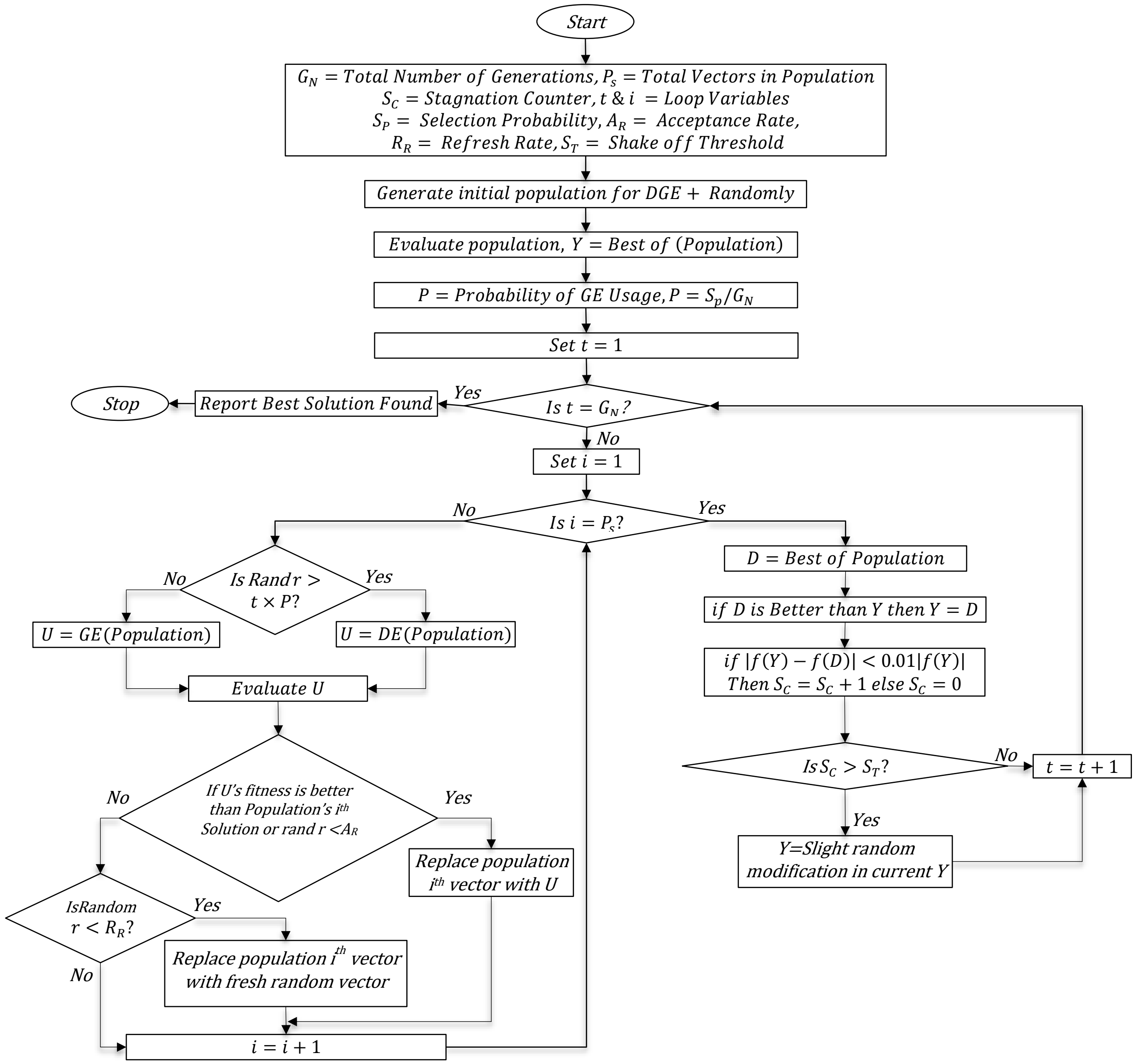

Figure 3. Flowchart for differential gradient evolution plus

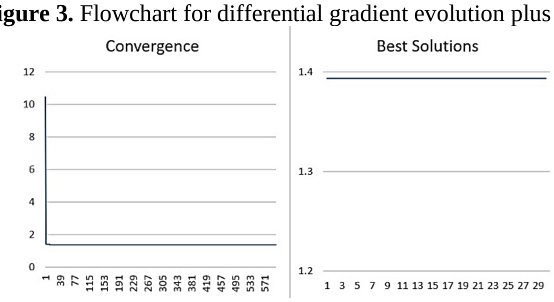

Figure 4. Convergence curve and 30 best solutions for constraint problem 1

Table 1. Comparative algorithms with references

Key	Algorithm Name	Key	Algorithm Name
MBA [33]	Mine Blast Algorithm	HM [51]	Homomorphous Mappings
ISR [52]	Improved Stochastic Ranking	HPSO [53]	Hybrid Particle Swarm Optimization
ABC [54, 55]	Artificial Bee Colony	HS [56, 57]	Harmony Search
IGA [58]	Interactive Genetic Algorithm	CRGA [59]	Changing Range Genetic Algorithm
ASCHEA [60]	Adaptive Segregational Constraint Handling Evolutionary Algorithm	CPSO-GD [61]	Co-evolutionary Particle Swarm Optimization Using Gaussian Distribution
CAEP [62]	Cultural Algorithm using Evolutionary Programming	Co-DE [63]	Effective Co-Evolutionary Differential Evolution
NM-PSO [64]	Nelder-Mead Particle Swarm Optimization	PSO [53]	Particle Swarm Optimization
CULDE [65]	Cultured Differential Evolution	PESO [66]	Particle Evolutionary Swarm Optimization
SAPF [67]	Self-Adaptive Penalty Function	EP [67]	Evolutionary Programming
DE [68]	Differential Evolution	GA [69-71]	Genetic Algorithms
DEDS [73]	Differential Evolution with Dynamic Stochastic	DELC [74]	Differential Evolution with Level Comparison
FSA [75]	Filter Simulated Annealing	SR [52]	Stochastic Ranking
GA with TS, PS [75]	Efficient Constraint Handling Method For Genetic Algorithms	α-Simples [77]	A Constrained Method
GA1 [76]	Genetic Algorithms 1	SMES [78]	Simple Multi-membered Evolution Strategy
GA2 [79]	Genetic Algorithms 2	TLBO [80]	Teaching-Learning-Based Optimization
HEAA [81]	Hybrid Evolutionary Algorithm and Adaptive technique	PSO-DE [82] [83]	Particle Swarm Optimization with Differential Evolution

Table 2. Reported results for constrained problem 1 from different optimizers

Methods	***Design variables***		$\boldsymbol{f(x)}$	***Constraints***	
	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$		$\boldsymbol{h_1(x)}$	$\boldsymbol{h_2(x)}$
HS	0.8343	0.9121	1.3770	$5E-03$	$5.4E-03$
GA	0.8080	0.8854	1.4339	$3.7E-02$	$5.2E-02$
MBA	0.822875	0.911437	1.3934649	$1.11E-06$	0
EP	0.8350	0.9125	1.3772	$1.0E-02$	$-7.0E-02$
$DGE+$	$\mathbf{0.822875656}$	$\mathbf{0.911437828}$	$\mathbf{1.393464981}$	$\mathbf{0}$	$\mathbf{0}$

4.2.2. Constrained problem 2

This problem is taken from [33] which is a relatively simple constrained problem of minimization having two variables and one equality constraint.

$\min\ f(x) = {x_1}^2 + (x_2 - 1)^2$

$subject\ to\ \{h(x) = x_2 - {x_1}^2 = 0,$

$-1 \le x_1, x_2 \le 1.$

Table 3 demonstrates the comparison of the best solution among the different optimizers and the corresponding design variables. *CULDE*, *SAPF*, *PSO-DE*, and *MBA* violates the constraint but *DGE+* satisfies constraint for the final solution. The results obtained by $DGE+$ are also compared with 10 state-of-the-art algorithms that are abbreviated and listed in Table 1. The comparison of statistical results for constrained problem 2 is given in Table 4. It is evident from Tables 3 and 4 that the proposed *DGE+* algorithm performed better and superior to all the state-of-the-art methods without any violation.

Table 3. Reported results for constrained problem 2 from different optimizers

Methods	***Design variables***		$\boldsymbol{f(x)}$	***Constraint***
	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$		$\boldsymbol{h(x)}$
$PSO-DE$	-0.7069	0.49975	0.749957673	$4.2E-05$
$CULDE$	-0.707036	0.5	0.749899905	0.0001
$SAPF$	-0.706	0.4996	0.74883616	0.00116
MBA	-0.706958	0.49979	0.749999658	$3.9E-07$
$\boldsymbol{DGE+}$	$\mathbf{-0.707106782}$	$\mathbf{0.5}$	$\mathbf{0.75}$	$\mathbf{0}$

Table 4. Statistical comparison of results for constrained problem 2 of various algorithms

$\boldsymbol{Method}$	$\boldsymbol{Worst}$	$\boldsymbol{Mean}$	$\boldsymbol{Best}$	$\boldsymbol{SD}$
HM	0.75	0.75	0.75	$N.A$
$ASCHEA$	$N.A$	0.75	0.75	$N.A$
$CRGA$	0.757	0.752	0.750	$2.5E-03$
$SMES$	0.75	0.75	0.75	$1.52E-04$
PSO	0.998823	0.860530	0.750000	$8.4E-02$
SR	0.750	0.750	0.750	$8E-05$
$DELC$	0.750	0.750	0.750	0
$HEAA$	0.750	0.750	0.750	$3.4E-16$
ISR	0.750	0.750	0.750	$1.1E-16$
ABC	0.75	0.75	0.75	0
$\boldsymbol{DGE+}$	$\boldsymbol{0.75}$	$\boldsymbol{0.75}$	$\boldsymbol{0.75}$	$\boldsymbol{0}$

"N.A" means not available.

The convergence curve shows the function values versus the number of generations for the constrained problem 2. The 30 trials of the best solution obtained from the *DGE+* algorithm are given in Figure 5.

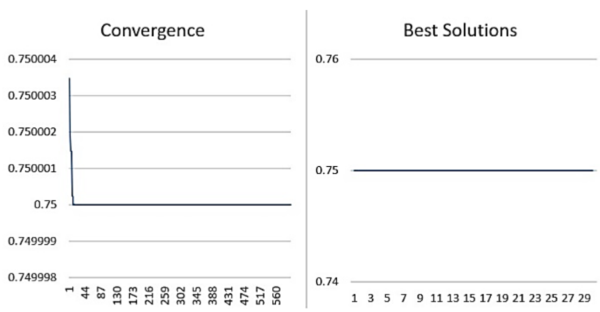

Figure 5. Convergence curve and 30 best solutions for constraint problem 2

4.2.3. Constrained problem 3

This problem is taken from [33] which is a relatively simple constrained problem of minimization having two variables and two inequality constraints.

$$min\ f(x) = ({x_1}^2 + x_2 - 11)^2 + (x_1 + {x_2}^2 - 7)^2$$

$$subject\ to \begin{cases} h_1(x) = 4.84 - (x_1 - 0.05)^2 - (x_2 - 2.5)^2 \geq 0 \\ h_2(x) = {x_1}^2 + (x_2 - 2.5)^2 - 4.84 \geq 0 \end{cases}$$

$$0 \leq x_1, x_2 \leq 6.$$

Table 5 demonstrates the comparison of the best solution among the different optimizers and the corresponding design variables. The results obtained by *DGE+* are compared with 5 state-of-the-art algorithms that are abbreviated and listed in Table 1. Harmony search violets both the constraints and mine blast algorithm violet second constraint for the final solution but *DGE+* satisfies all constraints for the final solution.

Table 5. Reported results for constrained problem 3 from different optimizers

$\boldsymbol{Methods}$	$\boldsymbol{Design\ variables}$		$\boldsymbol{f(x)}$	$\boldsymbol{Constraints}$	
	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$		$\boldsymbol{h_1(x)}$	$\boldsymbol{h_2(x)}$
$GA\ with\ PS\ (R = 0.01)$	$N.A$	$N.A$	13.58958	$N.A$	$N.A$
$GA\ with\ PS\ (R = 1)$	$N.A$	$N.A$	13.59108	$N.A$	$N.A$
$GA\ with\ TS$	2.246826	2.381865	13.59085	$N.A$	$N.A$
HS	2.24684	2.382136	13.590845	$-2.09E-06$	-0.222181
MBA	2.246833	2.381997	13.590842	0	-0.222183
$\boldsymbol{DGE+}$	$\boldsymbol{2.246825837}$	$\boldsymbol{2.381863455}$	$\boldsymbol{13.59084169}$	$\boldsymbol{0.027912486}$	$\boldsymbol{0.222182584}$

It is evident from Table 5 that the proposed *DGE+* algorithm performed better and superior to all the state-of-the-art methods without any violation.

The convergence curve shows the function values versus the number of generations for the constrained problem 3. The 30 trials of the best solution obtained from the *DGE+* algorithm are given in Figure 6.

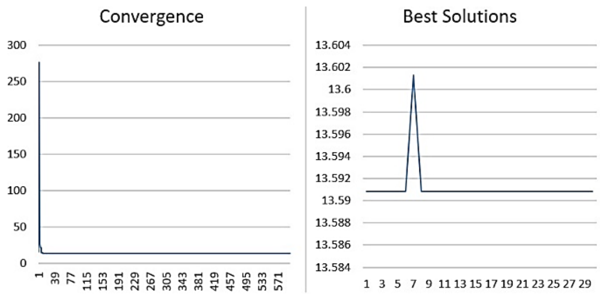

Figure 6. Convergence curve and 30 best solutions for constraint problem 3

4.2.4. Constrained problem 4

This problem taken from [33] which is a relatively simple constrained problem of minimization having two variables and two inequality constraints.

$$\min f(x) = -\frac{Sin^3(2\pi x_1)Sin(2\pi x_2)}{{x_1}^3(x_1 + x_2)}$$

$$subject\ to \begin{cases} h_1(x) = {x_1}^2 - x_2 + 1 \le 0 \\ h_2(x) = 1 - x_1 + (x_2 - 4)^2 \le 0 \end{cases}$$

$$0 \le x_1, x_2 \le 10$$

Table 6. Reported result for constrained problem 4 from DGE+

$Methods$	$Design\ variables$		$f(x)$	$Constraints$	
	x_1	x_2		$h_1(x)$	$h_2(x)$
$DGE+$	1.227971353	4.245373367	−0.0958250	−1.737459724	−0.167763263

Table 7. Statistical comparison of results for constrained problem 4 of various algorithms

$Method$	$Worst$	$Mean$	$Best$	SD
HM	−0.0291438	−0.0891568	−0.0958250	$N.A$
$ASCHEA$	$N.A$	−0.095825	−0.095825	$N.A$
SR	−0.0958250	−0.0958250	−0.0958250	$2.6E-17$
$CAEP$	−0.0958250	−0.0958250	−0.0958250	0
DE	−0.0958250	−0.0958250	−0.0958250	$N.A$
$HPSO$	−0.0958250	−0.0958250	−0.0958250	$1.2E-10$
$NM-PSO$	−0.0958250	−0.0958250	−0.0958250	$3.5E-08$
$CRGA$	−0.095808	−0.095819	−0.095825	$4.40E-06$
$SAPF$	−0.092697	−0.095635	−0.095825	$1.055E-03$
GA	−0.0958250	−0.0958250	−0.0958250	$2.70E-09$
$SMES$	−0.095825	−0.095825	−0.095825	0
$CULDE$	−0.095825	−0.095825	−0.095825	$1E-07$
$DELC$	−0.095825	−0.095825	−0.095825	$1.0E-17$
$DEDS$	−0.095825	−0.095825	−0.095825	$4.0E-17$
$HEAA$	−0.095825	−0.095825	−0.095825	$2.8E-17$
ISR	−0.095825	−0.095825	−0.095825	$2.7E-17$
$Simplex$	−0.095825	−0.095825	−0.095825	$3.8E-13$
ABC	−0.0958250	−0.095825	−0.095825	0
MBA	−0.0958250	−0.0958250	−0.0958250	0
$\mathbf{DGE+}$	$\mathbf{-0.093743605}$	$\mathbf{-0.095748202}$	$\mathbf{-0.0958250}$	$\mathbf{0.00037334}$

Table 6 represents the best solution and the value of corresponding design variables by using the *DGE+* algorithm. The results obtained by *DGE+* satisfies all constraints for the final solution, also compared with 19 state-of-the-art algorithms which are abbreviated and listed in Table 1.

It is evident from Table 7 that the proposed *DGE+* algorithm performed better and superior to all the state-of-the-art methods without any violation. The convergence curve shows the function values versus the number of generations for the constrained problem 4. The 30 trials of the best solution obtained from the *DGE+* algorithm are given in Figure 7.

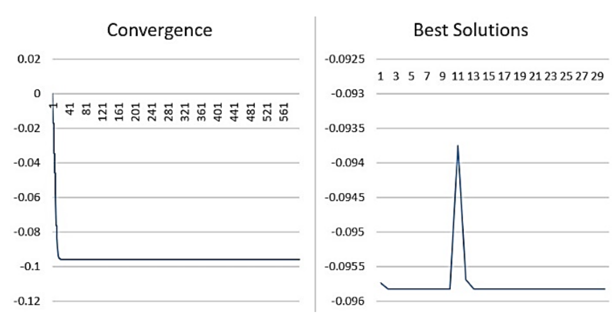

Figure 7. Convergence curve and 30 best solutions for constraint problem 4

4.2.5. Constrained problem 5

This problem is taken from [33] which is a relatively simple constrained problem of minimization having two variables and two inequality constraints.

$$\min f(x) = (x_1 - 10)^3 + (x_2 - 20)^3$$
$$subject\ to \begin{cases} h_1(x) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \geq 0 \\ h_2(x) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0 \end{cases}$$
$$13 \leq x_1 \leq 100, 0 \leq x_2 \leq 100.$$

Table 8. Reported result for constrained problem 5 from DEG+

Methods	***Design variables***		***f(x)***	***Constraints***	
	x_1	x_2		$h_1(x)$	$h_2(x)$
DGE +	14.095	0.842961	−6961.813644	165.4380518	−1.75248*E* − 06

Table 9. Statistical comparison of results for constrained problem 5 of various algorithms

Method	***Worst***	***Mean***	***Best***	***SD***
HM	−5473.9	−6342.6	−6952.1	*N.A*
PSO − DE	−6961.81388	−6961.81388	−6961.81388	2.3*E* − 09
ISR	−6961.814	−6961.814	−6961.814	1.9*E* − 12
HEAA	−6961.814	−6961.814	−6961.814	4.6*E* − 12
ABC	−6961.805	−6961.813	−6961.814	2*E* − 03
FSA	−6961.8139	−6961.8139	−6961.8139	0
PSO	−6961.81381	−6961.81387	−6961.81388	6.5*E* − 06
CRGA	−6077.123	−6740.288	−6956.251	2.70*E* + 2
DEDS	−6961.814	−6961.814	−6961.814	0
MBA	−6961.813875	−6961.813875	−6961.813875	0
ASCHEA	*N.A*	−6961.81	−6961.81	*N.A*
SR	−6350.262	−6875.940	−6961.814	160
SMES	−6962.482	−6961.284	−6961.814	1.85
DELC	−6961.814	−6961.814	−6961.814	7.3*E* − 10
SAPF	−6943.304	−6953.061	−6961.046	5.876
GA	−6961.8139	−6961.8139	−6961.8139	0
DE	−6961.814	−6961.814	−6961.81	*N.A*
CULDE	−6961.813876	−6961.813876	−6961.813876	1*E* − 07
NM − PSO	−6961.8240	−6961.8240	−6961.8240	0
Simplex	−6961.814	−6961.814	−6961.814	1.3*E* − 10
DGE +	**−6961.813894**	**−6961.813894**	**−6961.813894**	**0**

Table 8 represents the best solution and the value of corresponding design variables by using the *DGE+* algorithm. The results obtained by *DGE+* satisfies all constraints for the final solution, also compared with 21 state-of-the-art algorithms which are abbreviated and listed in Table 1.

It is evident from Table 9 that the proposed *DGE+* algorithm performed better and superior to all the state-of-the-art methods without any violation. The convergence curve shows the function values versus the number of generations for the constrained problem 4. The 30 trials of the best solution obtained from the *DGE+* algorithm are given in Figure 8.

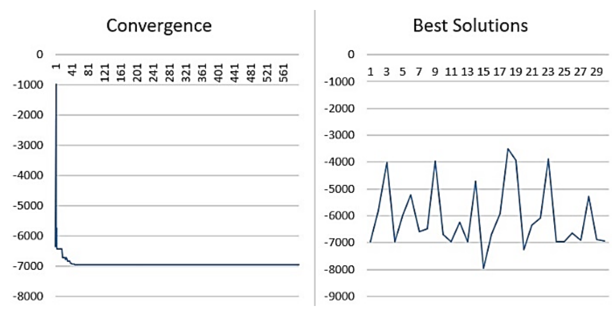

Figure 8. Convergence curve and 30 best solutions for constraint problem 5

4.2.6. Constrained problem 6

This problem is taken from [33] which is a relatively complex constrained problem of minimization having seven variables and four inequality constraints.

$$min f(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3{}^4 + 3(x_4 - 11)^2 + 10x_5{}^6 + 7x_6{}^2 + x_7{}^4 - 4x_6x_7 - 10x_6 - 8x_7,$$

$$subject\ to$$
$$\begin{cases} h_1(x) = 127 - 2x_1{}^2 - 3x_2{}^4 - x_3 - 4x_4{}^2 - 5x_5 \geq 0, \\ h_2(x) = 282 - 7x_1 - 3x_2 - 10x_3{}^2 - x_4 + x_5 \geq 0, \\ h_3(x) = 196 - 23x_1 - x_2{}^2 - 6x_6{}^2 + 8x_7 \geq 0, \\ h_4(x) = -4x_1{}^2 - x_2{}^2 + 3x_1x_2 - 2x_3{}^2 - 5x_6 + 11x_7 \geq 0, \end{cases}$$
$$-10 \leq x_1, x_2, x_3, x_4, x_5, x_6, x_7 \leq 10.$$

Table 10 demonstrates the comparison of the best solution among the different optimizers and the corresponding design variables. The results obtained by *DGE+*satisfies all constraints for the final solution are compared with 25 state-of-the-art algorithms that are abbreviated and listed in Table 1.

Table 10. Reported results for constrained problem 6 from different optimizers

$\boldsymbol{Methods}$	$\boldsymbol{Design\ variables}$							$\boldsymbol{f(x)}$
	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{x_3}$	$\boldsymbol{x_4}$	$\boldsymbol{x_5}$	$\boldsymbol{x_6}$	$\boldsymbol{x_7}$	
IGA	2.330499	1.951372	−0.477541	4.365726	−0.624487	1.038131	1.594227	680.63006
HS	2.323456	1.951242	−0.448467	4.361919	−0.630075	1.03866	1.605348	680.6413574
MBA	2.326585	1.950973	−0.497446	4.367508	−0.618578	1.043839	1.595928	680.6322202
$\boldsymbol{DGE+}$	**2.330404**	**1.95135**	**−0.47779**	**4.365786**	**−0.62427**	**1.038215**	**1.594204**	**680.63**

Table 11. Reported results for constrained problem 6 from different optimizers (continued)

$\boldsymbol{Methods}$	$\boldsymbol{f(x)}$	$\boldsymbol{Constraints}$			
		$\boldsymbol{h_1(x)}$	$\boldsymbol{h_2(x)}$	$\boldsymbol{h_3(x)}$	$\boldsymbol{h_4(x)}$
IGA	680.63006	$4.46E-05$	252.561723	144.878190	$7.63E-06$
HS	680.6413574	0.208928	252.878859	145.123347	0.263414
MBA	680.6322202	$1.17E-04$	252.400363	144.912069	$1.39E-04$
$\boldsymbol{DGE+}$	**680.63**	$\boldsymbol{7.90E-8}$	**252.5603**	**144.8792**	$\boldsymbol{2.42E-07}$

Table 12. Statistical comparison of results for constrained problem 6 of various algorithms

$\boldsymbol{Method}$	$\boldsymbol{Worst}$	$\boldsymbol{Mean}$	$\boldsymbol{Best}$	$\boldsymbol{SD}$
GA	680.6538	680.6381	680.6303	$6.61E-03$
$ASCHEA$	$N.A$	680.641	680.630	$N.A$
$CULDE$	680.630057	680.630057	680.630057	$1E-07$
$CRGA$	682.965	681.347	680.726	$5.70E-01$
$Simplex$	680.630	680.630	680.630	$2.9E-10$
HM	683.1800	681.1600	680.9100	$4.11E-02$
$GA1$	680.6508	680.6417	680.6344	$N.A$
MBA	680.7882	680.6620	680.6322	$3.30E-02$
$GA2$	$N.A$	$N.A$	680.642	$N.A$
$SAPF$	682.081	681.246	680.773	0.322
SR	680.763	680.656	680.63	0.034
HS	$N.A$	$N.A$	680.6413	$N.A$
DE	680.144	680.503	680.771	0.67098
IGA	680.6304	680.6302	680.6301	$1.00E-05$
PSO	684.5289146	680.9710606	680.6345517	$5.1E-01$
$CPSO-GD$	681.371	680.7810	680.678	0.1484
$SMES$	680.719	680.643	680.632	$1.55E-02$
$DELC$	680.630	680.630	680.630	$3.2E-12$
$DEDS$	680.630	680.630	680.630	$2.9E-13$
$HEAA$	680.630	680.630	680.630	$5.8E-13$
ISR	680.630	680.630	680.630	$3.2E-13$
$PESO$	680.630	680.630	680.631	$N.A$
$CoDE$	685.144	681.503	680.771	$N.A$
ABC	680.638	680.640	680.634	$4E-03$
$TLBO$	680.638	680.633	680.630	$N.A$
$\boldsymbol{DGE+}$	**680.6974951**	**680.6340181**	**680.63**	**0.012065102**

It is evident from Tables 10 & 11 that the proposed *DGE+* algorithm performed better and superior to all the state-of-the-art methods without any violation. The convergence curve shows the function values versus the number of generations for the constrained problem 1. The 30 trials of the best solution obtained from the *DGE+* algorithm are given in Figure 9.

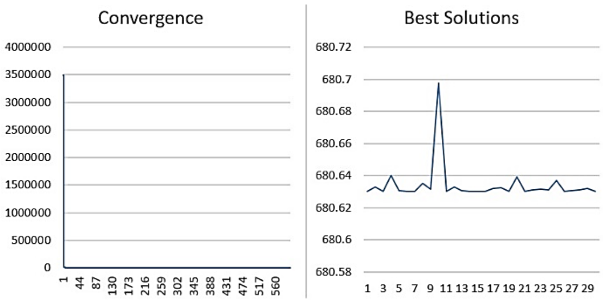

Figure 9. Convergence curve and 30 best solutions for constraint problem 6

4.2.7. Constrained problem 7

This problem is taken from [33] which is a relatively complex constrained problem of minimization having five variables and six inequality constraints. Table 12 demonstrates the comparison of the best solution among the different optimizers and the corresponding design variables. The results obtained by *DGE+* are compared with 5 state-of-the-art algorithms that are abbreviated and listed in Table 1. *CULDE*, Harmony search and *GA2* violet two constraints and remaining methods violet first constraint for the final solution but *DGE+* satisfies all constraints for the final solution.

$$\min f(x) = 5.3578547 x_3^3 + 0.8356891\, x_1 x_5 + 37.293239 x_1 + 40729.141,$$

$$subject\ to \begin{cases} h_1(x) = 85.334407 + 0.0056858 x_2 x_5 + 0.0006262 x_1 x_4 - 0.0022053 x_3 x_5 - 92 \le 0, \\ h_2(x) = -85.334407 - 0.0056858 x_2 x_5 - 0.0006262 x_1 x_4 - 0.0022053 x_3 x_5 \le 0, \\ h_3(x) = 80.51249 + 0.0071317 x_2 x_5 + 0.0029955 x_1 x_2 + 0.0021813 x_3^2 - 110 \le 0, \\ h_4(x) = -80.51249 - 0.0071317 x_2 x_5 - 0.0029955 x_1 x_2 - 0.0021813 x_3^2 + 90 \le 0, \\ h_5(x) = 9.300961 + 0.0047026 x_3 x_5 + 0.0012547 x_1 x_3 + 0.0019085 x_3 x_4 - 25 \le 0, \\ h_6\{x\} = -9.300961 - 0.0047026 x_3 x_5 - 0.0012547 x_1 x_3 - 0.0019085 x_3 x_4 + 20 \le 0, \end{cases}$$

$$78 \le x_1 \le 102, 33 \le x_2 \le 45, 27 \le x_3, x_4, x_5 \le 45.$$

Table 13. Reported results for constrained problem 7 from different optimizers

Methods	***Design variables***					$\boldsymbol{f(x)}$
	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{x_3}$	$\boldsymbol{x_4}$	$\boldsymbol{x_5}$	
$CULDE$	78.000000	33.000000	29.995256	45.000000	36.775813	−30665.5386
HS	78.0	33.0	29.995	45.0	36.776	−30665.500
$GA1$	80.39	35.07	32.05	40.33	33.34	−30005.700
$GA2$	78.0495	33.007	27.081	45.00	44.94	−31020.859
MBA	78.00000	33.00000	29.99526	44.99999	36.77581	−30665.5386
DGE+	**78**	**33**	**29.99525603**	**45**	**36.77581**	**−30665.5386**

Table 14. Reported results for constrained problem 7 from different optimizers (continued)

Methods	$\boldsymbol{f(x)}$	***Constraints***					
		$\boldsymbol{h_1(x)}$	$\boldsymbol{h_2(x)}$	$\boldsymbol{h_3(x)}$	$\boldsymbol{h_4(x)}$	$\boldsymbol{h_5(x)}$	$\boldsymbol{h_6(x)}$
$CULDE$	−30665.5386	$1.35E-08$	−92.00000001	−11.15945	−8.840500	−4.999999	$4.12E-09$
HS	−30665.500	$4.34E-05$	−92.000043	−11.15949	−8.840510	−5.000064	$6.49E-05$
$GA1$	−30005.700	−0.343809	−91.656190	−10.463103	−9.536896	−4.974473	−0.025526
$GA2$	−31020.859	1.283813	−93.283813	−9.592143	−10.407856	−4.998088	$1.91E-03$
MBA	−30665.5386	$1.33E-08$	−91.99999	−11.159499	−8.84050	−4.99999	$-3.06E-09$
DGE+	**−30665.5386**	**0**	**−92**	**−11.15949969**	**−8.840500309**	**−5**	**0**

The results obtained by *DGE+* are also compared with 20 state-of-the-art algorithms, the comparison of statistical results for constrained problem 7 is given in Table 13.

It is evident from Table 12 & 13 that the proposed *DGE+* algorithm performed better and superior to all the state-of-the-art methods without any violation. The convergence curve shows the function values versus the number of generations for the constrained problem 1. The 30 trials of the best solution obtained from the *DGE+* algorithm are given in Figure 10.

Table 15. Statistical comparison of results for constrained problem 7 of various algorithms

Method	*Worst*	*Mean*	*Best*	*SD*
MBA	−30665.3300	−30665.5182	−30665.5386	5.08E − 02
ASCHEA	N.A	−30665.5	−30665.5	N.A
SR	−30665.539	−30665.539	−30665.539	2E − 05
ISR	−30665.539	−30665.539	−30665.539	1.1E − 11
CAEP	−30662.200	−30662.500	−30665.500	9.3
HEAA	−30665.539	−30665.539	−30665.539	7.4E − 12
SAPF	−30656.471	−30655.92	−30665.401	2.043
HPSO	−30665.539	−30665.539	−30665.539	1.7E − 06
HS	N.A	N.A	−30665.500	N.A
DE	−30665.509	−30665.536	−30665.539	5.067E − 03
SMES	−30665.539	−30665.539	−30665.539	0
CRGA	−30660.313	−30664.398	−30665.520	1.6
ABC	−30665.539	−30665.539	−30665.539	0
CULDE	−30665.5386	−30665.5386	−30665.5386	1E − 07
DEDS	−30665.539	−30665.539	−30665.539	2.7E − 11
PSO − DE	−30665.5387	−30665.5387	−30665.5387	8.3E − 10
HM	−30645.900	−30665.300	−30664.500	N.A
DELC	−30665.539	−30665.539	−30665.539	1.0E − 11
Simplex	−30665.539	−30665.539	−30665.539	4.2E − 11
PSO	−30252.3258	−30570.9286	−30663.8563	81
DGE +	**−30665.53823**	**−30665.539**	**−30665.53867**	**9.26E − 05**

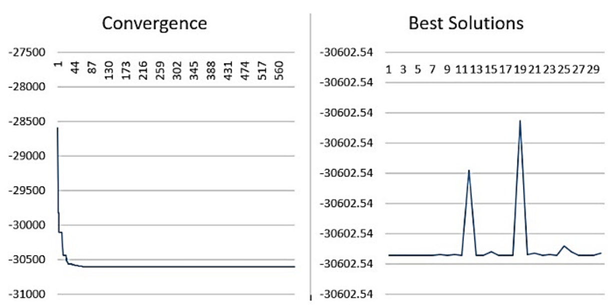

Figure 10. Convergence curve and 30 best solutions for constraint problem 7

5. Conclusions

A new hybrid meta-heuristic has been presented in this paper, called *DGE+*, for dealing with seven benchmark constraint optimization problems. The main motivation behind the present study is to combine the desirable explorative features of *DE* with exploitative features of *GE* algorithms. The proposed method is mainly based on Differential Evolution, Gradient Evolution, and novel jumping technique. The proposed algorithm hybridizes the above-mentioned algorithms with the help of an improvised dynamic probability distribution, additionally provides a new shake off method to avoid premature convergence towards local minima. To evaluate the efficiency and robustness of *DGE+* it has been applied on seven benchmark constraint optimization problems, the results of comparison revealed that *DGE+* can provide very compact, competitive and promising results. As future works, various research directions can be followed. Based on certain preliminary observations, the parameter values for *DGE+* are modified. A full sensitivity analysis on the impact of parameters may, therefore, be a guideline for future research. The implementation of the proposed algorithm to several real-world problems is also extremely valuable.

Acknowledgment

The authors thank the reviewers and editors for their useful comments, which led to the improvement of the content of the paper.

References

[1] Khalilpourazari, S. & Khalilpourazary. S. (2018). Optimization of production time in the multi-pass milling process via a robust grey wolf optimizer. *Neural Computing and Applications*. 29(12), 1321-1336.

[2] Yang, X.-S. (2010). *Nature-inspired metaheuristic algorithms*. Luniver press.

[3] Gandomi, A. H., Yang, X.-S. & Alavi, A. H. (2011). Mixed variable structural optimization using firefly algorithm. *Computers & Structures*. 89(23-24), 2325-2336.

[4] Zhang, L., et al. (2016). A novel hybrid firefly algorithm for global optimization. *PloS one*. 11(9), e0163230.

[5] Alba, E. & Dorronsoro, B. (2005). The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. *IEEE transactions on evolutionary computation*. 9(2), 126-142.

[6] Olorunda, O. and Engelbrecht, A. P. (2008). *Measuring exploration/exploitation in particle swarms using swarm diversity*. in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*.

[7] Lozano, M. & García-Martínez, C. (2010). Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report. *Computers & Operations Research*. 37(3), 481-497.

[8] Simon, D. (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation*. 12(6), 702-713.

[9] Storn, R. (1996). *On the usage of differential evolution for function optimization*. in *Proceedings of North American Fuzzy Information Processing*. IEEE.

[10] Beyer, H.-G. & Schwefel, H.-P. (2002). Evolution strategies–a comprehensive introduction. *Natural computing*. 1(1), 3-52.

[11] Bonabeau, E., Dorigo, M. & Theraulaz, G. (1999). *From natural to artificial swarm intelligence*. Oxford university press, UK.

[12] Koza, J.R. & J.R. Koza. (1992). *Genetic programming: On the programming of computers by means of natural selection*. MIT press.

[13] Alatas, B. (2011). Acroa: Artificial chemical reaction optimization algorithm for global optimization. *Expert Systems with Applications*. 38(10), 13170-13180.

[14] Erol, O. K. & I. Eksin. (2006). A new optimization method: Big bang–big crunch. *Advances in Engineering Software*. 37(2), 106-111.

[15] Rashedi, E., H. Nezamabadi-Pour, & S. Saryazdi. (2009). Gsa: A gravitational search algorithm. *Information sciences*. 179(13), 2232-2248.

[16] Kaveh, A. & M. Khayatazad. (2012). A new meta-heuristic method: Ray optimization. *Computers & Structures*. 112: p. 283-294.

[17] Kirkpatrick, S., C. D. Gelatt, & M. P. Vecchi. (1983). Optimization by simulated annealing. *science*. 220(4598), 671-680.

[18] Du, H., X. Wu, & J. Zhuang. (2006) *Small-world optimization algorithm for function optimization*. in *International Conference on Natural Computation*. Springer.

[19] Evirgen, F., & Yavuz, M. (2018). An alternative approach for nonlinear optimization problem with Caputo-Fabrizio derivative. *In ITM Web of Conferences* (Vol. 22, p. 01009). EDP Sciences.

[20] Evirgen, F., & Özdemir, N. (2012). A fractional order dynamical trajectory approach for optimization problem with HPM. *In Fractional Dynamics and Control* (pp. 145-155). Springer, New York, NY.

[21] Evirgen, F. (2017). Conformable Fractional Gradient Based Dynamic System for Constrained Optimization Problem. *Acta Physica Polonica A*, 132(3), 1066-1069.

[22] Evirgen, F. (2016). Analyze the optimal solutions of optimization problems by means of fractional gradient based system using VIM. *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, 6(2), 75-83.

[23] Evirgen, F. (2017). Solution of a Class of Optimization Problems Based on Hyperbolic Penalty Dynamic Framework. *Acta Physica Polonica A*, 132(3), 1062-1065.

[24] Jumani, T. A., et al. (2020). Jaya optimization algorithm for transient response and stability enhancement of a fractional-order PID based automatic voltage regulator system. *Alexandria Engineering Journal*, 59(4), 2429-2440.

[25] Al-Dhaifallah, M., et al. (2018). Optimal parameter design of fractional order control based INC-MPPT for PV system. *Solar Energy*, 159, 650-664.

[26] Bitirgen, R., Hancer, M., & Bayezit, I. (2018). All Stabilizing State Feedback Controller for Inverted Pendulum Mechanism. *IFAC-PapersOnLine*, 51(4), 346-351.

[27] Stützle, T., et al. (2011). *Parameter adaptation in ant colony optimization*, in Autonomous search. Springer. 191-215.

[28] Yang, X.-S. (2010). *A new metaheuristic bat-inspired algorithm*, in Nature inspired cooperative strategies for optimization (nicso 2010). Springer. 65-74.

[29] Lu, X. and Y. Zhou. (2008). A novel global convergence algorithm: Bee collecting pollen algorithm. in *International Conference on Intelligent Computing*. 2008. Springer.

[30] Singh, H., et al. (2019). A reliable numerical algorithm for the fractional klein-gordon equation. *Engineering Transactions*. 67(1), 21–34.

[31] Kennedy, J. & R. Eberhart. *Particle swarm optimization (pso)*. in *Proc IEEE International Conference on Neural Networks, Perth, Australia*. 1995.

[32] Kaveh, A. & V. Mahdavi. (2014). Colliding bodies optimization: A novel meta-heuristic method. *Computers & Structures*. 139: p. 18-27.

[33] Sadollah, A., et al. (2013). Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Applied Soft Computing*. 13(5), 2592-2612.

[34] Wang, B., C. Liu, & H. Wu. (2014). *The research of pattern synthesis of linear antenna array based on seeker optimization algorithm*. in *2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*. IEEE.

[35] He, S., Q. H. Wu, & J. Saunders. (2009). Group search optimizer: An optimization algorithm inspired by animal searching behavior. *IEEE transactions on evolutionary computation*. 13(5), 973-990.

[36] Ramezani, F. & Lotfi, S. (2013). Social-based algorithm (sba). *Applied Soft Computing*. 13(5), 2837-2856.

[37] Lu, Y., et al. (2010). An adaptive chaotic differential evolution for the short-term hydrothermal generation scheduling problem. *Energy Conversion and Management*. 51(7), 1481-1490.

[38] Lu, Y., et al. (2010). An adaptive hybrid differential evolution algorithm for dynamic economic dispatch with valve-point effects. *Expert Systems with Applications*. 37(7), 4842-4849.

[39] Chang, L., et al. (2012). A hybrid method based on differential evolution and continuous ant colony optimization and its application on wideband antenna design. *Progress in electromagnetics research*. 122: p. 105-118.

[40] Abdullah, A., et al. (2013). An evolutionary firefly algorithm for the estimation of nonlinear biological model parameters. *PloS one*. 8(3), e56310.

[41] Niknam, T., Azizipanah-Abarghooee, R. & Aghaei, J. (2012). A new modified teaching-learning algorithm for reserve constrained dynamic economic dispatch. *IEEE Transactions on power systems*. 28(2), 749-763.

[42] Bhattacharya, A. and Chattopadhyay, P. K. (2010). Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch. *IEEE Transactions on power systems*. 25(4), 1955-1964.

[43] Kuo, R. and Zulvia, F. E. (2015). The gradient evolution algorithm: A new metaheuristic. *Information Sciences*. 316: p. 246-265.

[44] Bazaraa, M. S., H. D. Sherali, & C. M. Shetty. (2013). *Nonlinear programming: Theory and algorithms*. John Wiley & Sons.

[45] Wang, S.-K., J.-P. Chiou, & C.-W. Liu. (2007). Non-smooth/non-convex economic dispatch by a novel hybrid differential evolution algorithm. *IET Generation, Transmission & Distribution*. 1(5), 793-803.

[46] Chiou, J.-P. (2007). Variable scaling hybrid differential evolution for large-scale economic dispatch problems. *Electric Power Systems Research*. 77(3-4), 212-218.

[47] Storn, R. & K. Price. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*. 11(4), 341-359.

[48] Kuo, R. & F. E. Zulvia. *Cluster analysis using a gradient evolution-based k-means algorithm*. in *2016 IEEE Congress on Evolutionary Computation (CEC)*. 2016. IEEE.

[49] Mezura-Montes, E. & C. A. C. Coello. (2008). An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *International Journal of General Systems*. 37(4), 443-473.

[50] Kaveh, A. & S. Talatahari. (2009). A particle swarm ant colony optimization for truss structures with discrete variables. *Journal of Constructional Steel Research*. 65(8-9), 1558-1568.

[51] Koziel, S. & Z. Michalewicz. (1999). Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. *Evolutionary computation*. 7(1), 19-44.

[52] Runarsson, T. P. & X. Yao. (2000). Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*. 4(3), 284-294.

[53] Parsopoulos, K. E. & M. N. Vrahatis. (2005). *Unified particle swarm optimization for solving constrained engineering optimization problems*. in *International conference on natural computation*. Springer.

[54] Karaboga, D. & B. Basturk. (2007). *Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems*. in *International fuzzy systems association world congress*. Springer.

[55] Akay, B. & D. Karaboga. (2012). Artificial bee colony algorithm for large-scale problems and engineering design optimization. *Journal of intelligent manufacturing*. 23(4), 1001-1014.

[56] Geem, Z. W., J. H. Kim, & G. V. Loganathan. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation*. 76(2), 60-68.

[57] Lee, K. S. & Z. W. Geem. (2005). A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering*. 194(36-38), 3902-3933.

[58] Farooq, H. & M. T. Siddique. (2014). A comparative study on user interfaces of interactive genetic algorithm. *Procedia Computer Science*. 32: p. 45-52.

[59] Amirjanov, A. (2008). Investigation of a changing range genetic algorithm in noisy environments. *International journal for numerical methods in engineering*. 73(1), 26-46.

[60] Hamida, S. B. & M. Schoenauer. (2002). *Aschea: New results using adaptive segregational constraint handling*. in *Proceedings of the 2002 Congress on Evolutionary Computation CEC'02 (Cat No 02TH8600)*. IEEE.

[61] Krohling, R. A. & L. dos Santos Coelho. (2006). Coevolutionary particle swarm optimization using gaussian distribution for solving constrained optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*. 36(6), 1407-1416.

[62] Coello Coello, C. A. & R. L. Becerra. (2004). Efficient evolutionary optimization through the use of a cultural algorithm. *Engineering Optimization*. 36(2), 219-236.

[63] Huang, F.-z., L. Wang, & Q. He. (2007). An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics and Computation*. 186(1), 340-356.

[64] Zahara, E. & Y.-T. Kao. (2009). Hybrid nelder–mead simplex search and particle swarm optimization for constrained engineering design problems. *Expert Systems with Applications*. 36(2), 3880-3886.

[65] Becerra, R. L. & C. A. C. Coello. (2006). Cultured differential evolution for constrained optimization. *Computer Methods in Applied Mechanics and Engineering*. 195(33-36), 4303-4322.

[66] Muñoz Zavala, A. E., A. H. Aguirre, & E. R. Villa Diharce. (2005). *Constrained optimization via particle evolutionary swarm optimization algorithm (peso)*. in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*. ACM.

[67] Tessema, B. & G. G. Yen. (2006). *A self adaptive penalty function based algorithm for constrained optimization*. in *2006 IEEE International Conference on Evolutionary Computation*. IEEE.

[68] Lampinen, J. (2002). *A constraint handling approach for the differential evolution algorithm*. in *Proceedings of the 2002 Congress on Evolutionary Computation CEC'02 (Cat No 02TH8600)*. IEEE.

[69] Fogel, D. B. (1995). A comparison of evolutionary programming and genetic algorithms on selected constrained optimization problems. *Simulation*. 64(6), 397-404.

[70] Amirjanov, A. (2006). The development of a changing range genetic algorithm. *Computer Methods in Applied Mechanics and Engineering*. 195(19-22), 2495-2508.

[71] Chootinan, P. & A. Chen. (2006). Constraint handling in genetic algorithms using a gradient-based repair method. *Computers & operations research*. 33(8), 2263-2281.

[72] Gupta, S., R. Tiwari, & S. B. Nair. (2007). Multi-objective design optimisation of rolling bearings using genetic algorithms. *Mechanism and Machine Theory*. 42(10), 1418-1443.

[73] Zhang, M., W. Luo, & X. Wang. (2008). Differential evolution with dynamic stochastic selection for constrained optimization. *Information Sciences*. 178(15), 3043-3074.

[74] Wang, L. & L.-p. Li. (2010). An effective differential evolution with level comparison for constrained engineering design. *Structural and Multidisciplinary Optimization*. 41(6), 947-963.

[75] Hedar, A.-R. & M. Fukushima. (2006). Derivative-free filter simulated annealing method for constrained continuous global optimization. *Journal of global optimization*. 35(4), 521-549.

[76] Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*. 186(2-4), 311-338.

[77] Runarsson, T. P. & X. Yao. (2005). Search biases in constrained evolutionary optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*. 35(2), 233-243.

[78] Mezura-Montes, E. & C. A. C. Coello. (2005). A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Transactions on Evolutionary Computation*. 9(1), 1-17.

[79] Michalewicz, Z. (1995). *Genetic algorithms, numerical optimization, and constraints*. in *Proceedings of the sixth international conference on genetic algorithms*. Citeseer.

[80] Rao, R. V., V. J. Savsani, & D. Vakharia. (2011). Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design*. 43(3), 303-315.

[81] Wang, Y., et al. (2009). Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. *Structural and Multidisciplinary Optimization*. 37(4), 395-413.

[82] de Fátima Araújo, T. & W. Uturbey. (2013). Performance assessment of pso, de and hybrid pso–de algorithms when applied to the dispatch of generation and demand. *International Journal of Electrical Power & Energy Systems*. 47: p. 205-217.

[83] Liu, H., Z. Cai, & Y. Wang. (2010). Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Applied Soft Computing*. 10(2), 629-640.

[84] Bracken, J. & G. P. McCormick (1968). Selected applications of nonlinear programming. Research Analysis Corp Mclean.

Muhammad Farhan Tabassum *is working as Assistant Professor at the University of Lahore, Pakistan and currently pursuing his PhD from UMT Lahore. He has published more than 30 research papers. His research interests are Operations research, Optimization, Numerical analysis, Algorithmic development and Multicriteria decision making. He has more than eight years of teaching experience at the university level and supervised the thesis of M.Phil. Mathematics students.*

http://orcid.org/0000-0002-9958-5015

Sana Akram *is working as Assistant Professor in Lahore Garrison University, Lahore, also doing a Ph.D form UMT Lahore, She published more than 40 research papers. Research field is Graph theory, Operations research, Optimization, Numerical Analysis. He has more than seven year teaching experience at University Level also supervised the thesis of M.Phil Mathematics Students.*

https://orcid.org/0000-0003-2038-9511

Saadia Hassan *is currently working as a senior lecturer at the University of Lahore, Pakistan. After completing her MS in linguistics with nine research publications in stylistics, discourse analysis, sports sciences and physical education and translation analysis. With ample experience of research which she gained as a research scholar at the University of Punjab and co-supervision of more than 5 scholars helped her to enliven her hidden potentials. As a PhD scholar, she has plans to endeavour excellence in the domain of applied linguistics.*

http://orcid.org/0000-0003-1852-2854

Rabia Karim *is currently serving as Senior Lecturer at the University of Lahore, Pakistan and teaching sports management, sports modern technology and Sports sociology. She has 4 publications at her credit in different journals. Before devoting herself to this field, she has worked as Manager National Sports Events at the Sports Board Punjab. She has also played a leading role in developing and establishing several Cricket Coaching Academies, both for Girls and Boys, in various cities of Pakistan.*

http://orcid.org/0000-0001-7343-4262

Parvaiz Ahmad Naik *received his PhD in Mathematics from Maulana Azad National Institute of Technology, a leading institute of India, in December 2015 and currently working as Assistant Professor at the Department of Applied Mathematics, Xi'an Jiaotong University P. R. China. Earlier, he was a postdoctoral research fellow and worked with Prof. Jian Zu at the school of Mathematics and Statistics, Xi'an Jiaotong University, from December 2018-December 2019. His research interests mainly focus on infectious disease dynamics, fractional mathematical modeling, fractional mathematical theory and method and bifurcation analysis. He has published more than 20 SCI research papers in international repute journals like World Scientific, Elsevier, Springer, American Scientific, Taylor & Francis etc. He has received two young scientist awards (gold medals) for his outstanding research work in mathematical biology. Furthermore, he presided over one scientific research project at the national level from the China Postdoctoral Science Foundation under grant no. 2019M663653.*

http://orcid.org/ 0000-0001-8967-5992

Muhammad Farman *did his PhD at the University of Lahore, Pakistan. His research field is mathematical biology, Control theory, Numerical analysis. He has more than seven years of teaching and research experience at the university level and supervised the thesis of M.Phil. and PhD Mathematics students. He has published more than 65 research papers in a national and international journal. He completes several projects in fractional order nonlinear dynamical system with the collaboration of global universities.*

http://orcid.org/0000-0001-7616-0500

Mehmet Yavuz *received his PhD in Mathematics from Balıkesir University, Turkey. He visited the University of Exeter, UK. for post-doctoral research in mathematical biology and optimal control theory for a year. He is currently serving as associate professor at Necmettin Erbakan University, Turkey. His research interests mainly focus on infectious disease dynamics, fractional mathematical modeling, fractional mathematical theory and method, optimal control theory and bifurcation analysis. He has published more than 40 research papers in international esteemed journals and he is a reviewer for about seventy international repute journals.*

http://orcid.org/ 0000-0002-3966-6518

Mehraj-ud-din Naik *is currently working as Assistant Professor at the Department of Chemical Engineering, College of Engineering, Jazan University, Saudi Arabia. He received his PhD in Chemical Engineering from Chonbuk National University, South Korea, in August 2009. Besides this, he worked as a postdoctoral research fellow at the Department of Chemical Engineering and Applied Chemistry, Chungnam National University South Korea from September 2009-October 2010 and Department of Physics and Mechanical Engineering, University of Padova, Italy, from January 2011-February 2013. His research interests mainly focus on chemical engineering, catalysis, nanotechnology, nanomaterials, nanoparticles. He has published more than 15 SCI research papers in the journals of international repute and serving as a reviewer to many SCI-indexed journals.*

http://orcid.org/0000-0001-8192-4843

Hijaz Ahmad *works in a number of mathematical areas, but he is primarily interested in developing new numerical techniques for the solution of differential equations. Recently, he has published many papers in high quality journals on modifications of varaitional iteration algorithm-I, algorithm-II and fractional iteration algorithm. He has Ms in Computational Mathematics from COMSATS University, Pakistan and PhD in Computational Mathematics from the*

University of Engineering and Technology Peshawar, Pakistan. He is an associate member of Section of Mathematics, Uninettuno University, Rome, Italy. He is a reviewer for at least fifty international journals, and also serves on the editorial boards for many good international journals.

http://orcid.org/ 0000-0002-5438-5407