INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2021; 11(2): 271-289

A MULTI-PHASE GRADIENT METHOD FOR OPTIMIZATION OF MULTIMODAL PROBLEMS

F. Salajegheh and E. Salajegheh[*, †]
Department of Civil Engineering, Shahid Bahonar University of Kerman, Kerman, Iran

ABSTRACT

An ensemble method is introduced to solve optimization problems efficiently. The method is mainly based on using the gradient directions along which, the function is reduced at most. Large step sizes are employed for exploration in the first phase. The use of smaller step sizes in subsequence phases will allow for more accurate exploration. To increase the efficiency of the gradient techniques, some enhancements such as mutation, crossover and fly-back operations are introduced to explore the entire design space. The efficiency and the reliability of the multi-phase gradient approach are examined by solving 29 complicated multimodal functions introduced in CEC 2017 and a structural shape optimization problem under frequency constraints. The results are compared with several well-known population-based algorithms.

Keywords: multi-phase gradient, multimodal problems, global optimization, CEC 2017.

Received: 20 March 2021; Accepted: 25 May 2021

1. INTRODUCTION

The optimization methods are divided into two main categories. The first category is known as mathematical programming (MP) [1-3]. The MP methods generally require the gradient of objective functions. For this purpose, a sensitivity analysis of the function should be performed. These methods are suitable for convex problems and are efficient in problems with a large number of variables. Steepest descent, sequential linear programming (SLP or gradient descent), Newton-Raphson, Broyden-Fletcher-Goldfarb-Shanno (BFGS) are the most well-known gradient-based methods [1-3]. In recent years, some other MP methods are presented such as momentum [4], adaptive gradient (AdaGrad) [5], root mean square propagation (RMSProp) [6], adaptive moment estimation (Adam) [7], Kalman-based

[*]Corresponding author: Department of Civil Engineering, Shahid Bahonar University of Kerman, Kerman, Iran
[†]E-mail address: eysasala@uk.ac.ir (E. Salajegheh)

stochastic gradient descent (kSGD) [8]. These algorithms are modifications of the classical MP methods and they are known as stochastic gradient descent.

The second category is the metaheuristic algorithms. In these methods, there is no need to calculate the gradient directions. In other words, they are gradient-free techniques. A group (swarm) of particles is distributed in the design domain. The particles exchange information and update their variables to get a better solution. The metaheuristic methods perform well in multimodal problems but require more function evaluations than MP algorithms. Some well-known metaheuristic methods are particle swarm optimization (PSO) [9], gravitational search algorithm (GSA) [10], firefly algorithm (FA) [11], crow search algorithm (CSA) [12], bat algorithm (BA) [13], gray wolf optimization (GWO) [14], whale optimization algorithm (WOA) [15], krill herd (KH) [16], artificial bee colony (ABC) [17], differential evolution (DE) [18], backtracking search optimization algorithm (BSO or BSA) [19], sine cosine algorithm (SCA) [20], genetic algorithm (GA) [21] and tabu-scatter search (TS) [22]. These methods are inspired by natural phenomena, swarm intelligence (e.g. animal behaviors) and genetic subjects. Because of the robustness of metaheuristic methods, the number of these algorithms and their variants are increasing [23-27]. Their ability to optimize multimodal problems attracted great attention of researchers.

Attempts have been made to improve the performance of the MP methods for global optimization. A multi-start model is one of these ideas [28-30]. A random initial population is distributed in the design domain, then a local search will be performed around the population. The process of generating random populations and local searches will be repeated to find the global optimum. Another strategy is choosing proper step sizes for classical gradient methods [31]. Combination of gradient directions and metaheuristic methods is another way to make powerful optimization algorithms. For example, combination of simulated annealing and gradient [32, 33], gradient tabu search [34], gradient-based cuckoo search [35] and PSOG [36]. In most of the hybrid methods, global search (exploration) is performed by metaheuristic methods and gradient-based approaches are responsible for local search (exploitation) [37-41]. Another method for global optimization is NOVEL [42] which is a two-phase supervised learning method that has exploration and exploitation phases and applied for neural network training. It should be noted that stochastic gradient methods have a possible ability to escape from near local optima.

In this study, a multi-phase gradient method is presented and some well-known techniques are applied to achieve a better exploration. In the proposed method, the gradient direction is considered as the basis of the search and the pure ideas from metaheuristic methods will help for better exploration. In the next step, its performance for optimization of multimodal problems is investigated. For this purpose, 29 functions of CEC 2017 [43] are optimized by the proposed method. Moreover, a truss structure is optimized under frequency constraints. The results are compared with several metaheuristic methods.

In the next section, a brief explanation of the gradient methods is provided. Section 3 presents the proposed algorithm. Numerical examples are optimized in Section 4 and the results are compared with other methods. Eventually, the conclusion is presented in Section 5.

2. GRADIENT METHODS

Gradient-based methods are suitable for exploitation. In other words, they are generally useful for optimization of unimodal functions. The algorithms of MP methods are simple. A starting point should be selected. Then, the design variables are updated using a velocity vector (Eq. 1). This velocity vector has different formulations in each gradient-based method [1-3]. In this study, Eq. (2) is employed as updating-direction which is a modification of the momentum gradient method [4]. In the momentum, the gradient of each step is added to the velocity of the previous steps. In Eq. (2), the gradient direction is scaled by its maximum values and then it is multiplied by a predefined factor. As a result, step sizes are more controllable.

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{v}_k \tag{1}$$

$$\boldsymbol{v}_k = \beta \boldsymbol{v}_{k-1} + \alpha_k \frac{-\boldsymbol{\nabla F}_k}{\max(|\boldsymbol{\nabla F}_k|)} \tag{2}$$

where $\boldsymbol{x}$ is a vector that includes design variables and $\boldsymbol{v}$ is the velocity vector. Parameter k is the iteration number. α and β are scaling factors that should be defined by the user. $\boldsymbol{\nabla F}_k$ is the gradient vector at $\boldsymbol{x}_k$. In order to optimize the objective function $\boldsymbol{F}$, Eqs. (1) and (2) must be run in succession iterations. Gradient methods converge very fast and usually trap in a near local optimum. Therefore, the method should be powered which is described in the following sections.

3. PROPOSED METHOD: MPG

Iterative global-optimization methods should have several vital features. First, they should update the design variables to reach promising areas. Promising area means spaces in the domain design where the optimal solutions are likely to exist. For this purpose, the direction of the search and the step size must be taken into account. The negative-gradient directions are generally directed to areas where the objective function is less than the current location. From the mathematical point of view, the gradient direction is one of the best directions that causes the steepest decrease for the objective function. In addition, the step size of the directions must be specified. Considering large step sizes, the changes to the design variables are large. In this case, many spaces will be left unchecked. Taking small steps sizes, only close spaces can be searched. In other words, small and large values for step sizes lead to local and global searches, respectively. The ability to exit from the local minima and search other areas is the second important feature for global search methods.

The next essential feature for optimization methods is local search capability. Gradient-based methods alone are highly capable of this duty. These methods are efficient for unimodal optimization problems. Their exploitation is strong despite their poor exploration. In this paper, the capability of exploration is added to the method by using a multi-phase strategy and some other techniques. In order to explore the design domain, large step sizes for gradient-directions, mutation, moving toward the best result and fly-back techniques are

employed. On the other hand, gradient directions with small step sizes are used for exploitation.

In the proposed method, unlike the metaheuristic methods, only one particle is employed to search the design domain. Although in hybrid metaheuristic-gradient methods, exploration has been performed by metaheuristic and exploitation is carried out by gradient, in the proposed method the gradient plays an important role in both responsibilities. In the following, the phases and techniques of the presented method are described.

3.1 Exploration using gradient directions and phases

Since the gradient of a function is the steepest descent direction, it can be used as a searching-path for exploration (Eq. 3). For this purpose, exploration can be accomplished by using constant or variable step sizes (α). In the initial iterations, a large value is chosen for the step size (Fig. 1a). This results in an inaccurate domain search which is a global search. After exploration using long distances, the step size can be chosen smaller to allow for a more accurate search around the best-obtained answer (**Gbest**) (Fig. 1b).

In the proposed method, the area around **Gbest** is searched several times by different step sizes. Each of these search around the best-obtained result with predefined α is called a "phase". Applying this method alone cannot create a complete search in the design domain. Therefore, there is a need for techniques to make the method to search in other areas. This process will be described in the subsequent sections.

$$\begin{aligned} \boldsymbol{v}_k &= \beta \boldsymbol{v}_{k-1} + \alpha_k \frac{-\boldsymbol{\nabla F}_k}{\max(|\boldsymbol{\nabla F}_k|)} \; which\ \alpha \\ &= \begin{cases} large\ numbers\ in\ exploration\ phases \\ small\ number\ in\ exploitation\ phase \end{cases} \end{aligned} \tag{3}$$

3.2 Mutation

Gradient-based approaches have the great potential to be trapped in a local minimum. Therefore, there is a need to use techniques in order to exit the local points and search in the solution domain. One of these techniques is mutation. Mutation is a well-known operation in the metaheuristic approaches [21]. In order to explore the design domain, decision variables can be changed randomly. In other words, the search-particle flies to another point. With this action, spaces that were difficult to access are more accessible.

There are various ways to implement this technique. Generally, mutation only changes one random design variable in each of its activations. By mutation of only one variable, metaheuristic methods can converge with proper speed. Since the gradient has a very good convergence rate, a different pattern can be used for mutation. In this study, all the design variables are randomly mutated. Then the gradient direction comes to help the mutated particle to reach a promising location in those areas. Also in this paper, the mutation method is proposed as a velocity vector. With this method, all subsequent search directions are also affected. In other words, all updating-velocities will be a combination of random and gradient directions. This step can be performed using the following equation;

$$\boldsymbol{v}_k = \begin{cases} \gamma_k(\boldsymbol{x}_{\max} - \boldsymbol{x}_{\min})(-1 + 2\boldsymbol{random}), \gamma = \begin{cases} large\ numbers\ for\ large\ mutation \\ small\ number\ for\ small\ mutation \end{cases} & if\ \ mutation\ is\ activated \\ \beta \boldsymbol{v}_{k-1} + \alpha_k \dfrac{-\nabla \boldsymbol{F}_k}{\max(|\nabla \boldsymbol{F}_k|)} & otherwise \end{cases} \tag{4}$$

which γ is a factor that determines the size of a mutation. These changes can be defined as large or small mutations (Fig. 1c). Small and large mutations rescue the trapped search-particle in order to explore near and far spaces, respectively. Small mutations are more likely to occur than large mutations. The parameter *random* is a vector with uniform random variables between [0,1]. $\boldsymbol{x}_{\max}$ and $\boldsymbol{x}_{\min}$ are the upper and lower bounds of the design variables, respectively. Each element of the mutation vector, $\boldsymbol{v}_k$, can be randomly positive or negative. This strategy occurs in random iterations of exploration phases. Although these random mutations resulted in appropriate exploration, some regions might remain unsearched. In the next section, some simple techniques are introduced to tackle this difficulty.

3.3 Moving toward the best result

Due to random mutations, spaces in the domain may not be well searched. Different algorithms can be used to fill these gaps around the best solution (**Gbest**). Among these algorithms, three simple approaches are used for this purpose. First, a velocity vector that connects $\mathbf{x}_k$ to **Gbest** is employed (Eq. 5 and Fig. 1e). Therefore, the flown particle gradually attempts to return to the best-obtained position. Second, the crossover (random recombination) is applied to combine $\mathbf{x}_k$ and **Gbest** (Eq. 6 and Fig. 3e). Crossover is a well-known technique in GA and DE algorithms. Using this technique, the position of the search particle is changed and moved to the near **Gbest**. For the third method, **x** returns to **Gbest** and continue to search (Eq. 7). Using this action, the search will be restarted. Then, with smaller step sizes, the method begins to search more accurately in those areas. After a closer search, the near and far regions are searched again by small and large mutations. As mentioned before, each return to **Gbest** and accurate research is referred to as "Phase".

$$\boldsymbol{v}_k = \theta(\boldsymbol{Gbest}_k - \boldsymbol{x}_k) \tag{5}$$

$$\boldsymbol{x}_{k+1} = \text{random combination of } (\boldsymbol{x}_k, \boldsymbol{Gbest}_k) \tag{6}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{Gbest}_k \tag{7}$$

where θ is a factor that scales the velocity vector. These techniques are performed randomly through exploration phases. After exploration, the exploitation phase should be accomplished to find the best result.

3.4 Exploitation

In the exploration phases, different points are explored in the design domain. After many searches, it is very likely that the global optimum is near **Gbest**. So in the final phase, a local search should be performed in the areas near **Gbest**. Gradient-based methods are adequate for exploitation. By choosing small α_k exploitation can be performed around **Gbest**. In other

words, a unimodal region is optimized in the last phase (Fig. 1g).

3.5 Enhancement

In many cases, decision variables are limited to a defined interval. Since mutation is performed using random vectors, many solutions may be moved out of the permissible range (permissible range: $x_{min} \leq x \leq x_{max}$). In this situation, **x** is usually moved to the nearest boundary ($x=x_{min}$ or $x=x_{max}$). In the proposed method, x enters the solution domain by Eqs. (8). This process is a kind of fly-back technique (Fig. 1f).

$$if\ x_{k+1} > x_{max}; x_{k+1} = x_{max} - \rho\Delta;\ \Delta = x_{k+1} - x_{max} \quad (8a)$$

$$if\ x_{k+1} < x_{min}; x_{k+1} = x_{min} + \rho\Delta;\ \Delta = x_{min} - x_{k+1} \quad (8b)$$

where ρ is a scaling parameter that compels the exited **x** to return into the design domain. In order to summarize this Section, the pseudocode of the proposed method is presented in Pseudocode 1. In the next section, complex and multimodal problems are optimized by the proposed method and the results will be compared with several other approaches.

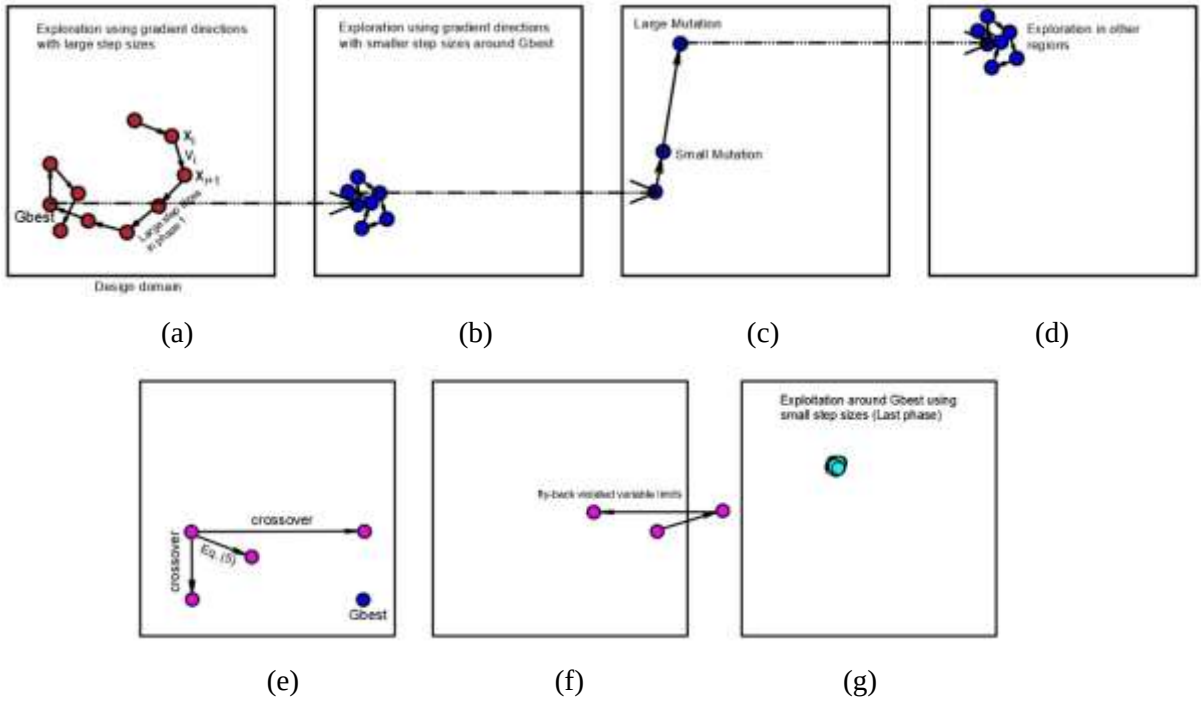


Figure 1. Exploration using gradient directions by: (a) large steps in the first phase (b) small steps in subsequent phases; (c) mutation; (d) exploration after mutation; (e) moving toward *Gbest*; (f) fly-back; (g) exploitation

Pseudocode 1
Proposed algorithm

phase_number=1
Set a large step size for the first phase
Set an initial $\boldsymbol{x}$ ($\boldsymbol{x}_{k=1}$), ***Gbest***= $\boldsymbol{x}_{k=1}$
for *k*=1: *max_iteration*
Calculate objective function: $F(\boldsymbol{x}_k)$
Calculate gradient of the function: $\boldsymbol{\nabla F}_k$
if $F(\boldsymbol{x}_k)$<=min($F(\boldsymbol{x}_{i=1:k})$) and *k*>=2
    Gbest= $\boldsymbol{x}_k$ %Updating ***Gbest***
end if
Based on the parameter *k*, specify phase number
if it is the start of a phase
    $\boldsymbol{x}_k$= ***Gbest*** %turning back to the best-found solution
end if
Based on phase number, specify α_k (a large number for the first phase, smaller numbers for next phases, a small number for exploitation phase)
Calculate the velocity vector: $\boldsymbol{v}_k = \beta \boldsymbol{v}_{k-1} + \alpha_k \frac{-\boldsymbol{\nabla F}_k}{\max(|\boldsymbol{\nabla F}_k|)}$
if $random_1$<=ε_1, in exploration phases
    Small mutation: $\boldsymbol{v}_k = \gamma_k(\boldsymbol{x}_{\max} - \boldsymbol{x}_{\min}).(-1 + 2\boldsymbol{random})$ %(small value for $0 < \gamma < 1$)
end if
if $random_2$<=ε_2, in exploration phases % ε_2< ε_1
    Large mutation: $\boldsymbol{v}_k = \gamma_k(\boldsymbol{x}_{\max} - \boldsymbol{x}_{\min}).(-1 + 2\boldsymbol{random})$ %(large value for $0 < \gamma < 1$)
end if
if $random_3$<=ε_3 and *k*>=2, in exploration phases
    Travel to ***Gbest*** gradually: $\boldsymbol{v}_k = \theta(\boldsymbol{Gbest}_k - \boldsymbol{x}_k)$ %($0 < \theta < 1$)
end if
If $random_4$<=ε_4 and *k*>=2, in exploration phases
    %Crossover with ***Gbest***:
    $\boldsymbol{v}_k = zeros\ vector$
    a= a random integer number< number of design variables
    if $random_5$<=*0.5*
        $\boldsymbol{x}_{k+1}$(1:*a*)= $\boldsymbol{x}_k$(1:*a*), $\boldsymbol{x}_{k+1}$(*a*+1:*end*)= ***Gbest***(*a*+1:*end*)
    else
        $\boldsymbol{x}_{k+1}$(1:*a*)= ***Gbest***(1:*a*), $\boldsymbol{x}_{k+1}$(*a*+1:*end*)= $\boldsymbol{x}_k$(*a*+1:*end*)
    end if
end if
Update the design variables: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{v}_k$
% Fly back:
for *j*=1:*noe* %(*noe*=number of design variables)
    if $\boldsymbol{x}_{k+1}(j) > x_{\max}$

delta= $\boldsymbol{x}_{k+1}(j)$-x_{max}
$\boldsymbol{x}_{k+1}(j)$= x_{max}- ρ delta
end if
if $\boldsymbol{x}_{k+1}(j)<x_{min}$
delta= x_{min} −$\boldsymbol{x}_{k+1}(j)$
$\boldsymbol{x}_{k+1}(j)$= x_{min} + ρ delta
end if
end
%k=k+1 and go to the next iteration
end

%: Comments

4. NUMERICAL EXAMPLES

To evaluate the performance of the proposed method, functions of CEC 2017 [43] have been investigated and the outcomes are compared with the results of several metaheuristic algorithms. CEC 2017 includes 3 unimodal, 7 simple multimodal, 10 hybrid and 10 composition benchmark problems. CEC 2017 benchmarks are shifted, rotated and hybrid of well-known test functions such as; Bent Cigar, Zakharov, Rosenbrock, Rastrigin, Schaffer, Lunacek bi-Rastrigin, Non-Continuous Rastrigin, Levy, Schwefel, Discuss, Ackley, Weierstrass, Griewank, Katsuura, HappyCat and HGBat [43]. Some basic equations of CEC 2017 are presented in Table 1. These functions are single objective problems with bounded constraints on design variables. CEC benchmarks vastly have been optimized by researchers to check and rate their algorithms in dealing with real problems.

In order to visualize the performance of the presented method, several CEC functions are optimized in two-dimensional space. In the next step, the benchmarks with 10, 30 and 50 variables are compared with other well-known methods.

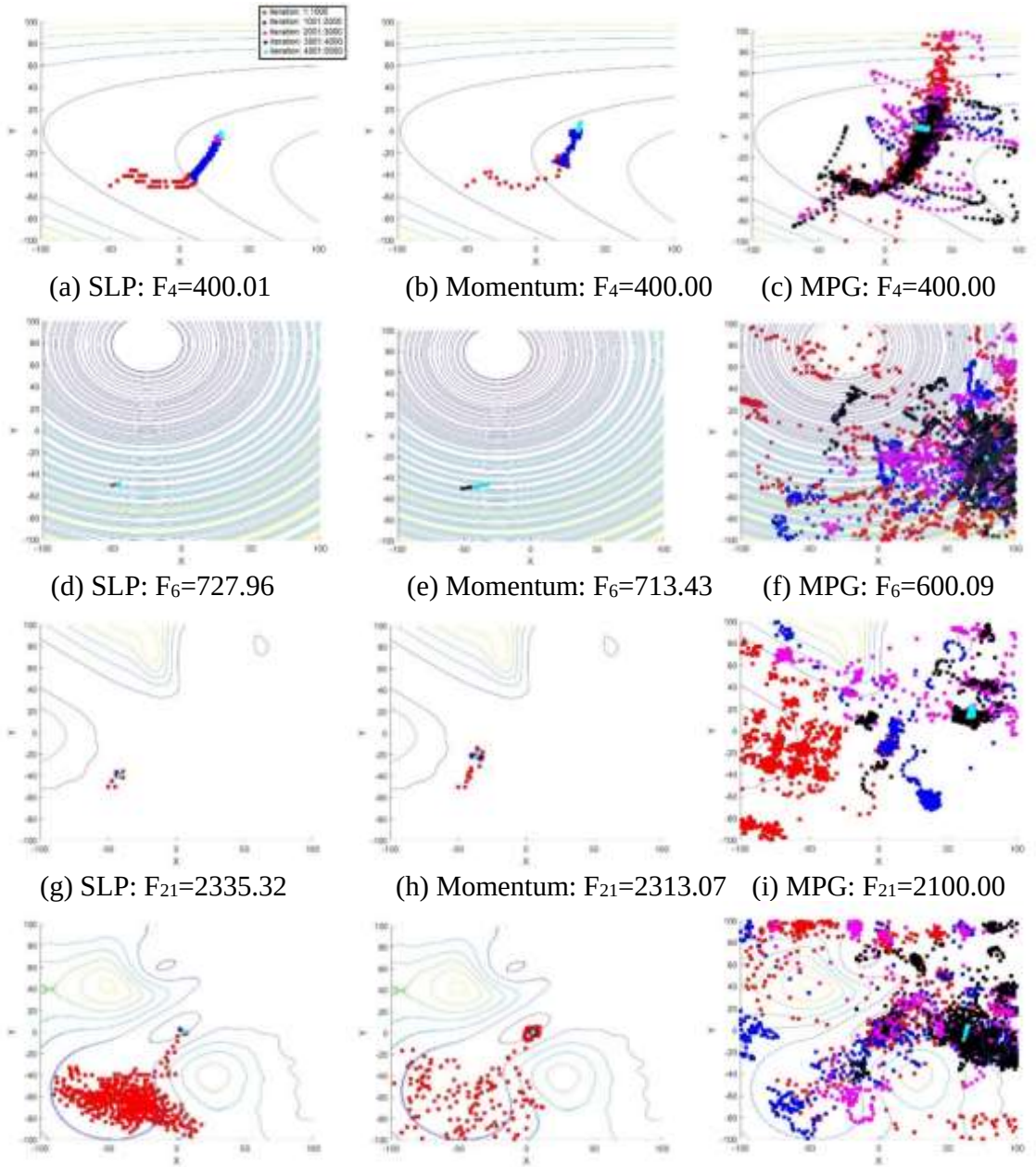
Table 1: Some basic functions of CEC 2017 [43]

Functions	Equations
Bent Cigar	$f(\boldsymbol{x}) = x_1^2 + 10^6 \sum_{i=2}^{n} x_i^2$
Zakharov	$f(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1}^{n} 0.5 i x_i^2)^2 + (\sum_{i=1}^{n} 0.5 i x_i^2)^4$
Rosenbrock	$f(\boldsymbol{x}) = \sum_{i=1}^{n-1} [100\ (x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
Rastrigin	$f(\boldsymbol{x}) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$

Expanded Schaffer	$f(\boldsymbol{x}) = (\sum_{i=1}^{n-1} g(x_i, x_{i+1})) + g(x_n, x_1)$ $g(x,y) = 0.5 + \frac{\sin^2\left(\sqrt{x^2+y^2}\right) - 0.5}{(1+0.001(x^2+y^2))^2}$
Lunacek bi-Rastrigin	$f(\boldsymbol{x}) = \min\left(\begin{cases} \sum_{i=1}^{n}(x_i-\mu_0)^2 \\ d\cdot n + s\sum_{i=1}^{n}(x_i-\mu_1)^2 \end{cases}\right) + 10(n - \sum_{i=1}^{n}\cos(2\pi z_i))$
Non-Continuous Rastrigin	$f(\boldsymbol{x}) = \sum_{i=1}^{n} z_i^2 - 10\cos(2\pi z_i) + 10$
Levy	$f(\boldsymbol{x}) = \sin^2(\pi w_1) + \sum_{i=1}^{n-1}(w_i-1)^2[1+10\sin^2(\pi w_i+1)] + (w_n - 1)^2[1+\sin^2(2\pi w_n)]$ $w_i = 1 + \frac{x_i - 1}{4}$
Modified Schwefel	$f(\boldsymbol{x}) = 418.9829n - \sum_{i=1}^{n} g(z_i)$
High Conditioned Elliptic Function	$f(\boldsymbol{x}) = \sum_{i=1}^{n} (10^6)^{i-1/n-1} x_i^2$
Discus Function	$f(\boldsymbol{x}) = 10^6 x_1^2 + \sum_{i=2}^{n} x_i^2$
Griewank	$f(\boldsymbol{x}) = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right)$

4.1 2D problems

In this section, some of the 2D problems of CEC 2017 are investigated by three gradient-based methods: SLP, momentum and MPG (Fig. 2). It is obvious that the proposed method is capable of leaving local minima and have much better exploration capability. It should be noted that the initial point and the number of function evaluations are chosen the same for all methods.



(a) SLP: F_4=400.01 (b) Momentum: F_4=400.00 (c) MPG: F_4=400.00

(d) SLP: F_6=727.96 (e) Momentum: F_6=713.43 (f) MPG: F_6=600.09

(g) SLP: F_{21}=2335.32 (h) Momentum: F_{21}=2313.07 (i) MPG: F_{21}=2100.00

(j) SLP: F_{24}=2500.01 (k) Momentum: F_{24}=2500.01 (l) MPG: F_{24}=2426.32

Figure 2. Optimization of some 2D functions by SLP, momentum and the proposed method with the same initial point and number of function evaluations

4.2 Other dimensions of CEC 2017

CEC benchmarks are optimized for 10, 30 and 50 variables. The outcomes of the proposed method are compared with the results of several metaheuristic approaches. The number of function evaluations (FE) is chosen as 10000*D*. Where *D* is the number of design variables

and the number of FE is specified by the benchmark paper [43]. Each function is optimized 30 times independently. In each run, 4/5 of the FEs are assigned for exploration and performed in 4 phases. 1/5 of the final iterations are dedicated to exploitation. α is selected as 5 in the first phase and 2 in the other three phases and 0.005 in the final phase. The small and large mutation's probabilities are 10% and 1%, respectively. Each method of "moving toward the best result" is expected to be performed once in every 100 iterations.

The results are presented in Tables 2, 3 and 4 and Fig. 3. As mentioned before, the proposed algorithm is run 30 times and the average results are compared with several well-known metaheuristic methods. The outcomes of other algorithms are achieved from Refs [36, 44-47].

To evaluate the performance of the proposed method, the average results of PSO (1995), DE (1997), ABC (2007), FA (2009), BA (2010), KH (2012), BSA (2013), GWO (2014), CSA (2016), SCA (2016), WOA (2016) and PSOG (2019) are compared with the average results of the proposed method. In 10D problems, the proposed method found better or equal results in 90% of functions in comparison with PSO, FA and PSOG methods. In the mean ranking of 10, 30 and 50-dimensional problems, the proposed method obtained the first rank among all mentioned metaheuristic algorithms. It should also be noted that in some functions, large differences in the optimal solutions show the robustness of the proposed method. Overall, Numerical results from Tables 2-4 and Fig. 3 show that the presented method converges faster than the other methods and achieved better results.

The better performance of the proposed method is due to the novel use of exploration ideas with the help of gradient directions. It is important to note that the presented algorithm is simple and can be easily coded. The necessary parameters are selected the same for all the problems under investigation, so the approach is not sensitive to the predefined parameters. The optimal solution of function F_i is $100i$ which i is the number of CEC function.

Table 2: Comparing the results of the proposed method (MPG) with some metaheuristic approaches (10D)

	PSO	FA	PSOG	MPG	
F_i	[9]	[11]	[36]	Average	Best
1	3.22E8	1.63E9	**613**	1518.81	101
3	1.79E4	3.04E4	**300**	**300**	300
4	547.87	510.21	**400**	**400**	400
5	572.48	561.42	511	**500.47**	500
6	630.32	616.87	**601**	602.18	600.68
7	781.46	792.02	719	**718**	707.58
8	860.55	858.22	810	**800.38**	800
9	1109.9	1184.3	**900**	900.21	900.05
10	3008.0	2628.5	1340	**1259.09**	1006.95
11	1675.9	3540.6	1140	**1101.27**	1100
12	1.86E7	7.71E6	5780	**2508.32**	1673.1
13	4.95E5	1.82E4	8190	**1486.33**	1358.78

14	7642.8	1.04E4	2670	**1462.23**	1432.36
15	4.83E4	1.88E4	1.58E4	**1620.62**	1525.77
16	2120.1	1987.3	1610	**1602.41**	1601.18
17	1865	1803.5	1750	**1723.93**	1709.39
18	7.05E5	2.17E4	1.04E4	**2403.62**	1932.4
19	3.59E4	1.46E4	7210	**1947.28**	1914.71
20	2266.8	2149.8	2050	**2024.58**	2007.03
21	2355.3	2353.2	2300	**2193.33**	2100
22	2668.4	2429.7	2300	**2255.22**	2200
23	2690.6	2665.6	2610	**2584.44**	2300
24	2797.2	2777.8	2690	**2491.18**	2413.22
25	2964.8	2986.8	2930	**2661.15**	2500.04
26	3454.5	3510.2	2890	**2734.06**	2600
27	3152.9	3122.1	3090	**3087.81**	3086.89
28	3345	3476.8	3100	**2955.42**	2800.02
29	3429.9	3329.9	3180	**3142.94**	3128.66
30	4.15E6	2.64E6	3.95E5	**4490.52**	3581.5
Avg. Rank	3.69	3.31	1.83	**1.10**	-
Rank	4	3	2	**1**	-

Bold numbers: the best solution

Table 3: Comparing the results of the proposed method (MPG) with some metaheuristic approaches (30D)

	CSA	GWO	KH	ABC	DE	BSA	SCA	PSOG	MPG	
F_i	[12]	[14]	[16]	[17]	[18]	[19]	[20]	[36]	Average	Best
1	2540	8.87E8	1.72E9	**435**	4334.44	492	1.18E+10	2340	4442.51	126.53
3	475	2.50E4	4.48E4	1.10E5	2.21E4	2.86E4	3.50E4	462	**300.03**	300
4	502	536	505	431	519.42	495	1400	419	**400**	400
5	616	587	644	585	737.79	556	771	563	**508.94**	504.97
6	620	603	639	**600**	652.58	**600**	649	610	616.19	603.85
7	824	822	827	807	962.59	802	1120	810	**790.92**	768.71
8	894	873	905	895	967.25	861	1050	870	**806.75**	802.98
9	1330	1230	3160	2110	7878.78	901	5520	**900**	900.3	900
10	4480	3850	5230	3510	4536.99	5490	8120	3680	**2094.27**	1404.12
11	1250	1330	1620	1750	1184.63	**1150**	2190	1230	1203.79	1141.8
12	1.60E6	2.52E7	4.27E8	9.47E5	3.18E5	1.57E5	1.21E9	3.26E4	**4806.77**	2828.16
13	2.34E4	4.67E6	2.71E8	3.65E4	1.88E4	**9220**	4.07E8	8.43E4	1.73E4	4904.64
14	1580	1.16E5	3.15E5	1.41E5	5502.16	**1470**	1.19E5	7390	1836.12	1571.95
15	4520	1.90E5	1.90E4	9560	2484.69	**1630**	1.56E7	4.36E4	1.56E4	4562.41
16	2380	2350	3190	2210	2827.01	2220	3640	**1860**	1979.66	1642.18
17	1960	1940	2220	1890	2604.53	1820	2420	1870	**1760.27**	1733.27

18	1.74E4	5.33E5	5.26E5	3.39E5	9.42E4	**7840**	2.80E6	8.68E4	2.19E4	6361.6
19	7180	2.42E5	3.33E5	1.78E4	3010.24	**2030**	2.48E7	3.17E4	8360.76	2260.64
20	2330	2330	2540	2250	2864.83	2150	2610	2330	**2134.7**	2032.77
21	2400	2380	2430	2300	2504.78	2360	2560	2360	**2296.86**	2200
22	**2300**	4420	2710	2320	5655.57	2500	8250	**2300**	2722.93	2300
23	2840	2740	3020	2720	3572.97	2710	2990	2720	**2662.12**	2639.74
24	2950	2890	3250	**2710**	3290.70	2890	3160	2860	2737.71	2600
25	2910	2940	2920	2890	2946.71	2890	3200	2890	**2883.49**	2883.39
26	3190	4450	5920	**2900**	6756.37	4030	6870	3590	3021.27	2800.02
27	3360	3230	3460	3210	3998.88	3210	3390	3220	**3178.56**	3158.57
28	3230	3340	3250	3220	3326.26	3390	3780	**3100**	**3100**	3100
29	3860	3590	4230	3530	4115.19	3490	4620	3820	**3434.75**	3286.12
30	1.49E5	3.97E6	1.98E8	2.26E4	**3900.83**	7200	7.44E7	6.47E4	1.22E4	8773.51
Avg. Rank	4.66	5.69	7.24	3.79	6.34	2.79	8.48	3.38	**2.17**	-
Rank	5	6	8	4	7	2	9	3	**1**	-

Table 4: Comparing the results of the proposed method (MPG) with some metaheuristic approaches (50D)

	CSA	BA	GWO	WOA	KH	ABC	DE	PSOG	MPG	
F_i	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[36]	Average	Best
1	3.55E5	5.58E+10	3.63E9	5.27E7	2.41E6	2.40E4	3.67E8	**2430**	3356.58	179.32
3	1.24E4	3.67E5	6.95E4	8.68E4	1.19E5	2.17E5	6.22E4	1.90E4	**301.37**	300.82
4	630.2	1.26E4	798.5	748	581.1	481.6	801.38	463	**402.17**	400
5	745.4	873	673.2	938.8	753.4	713.7	843.26	654	**519.28**	509.95
6	633.3	661.7	608.8	679.9	650.8	**600.1**	655.79	623	626.71	613.9
7	1003	1882	982.9	1687	1065	941.1	1263.04	957	**882.91**	828.64
8	1048	1183	980.6	1219	1069	1023	1175.89	929	**819.73**	812.93
9	3829	1.36E4	4089	2.30E4	1.00E4	1.06E4	2.92E4	**900**	901.32	900.36
10	6972	8828	7183	1.02E4	7840	5936	7289.18	5230	**2895.7**	1983.33
11	1452	22150	2393	1699	6483	4821	**1258.52**	1260	1344.36	1248.6
12	2.85E7	1.67E+10	2.33E8	2.38E8	2.72E9	6.98E6	1.70E7	3.47E5	**1.92E4**	5013.5
13	5.13E4	3.43E9	7.13E7	3.33E5	1.84E9	9.90E4	**1.69E4**	7.59E4	2.82E4	1.46E4
14	8667	7.05E6	5.09E5	8.73E5	3.41E6	1.31E6	1.74E5	2.15E4	**1887.74**	1688.13
15	**1.23E4**	2.95E7	3.89E6	8.79E4	3.92E8	2.77E4	2.70E4	2.69E4	2.06E4	9025.72
16	3065	4796	2833	4811	3753	2854	3176.92	**2000**	2059.99	1806.86
17	2952	4669	2696	3910	3431	2731	3289.62	2520	**2179.44**	1893.77
18	1.03E5	3.69E7	1.94E6	6.96E6	4.62E6	2.14E6	8.72E5	1.13E5	**1.29E4**	5271.47
19	7.47E4	1.34E7	2.30E6	3.27E6	4.52E5	4.85E4	2.04E4	3.13E4	**2166.02**	2015.1
20	2802	3666	2858	3639	3314	2943	3274.33	2620	**2421.83**	2141.51
21	2514	2776	2469	2889	2557	2529	2689.69	2430	**2321.89**	2315.71

22	**3346**	1.04E4	8885	1.16E4	9809	6447	1.08E4	6520	4281.21	2921.21
23	3206	3991	2901	3588	3633	2956	4200.17	2880	**2760.65**	2733.91
24	3290	4191	3094	3668	3888	3412	3682.85	2970	**2807.69**	2700
25	3114	7997	3299	3190	3104	3055	3292.30	2980	**2923.47**	2900
26	4144	1.38E4	5727	1.27E4	9828	3896	7989.49	4710	**3538.49**	2900
27	4132	**3200**	3466	4364	4880	3375	3792.65	3400	3216.2	3181.93
28	3413	3300	3741	3570	3509	3336	3431.57	3280	**3256.6**	3212.16
29	5025	1.15E4	4186	7318	5576	4040	4605.35	4380	**3662.99**	3369.14
30	3.41E7	3.74E8	8.18E7	9.34E7	7.25E7	1.02E6	**5813.17**	4.56E6	9.55E5	6.69E5
Avg. Rank	4.07	8.10	5.00	7.48	6.69	4.07	5.59	2.55	**1.45**	-
Rank	3	9	5	8	7	3	6	2	**1**	-

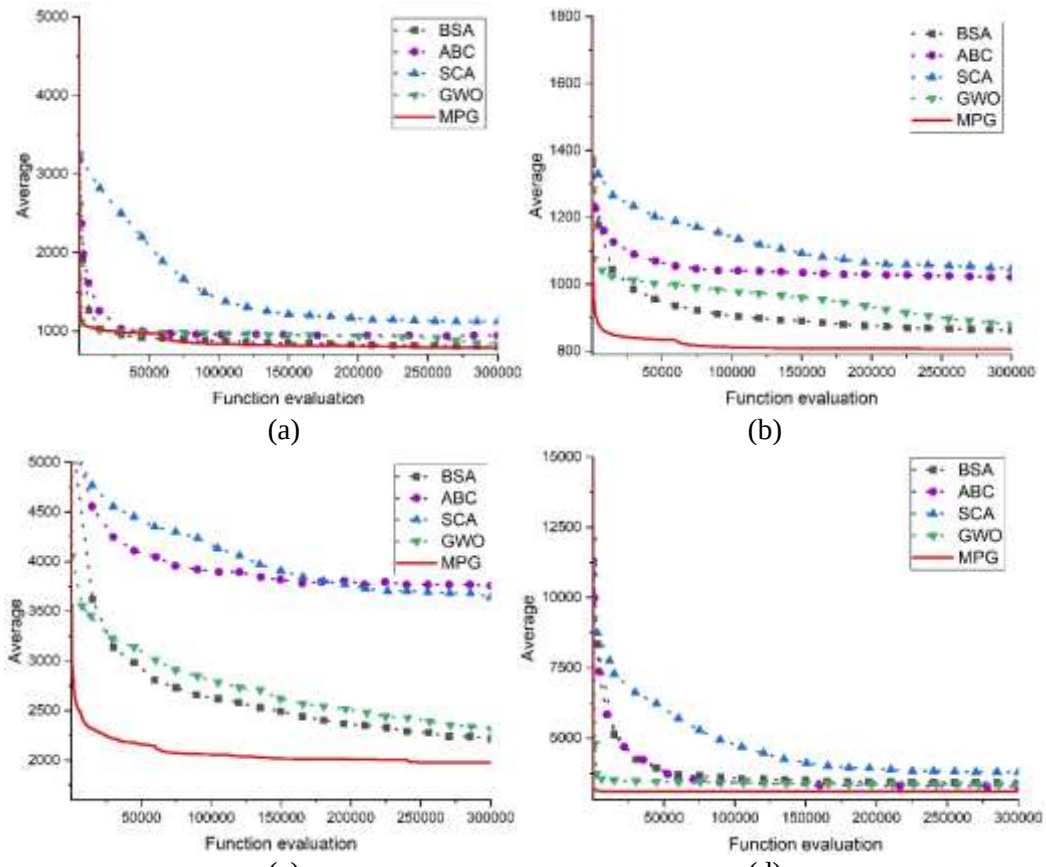


Figure 3. Comparison of iteration histories for (a) F_7 (b) F_8 (c) F_{16} (d) F_{28}

4.3 Structural example

In this example, a truss with 120 members is optimized under frequency constraints (Fig.4).

The cross-section of the members and the coordinates of the nodes are considered as variables. In this shape optimization problem, 7 size variable and 5 node coordinates are optimized for the minimum weight. The constraint is only structural frequency. The first and second frequencies must be greater or equal to 9 and 11 Hz, respectively. The modulus of elasticity of the material is 2.1E + 11 N/m^2 with material density of 7971.81 kg /m^3. Concentrated weights of 3000, 500 and 100 kg are located at points 1, 2:13 and the other nodes, respectively. Optimization is performed by three algorithms of PSO, PSOG and MPG and the results are presented in Table 5 which show that the presented method performs well even in complicated structural optimization problems.

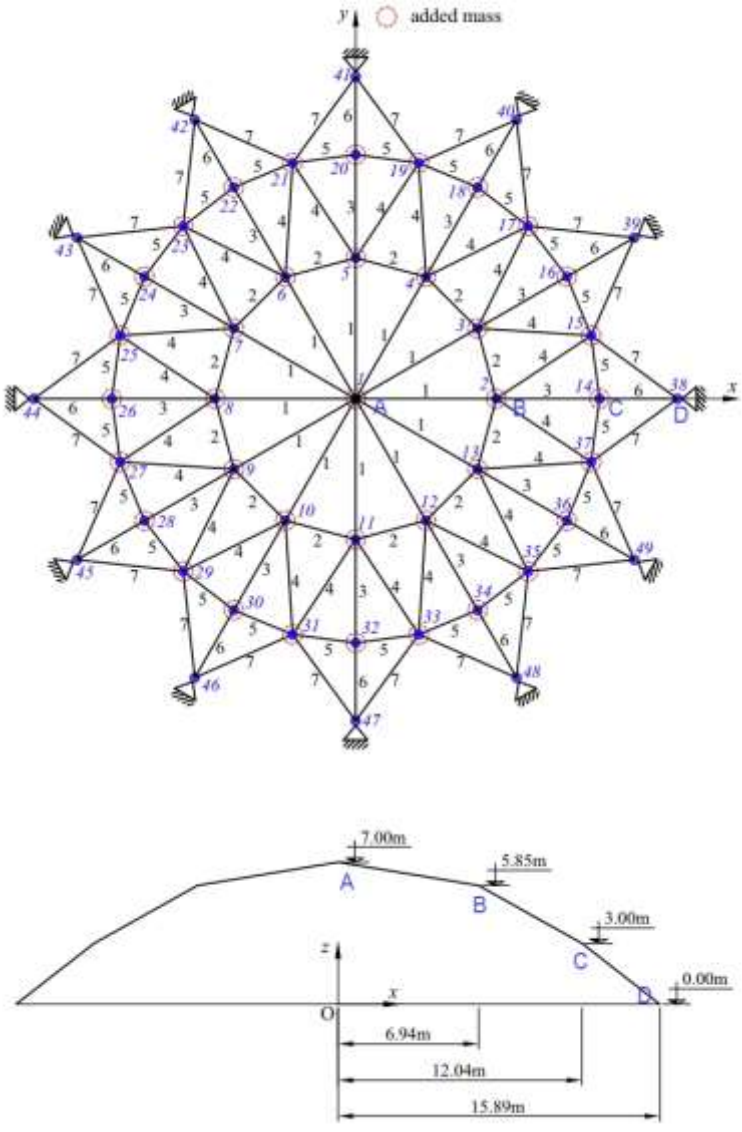


Figure 4. Truss example with 120 elements

Table 5: Weights, cross sections and node positions for structural shape optimization.

Tip	PSO	PSOG	MPG
1	8.536	9.358	9.126
2	13.461	17.13	18.61
3	2.088	6.453	3.647
4	8.868	7.586	6.926
5	9.828	4.946	4.72
6	10.017	6.793	4.415
7	9.466	8.187	6.101
x_2	7.84	9.215	9.926
x_{14}	11.519	12.061	13.084
z_1	9.03	8.392	9.09
z_2	6.795	5.88	6.312
z_{14}	2.545	3.805	3.519
Weight (kg)	4510.512	4169.111	**3828.783**

5. CONCLUSIONS

A multi-strategy approach is presented for the optimization of continuous and differentiable functions with several local optima. The main idea is based on choosing the gradient directions which is the steepest descent direction. To increase the exploration of the method, several techniques are in cooperated. Some basic ideas from metaheuristic methods are employed to explore most of the design media. To enhance the search directions for advanced global search in the defined design space, the ideas of mutation, crossover and fly-back operations are employed. The design points that are directed outside of the space are flown back to the permissible design space. For the search directions, move limits are imposed and the step lengths are gradually reduced as the approach is progressed. At the final stage, around the best optimum, the process is repeated for final better exploitation. Thus a multi-phase gradient technique is outlined for optimization of multimodal functions.

To evaluate the efficiency of the presented approach, the functions presented in CEC 2017 and a structural example are examined. CEC benchmark functions are multimodal that is difficult to find the global optima. The results of the presented method are compared to some of the existing approaches in the literature. The average results and the efficiency ranks are compared and the superiority of the method is observed. The iteration histories of some of the functions are also presented in a graphical manner and compared with other available results. In the structural example under frequency constraints, the proposed method finds a better optimal structure compared to several swarm-based methods.

REFERENCES

1. Haftka RT, Gürdal Z. *Elements of Structural Optimization*, 3rd ed, Springer Science & Business Media, Dordrecht, Germany, 1990.
2. Vanderplaats GN. *Numerical Optimization Techniques for Engineering Design*, 3rd ed., Vanderplaats Research & Development, Inc, Colorado Springs, CO, USA, 1999.
3. Rao SS. *Engineering Optimization: Theory and Practice*, 4th ed, John Wiley & Sons, New Jersey, USA, 2009.
4. Sutskever I, Martens J, Dahl G, Hinton G. On the importance of initialization and momentum in deep learning, in: *Proceedings of Machine Learning Research*, Atlanta, Georgia, USA, 2013, pp. 1139-1147.
5. Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization, *J Mach Learn Res* 2011; **12**: 2121-59.
6. Tieleman T, Hinton G. Divide the gradient by a running average of its recent magnitude, Coursera: Neural Networks for Machine Learning, Technical Report, 2017.
7. Kingma DP, Ba J. Adam: a method for stochastic optimization, Arxiv preprint arxiv: 1412.6980, 2014.
8. Patel V. Kalman-based stochastic gradient method with stop condition and insensitivity to conditioning, *Siam J Optim* 2016; 26: 2620-48. https://doi.org/10.1137/15M1048239.
9. Eberhart R, Kennedy J. Particle swarm optimization, in: *Proceeding of IEEE International Conference Neural Networks (ICNN' 95)*, Perth, Australia, 1995, pp. 1942-1948. https://doi.org/10.1109/ICNN.1995.488968
10. Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search method, *Inform Sci* 2009; **179**: 2232-48. https://doi.org/10.1016/j.ins.2009.03.004
11. Yang XS. Firefly algorithms for multimodal optimization, *in: International Symposium on Stochastic Algorithms*, Springer, 2009, pp. 169-178.
12. Askarzadeh A. A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm, *Comput Struct* 2016; **169**: 1-12. https://doi.org/10.1016/j.compstruc.2016.03.001.
13. Yang XS. A new metaheuristic bat-inspired algorithm, *in: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, Springer, 2010, pp. 65-74.
14. Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer, *Adv Eng Softw* 2014; **69**: 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007.
15. Mirjalili S, Lewis A. The whale optimization algorithm, *Adv Eng Softw* 2016; **95**: 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008.
16. Gandomi AH, Alavi AH. Krill herd: A new bio-inspired optimization algorithm, Commun, Nonline Sci Numer Simul 2012; **17**: 4831-45. https://doi.org/10.1016/j.cnsns.2012.05.010.
17. Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *J Glob Optim* 2007; 39: 459-71. https://doi.org/10.1007/s10898-007-9149-x.

18. Storn R, Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, *J Glob Optim* 1997; **11**: 341-59. https://doi.org/10.1023/A:1008202821328.
19. Civicioglu P. Backtracking search optimization algorithm for numerical optimization problems, *Appl Math Comput* 2013; **219**: 8121-44. https://doi.org/10.1016/j.amc.2013.02.017
20. Mirjalili S. SCA: A sine cosine algorithm for solving optimization problems, *Knowl Based Syst* 2016; **96**: 120-33. https://doi.org/10.1016/j.knosys.2015.12.022.
21. Mitchell M. *An Introduction to Genetic Algorithms*, MIT Press, London, England, 1998.
22. Glover F, Laguna M. *Tabu search, in: Handbook of Combinatorial Optimization*, Springer, 1998, pp. 2093-2229.
23. Gholizadeh S, Barati H. A comprative study of three metaheuristics for optimum design of trusses, *Int J Optim Civil Eng* 2012; **2**: 423-41.
24. Khatibinia M, Naseralavi SS. Truss optimization on shape and sizing with frequency constraints based on orthogonal multi-gravitational search algorithm, *J Sound Vib* 2014; **333**: 6349-69.
25. Kaveh A, Ilchi Ghazaan M. Structural reliability assessment utilizing four metaheuristic algorithms, *Int J Optim Civil Eng* 2015; **5**: 205-25.
26. Kaveh A, Dadras A. Optimal decomposition of finite element meshes via k-median methodology and different metaheuristics, *Int J Optim Civil Eng* 2018; **8**: 227-46.
27. Pakseresht D, Gholizadeh S. Metaheuristic-based sizing and topology optimization and reliability assessment of single-layer lattice domes, *Int J Optim Civil Eng* 2021; **11**: 1-14.
28. Hickernell FJ, Yuan YX. A simple multistart algorithm for global optimization, *Trans* 1997; **1**:1-12.
29. Peri D, Tinti F. A multistart gradient-based algorithm with surrogate model for global optimization, *Commun Appl Ind Math* 2012; **3**: 1-22.
30. Martí R, Lozano JA, Mendiburu A, Hernando L. *Multi-Start Methods*, Handbook of Heuristics, 2016, pp. 1-21.
31. Soterroni AC, Galski RL, Ramos FM. The q-gradient method for continuous global optimization, in: *AIP Conference Proceeding*, AIP, 2013, pp. 2389-2393. https://doi.org/10.1063/1.4826022
32. Yiu KFC, Liu Y, Teo KL. A hybrid descent method for global optimization, *J Glob Optim* 2004; **28**: 229-38. https://doi.org/10.1023/B:JOGO.0000015313.93974.b0
33. Wang YJ, Zhang JS. An efficient algorithm for large scale global optimization of continuous functions, *J Comput Appl Math* 2007; 206: 1015-26. https://doi.org/10.1016/j.cam.2006.09.006
34. Stepanenko S, Engels B. Gradient tabu search, *J Comput Chem* 2007; **28**: 601-11. https://doi.org/10.1002/jcc.20564.
35. Fateen SEK. A. Bonilla-Petriciolet, Gradient-based cuckoo search for global optimization, *Math Probl Eng* 2014; **2014**: 1-13. http://dx.doi.org/10.1155/2014/493740.
36. Salajegheh F, Salajegheh E. PSOG: enhanced particle swarm optimization by a unit vector of first and second order gradient directions, *Swarm Evol Comput* 2019; **46**: 28-51. https://doi.org/10.1016/j.swevo.2019.01.010.

37. Martínez-Estudillo AC, Hervás-Martínez C, Martínez-Estudillo FJ, García-Pedrajas N. Hybridization of evolutionary algorithms and local search by means of a clustering method, *IEEE Trans Syst Man Cybern Syst* 2005; **36**: 534-45. https://doi.org/10.1109/TSMCB.2005.860138
38. Harada K, Ikeda K, Kobayashi S. Hybridization of genetic algorithm and local search in multiobjective function optimization: recommendation of GA then LS, in: *Proceedings of the 8th Genetics Evolutionary Computation Conference*, *ACM*, 2006, pp. 667-674. https://doi.org/10.1145/1143997.1144116
39. Ahandani MA, Vakil-Baghmisheh MT, Talebi M. Hybridizing local search algorithms for global optimization, *Comput Optim Appl* 2014; **59**: 725-48. https://doi.org/10.1007/s10589-014-9652-1.
40. Wu Y, Weise T, Liu W. Hybridizing different local search algorithms with each other and evolutionary computation: better performance on the traveling salesman problem, in: *Genetics Evolutionary Computation Conference* 2016, pp. 1-8. http://dx.doi.org/10.1145/2908961.2909001.
41. Pourchot A, Sigaud O. CEM-RL: Combining evolutionary and gradient-based methods for policy search, in: *Seventh Internatinal Conference Learning Representations*, U.S, 2019, pp. 1-18.
42. Shang Y, Wha B. A global optimization method for neural networks training, in: *IEEE International Conference Neural Networks* 2011, pp. 7-11.
43. Awad NH, Ali MZ, Suganthan PN, Liang JJ, Qu BY. Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization, Technical Report, Nanyang Technological University, Singapore and Jordan University of Science and Technology, Jordan and Zhengzhou University, Zhengzhou China, 2017, pp. 1-34.
44. Dokania S, Chopra A, Ahmad F, Parihar AS. Hierarchy Influenced differential evolution: a motor operation inspired approach, Arxiv preprint arxiv: 1702.05308, 2017.
45. Aydilek IB. A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems, Appl Soft Comput 2018; **66**: 232-49. https://doi.org/10.1016/j.asoc.2018.02.025.
46. Xu Z, Lei Z, Yang L, Li X, Gao S. Negative correlation learning enhanced search behavior in backtracking search optimization, in: *10th International Conference Intelligent Human Machine Systems and Cybernetics, (IHMSC)*, 2018, pp. 310-314. https://doi.org/10.1109/IHMSC.2018.00078
47. Zamani H, Nadimi-Shahraki MH, Gandomi AH. CCSA: conscious neighborhood-based crow search algorithm for solving global optimization problems, *Appl Soft Comput* 2019: 1-45. https://doi.org/10.1016/j.asoc.2019.105583.