G Model
ASOC-1808; No. of Pages 21
ARTICLE IN PRESS

Applied Soft Computing xxx (2012) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

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

Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems

Ali Sadollah[a], Ardeshir Bahreininejad[a,*], Hadi Eskandar[b], Mohd Hamdi[a]

[a] Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia
[b] Faculty of Engineering, Semnan University, Semnan, Iran

ARTICLE INFO

Article history:
Received 15 August 2011
Received in revised form 24 October 2012
Accepted 19 November 2012
Available online xxx

Keywords:
Mine blast algorithm
Metaheuristic
Constrained optimization
Engineering design problems
Constraint handling
Global optimization

ABSTRACT

A novel population-based algorithm based on the mine bomb explosion concept, called the mine blast algorithm (MBA), is applied to the constrained optimization and engineering design problems. A comprehensive comparative study has been carried out to show the performance of the MBA over other recognized optimizers in terms of computational effort (measured as the number of function evaluations) and function value (accuracy). Sixteen constrained benchmark and engineering design problems have been solved and the obtained results were compared with other well-known optimizers. The obtained results demonstrate that, the proposed MBA requires less number of function evaluations and in most cases gives better results compared to other considered algorithms.

1. Introduction

Over the last decades, various algorithms have been used to solve diverse constrained engineering optimization problems. Most of these algorithms are based on numerical linear and nonlinear programming methods that require substantial gradient information and usually seek to improve the solution in the neighborhood of a starting point.

These numerical optimization algorithms provide useful strategies to obtain the global optimum using simple and ideal models. Many real-world engineering optimization problems, however, are very complex in nature and quite difficult to solve. If a problem has more than one local optimum, the result may depend on the selection of an initial point, and the obtained optimal solution may not necessarily be the global optimum.

Furthermore, the gradient search may become difficult and unstable when the objective function and the constraints have multiple or sharp peaks. The computational drawbacks of existing numerical methods have forced researchers to rely on metaheuristic algorithms based on the simulations and mimicking the behavior of natural phenomena to solve complex engineering optimization problems.

The common factor in metaheuristic algorithms is that they combine rules and randomness to imitate natural phenomena [1]. These phenomena include the biological evolutionary process such as genetic algorithms (GAs) proposed by Holland [2] and Goldberg [3], swarm behavior such as particle swarm optimization (PSO) proposed by Kennedy and Eberhart [4], and the physical annealing which is generally known as simulated annealing (SA) proposed by Kirkpatrick et al. [5].

Among optimization methods, evolutionary algorithms (EAs) which are generally known as general purpose optimization algorithms are capable of finding the near-optimal solution to the numerical real-valued test problems. EAs have been successfully applied to constrained optimization problems [6].

GAs are based on the genetic process of biological organisms [2,3]. Over many generations, natural populations evolve according to the principles of natural selection (i.e., survival of the fittest). The efficiency of the different architectures of evolutionary algorithms in comparison to other heuristic techniques has been tested in both generic [7–9] and engineering design [10] problems.

Recently, Chootinan and Chen [11] proposed a constraint-handling technique by taking a gradient-based repair method. The proposed technique is embedded into GAs as a special operator. Tang et al. [12] proposed the improved genetic algorithm (IGA) based on a novel selection strategy to handle nonlinear programming problems. Accordingly, Yuan and Qian [13] developed a new hybrid genetic algorithm (HGA) to solve

* Corresponding author. Tel.: +60 126283265; fax: +60 379675330.
E-mail address: bahreininejad@um.edu.my (A. Bahreininejad).

1568-4946/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.11.026

twice continuously differentiable nonlinear programming (NLP) problems.

The HGA combines the genetic algorithm with local solver differently from some hybrid genetic algorithms. Amirjanov [14] investigated an approach that adaptively shifts and shrinks the size of the search space of the feasible region which is called changing range genetic algorithm (CRGA).

PSO is a recently developed heuristic technique, inspired by choreography of a bird flock developed by Kennedy and Eberhart [4]. The approach can be viewed as a distributed behavioral algorithm that performs a multidimensional search. It makes use of a velocity vector to update the current position of each particle in the swarm.

In Ref. [15], there are some suggestions for choosing the parameters used in PSO. He and Wang [16] proposed an effective co-evolutionary PSO (CPSO) for constrained problems, where PSO was applied to evolve both decision and penalty factors. Recently, Gomes applied PSO on truss optimization using dynamic constraints [17]. Coelho [18] studied quantum-behaved PSO (QPSO) which is derived using mutation operator with Gaussian probability distribution. He and Wang [19] developed a new hybrid PSO (HPSO) with a feasibility-based rule to solve constrained optimization problems.

Recently, several hybrid optimization methods have been proposed for constrained optimization problems [20–22]. In addition, differential evolution (DE) algorithm which is a scheme for generating trial parameter vectors has been widely used for constrained optimization problems [23,24]. Various other optimization methods have been developed for solving complex and real-life problems, particularly for solving engineering constrained problems [25].

Examples of such methods include teaching-learning-based optimization (TLBO) [26–28] which is based on the influence of a teacher on learners, the harmony search (HS) [1] algorithm which is conceptualized using the musical process of searching for a perfect state of harmony, and the society and civilization (SC) [29] which is inspired from intra and intersociety interactions within a formal society and the civilization model to solve constrained optimization problems. These algorithms have been applied to numerous engineering optimization problems and have shown the efficiencies in solving some specific kinds of problem.

This paper introduces a new metaheuristic algorithm, so called the mine blast algorithm (MBA) which the concepts are inspired form the explosion of mine bombs in real life situations. Recently, sizing optimization of truss structures with discrete variables was solved using the MBA [30]. In this paper, the proposed method is applied for constrained and engineering design problems with discrete and continues variables.

The remaining of this paper is organized as follows: Section 2 presents the concepts of the proposed method in details. Guidelines for selecting the related MBA parameters and their effects are also given in Section 2. In Section 3, the performance of MBA is tested on different constrained optimization and engineering design problems and the results are compared with other well-known optimizers in terms of number of function evaluations (computational cost) and function value (accuracy). Finally, conclusions are given in Section 4.

2. Mine blast algorithm

2.1. Basic concepts

The idea of the proposed algorithm is based on the observation of a mine bomb explosion, in which the thrown pieces of shrapnel collide with other mine bombs near the explosion area resulting in their explosion. To understand this situation, consider a mine field where the aim is to clear the mines. Hence, the goal is to find the mines, while importantly to find the one with the most explosive effect located at the optimal point X^* which can cause the most casualties (min or max $f(x)$ per X^*).

The mine bombs of different sizes and explosive powers are planted under the ground. When a mine bomb is exploded, it spreads many shrapnel pieces and the casualties ($f(x)$) caused by each piece of shrapnel are calculated. A high value for casualties per piece of shrapnel in an area may indicate the existence of other mines which may or may not have higher explosive power.

Each shrapnel piece has definite directions and distances to collide with other mine bombs which may lead to the explosion of other mines due to the collision. The collision of shrapnel pieces with other mines may lead us to discover the most explosive mine. The casualties caused by the explosion of a mine bomb are considered as the fitness of the objective function at the mine bomb's location. The domain (mine field) solution may be divided into infinite grid where there is one mine bomb in each portion of the grid.

2.2. Proposed method

The proposed MBA starts with an initial point(s) called the first shot point(s). The first shot point is represented by X_0^f. The superscript f refers to the number of first shot point(s) ($f = 1, 2, 3, \ldots$), where f can be a user defined parameter. However, the proposed algorithm can also randomly choose the location(s) of the first shot point(s) using the lower and upper bound values of a problem. This algorithm requires an initial population of individuals as is the case with several other metaheuristic methods. This population is generated by a first shot explosion producing a number of individuals (shrapnel pieces). The number of initial population (N_{pop}) is considered as the number of shrapnel pieces (N_s).

The algorithm uses the lower and upper bound values specified by a problem. It then creates the first shot point value by a small randomly generated value given as:

$$X_0 = LB + rand \times (UB - LB) \quad (1)$$

where X_0, LB and UB are the generated first shot point, lower and upper bounds of the problem, respectively. $rand$ is a uniformly distributed random number between 0 and 1. Although in all optimization simulations conducted in this research one first shot point was used efficiently, however, more than one first shot points may be used which results in an increase in the initial population and subsequently results in an increase in the number of function evaluations (computational cost).

Suppose that X is the current location of a mine bomb given as:

$$X = \{X_m\}, \quad m = 1, 2, 3, \ldots, N_d \quad (2)$$

in which N_d is the search space dimension equal to the number of independent variables. Consider that N_s shrapnel pieces are produced by the mine bomb explosion causing another mine to explode at X_{n+1} location:

$$X_{n+1}^f = X_{e(n+1)}^f + \exp\left(-\sqrt{\frac{m_{n+1}^f}{d_{n+1}^f}}\right) X_n^f, \quad n = 0, 1, 2, 3, \ldots \quad (3)$$

where $X_{e(n+1)}^f$, d_{n+1}^f and m_{n+1}^f are the location of exploding mine bomb collided by shrapnel, the distance and the direction (slope) of the thrown shrapnel pieces in each iteration, respectively. The location of exploding mine bomb $X_{e(n+1)}^f$ is defined as:

$$X_{e(n+1)}^f = d_n^f \times rand \times \cos(\theta), \quad n = 0, 1, 2, \ldots \quad (4)$$

where *rand* is a uniformly distributed random number and θ is the angle of the shrapnel pieces which is a constant value and is calculated using $\theta = 360/N_s$.

The concept of Eq. (4) is to simulate the explosion of mine bombs in the real world. Each shrapnel pieces (individual), having variable distances from the point of explosion and definite directions, explore the domain space in 360° at each iteration specified by θ and d_n^f in order to find the best optimal point. The value of θ is set to $360/N_s$ in order to conduct uniform search in the domain space. This can prevent the accumulation of individuals in a specific region of the domain search.

The exponential term in Eq. (3) is used to improve the obtained blast point by influencing the information from previous solutions (X_n^f). The distance d_{n+1}^f and the direction of shrapnel pieces m_{n+1}^f are defined as:

$$d_{n+1}^f = \sqrt{(X_{n+1}^f - X_n^f)^2 + (F_{n+1}^f - F_n^f)^2}, \quad n = 0, 1, 2, 3, \ldots \tag{5}$$

$$m_{n+1}^f = \frac{F_{n+1}^f - F_n^f}{X_{n+1}^f - X_n^f}, \quad n = 0, 1, 2, 3, \ldots \tag{6}$$

where F is the function value for the X. To calculate the initial distance for each shrapnel pieces $d_0 = (UB - LB)$ in each dimensions is used. The initial distance given by the proposed algorithm is used to search the best solution within a range ($LB < d_0 < UB$) that is computed by the product of the initial distance and a randomly generated number (for example *rand* function in MATLAB software).

Furthermore, in order to conduct exploration of the design space at smaller and larger distances, the exploration factor (μ) is introduced. This constant, which is used in the early iterations of the algorithm, is compared with an iteration number index (k), and if it is higher than k, then the exploration process begins. The equations for the exploration of the solution space are given as:

$$d_{n+1}^f = d_n^f \times (|randn|)^2, \quad n = 0, 1, 2, \ldots \tag{7}$$

$$X_{e(n+1)}^f = d_{n+1}^f \times \cos(\theta), \quad n = 0, 1, 2, \ldots \tag{8}$$

where *randn* is normally distributed pseudorandom number (obtained using *randn* function in MATLAB). During the exploration process when the μ is applied, the distance of each shrapnel pieces is modified using Eq. (7). The square of a normally distributed random number offers the advantage of search ability at smaller and larger distances giving better exploration in early iterations. Therefore, the exploration process shifts the shrapnel pieces closer to the optimum point in early iterations. A higher value for the exploration factor (μ) makes it possible to explore more remote regions (better exploration), thus, the value of μ determines the intensity of the exploration.

To increase the global search ability of the proposed method, initial distance of shrapnel pieces are reduced gradually to allow the mine bombs search the probable global minimum location. The reduction in d_0^f is given as:

$$d_n^f = \frac{d_{n-1}^f}{\exp(k/\alpha)}, \quad n = 1, 2, 3, \ldots \tag{9}$$

where α and k are reduction constants and iteration number index, respectively. The choice of α which is user parameter depends on the complexity of the problem. The effect of α is to reduce the distance of each shrapnel pieces adaptively using Eq. (9). Thus the whole interval from lower bound to upper bound for a given problem is searched. At the final iteration, the value of distance of shrapnel will be approximately equal to zero (ε = 2.2E−16 in MATLAB). The schematic diagram of the algorithm representing two aspects of the MBA (exploration in color lines and exploitation in black color lines) is shown in Fig. 1.

Based on Fig. 1, there are two processes for searching the solution domain in order to find the global optimum solution, the exploration and exploitation processes. The difference between these two processes is how they influence the whole search process toward the optimal solution. More specifically, the exploration factor describes the exploration process (color lines in Fig. 1). The exploration factor (μ) represents the number of first iterations. Hence, if μ is set to 10, then for 10 iterations the algorithm uses Eqs. (7) and (8) for calculating the distance of shrapnel pieces and the location of the exploded mine bomb, respectively.

On the other hand, for the exploitation process (black lines in Fig. 1), the algorithm is encouraged to focus on the optimal point. In particular, with respect to the exploitation process, the proposed algorithm converges to the global optimum solution using Eqs. (4)–(6) to calculate the location of exploded mine bomb, distance and the direction of shrapnel pieces, respectively. The distance of shrapnel pieces is reduced adaptively using Eq. (9) in exploitation process (i.e., as it converges to the optimal solution). The pseudocode for the exploration and exploitation processes is as follows:

if $\mu > k$
Exploration (Eqs. (7) and (8))
else
Exploitation (Eqs. (4)–(6), and (9))
end

where k is the iteration number index.

2.3. Setting the user parameters

Wrong choice of values for algorithm parameters may result in a low convergence rate, convergence to a local minimum, or undesired solutions. Hence, the following guidelines to fine tune the parameters are offered:

- For simple to medium complexity optimization problems, a choice of 10–15 pieces of shrapnel per mine bomb may be sufficient. For more complex problems, higher values for the number of shrapnel pieces (N_s) is recommended for more mine explosions in the field which allows better search for the design space. For highly complex problems N_s may be chosen as 50. On the other hand, increasing the number of shrapnel pieces increases the computation time in addition to an increase in the number of function evaluations. In other word, the number of shrapnel pieces is equal to the number of population ($N_s = N_{pop}$).
- Exploration factor (μ) highly depends on the complexity of the problem, the number of independent variables and constraints, in addition to the interval span. Usually, for less than four design variables and moderately complex functions, the value of μ may be taken as zero. Increasing μ may lead the possibility of getting trapped in local minima.
- Reduction constant (α) also depends on the complexity of the problem, number of decision variables, and interval span. When the interval span (LB and UB) is large, large value for α should be chosen for more exploration. That means if we have interval span [−100,100], then α = 100 may not be a good choice and instead α = 1000 may be better choice. A large value for α increases the probability of finding global minimum but leads to increase in computational time.

2.4. Effects of algorithm parameters

The choice and tuning of the initial parameters are highly important attributes for most metaheuristic algorithms. As explained in Section 2.3, the wrong choice of user parameters may lead to unsatisfactory results, high computational costs and getting trapped in local minima.

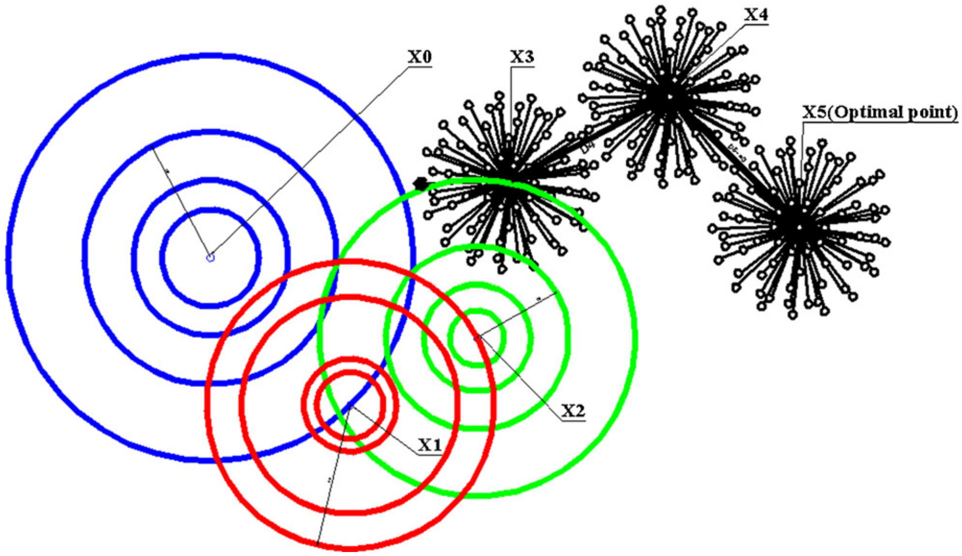

Fig. 1. Schematic view of the mine blast algorithm including of exploration (color lines) and exploitation (black lines) processes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

In order to further clarify the setting of the initial parameters for the proposed method, three cases are provided. In each case, different results using diverse user parameters are shown. Case 3 represents the best used initial parameters. The number of population (N_{pop}) which is considered as the number of shrapnel pieces (N_s) and number of iterations are assumed constant parameters.

The number of shrapnel pieces ($N_s = N_{pop}$) was chosen 50 for pressure vessel and speed reducer design problems. Similarly, the number of iterations was considered 2000 and 1000 iterations for pressure vessel and speed reducer design problems, respectively. These two test functions are given in Appendix B.

The exploration and reduction constants (μ and α) were varied to see the effects of changing the user parameters for statistical results. Table 1 demonstrates the effect of choosing initial parameters for the two considered design problems. The task of optimization was run for 30 independent runs for both problems.

As can be seen from Table 1 and described in Section 2.3, Case 3 offers superior results compared with other cases in terms of statistical results for both design problems. Using exploration factor (μ) gives the algorithm more freedom to search wider range resulting in detection of better solutions.

Accordingly, as the name of reduction constant (α) represents, it divides the distance of each shrapnel pieces to α interval distances and enables searching within the reduced intervals in each iteration. The simple concept behind α is that searching in smaller distance is carried out faster than searching in a large space. Hence, higher values of α results in increasing the probability of finding global optimum solution.

2.5. Constraint handling

In the current work, a modified feasible-based mechanism is used to handle the problem specific constraints which consist of the following four rules [31]:

- Rule 1: Any feasible solution is preferred to any infeasible solution.
- Rule 2: Infeasible solutions containing slight violation of the constraints (from 0.01 in the first iteration to 0.001 in the last iteration) are considered as feasible solutions.
- Rule 3: Between two feasible solutions, the one having the better objective function value is preferred.
- Rule 4: Between two infeasible solutions, the one having the smaller sum of constraint violation is preferred.

By using the first and fourth rules, the search is oriented to the feasible region rather than to the infeasible region, and by employing the third rule the search is directed to the feasible region with good solutions [31]. For most structural optimization problems, the global minimum locates on or close to the boundary of a feasible design space. By applying rule 2, the shrapnel pieces approach the boundaries and can reach the global minimum with a higher probability [32].

Fig. 2 shows the constraint handling approach by the MBA. As can be seen from Fig. 2, in the search space, shrapnel pieces may violate either the problem specific constraints or the limits of the design variables. In this case, the distance of infeasible shrapnel piece (e.g. X_3 in Fig. 2) is reduced adaptively using Eq. (9) whereas that violated shrapnel piece is also placed in the feasible region.

2.6. Convergence criteria

For termination criteria, as commonly considered in metaheuristic algorithms, the best result is calculated where the termination condition may be assumed as the maximum number of iterations, CPU time, or ε which is a small value and is defined as an allowable tolerance between the last two results. The MBA proceeds until the above convergence criteria are satisfied.

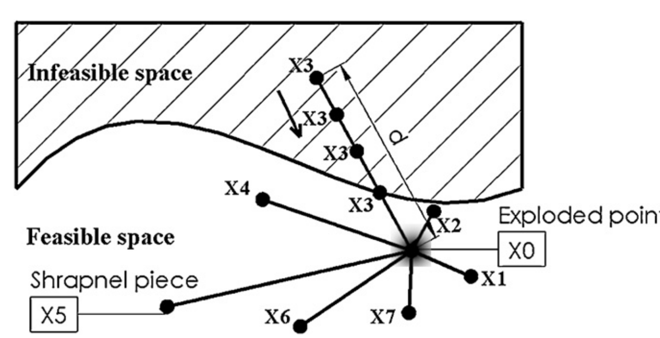

Fig. 2. Schematic view of constraint handling approach using the proposed method.

Table 1
Effects of the algorithm parameters for two design problems.

Results	Pressure vessel problem			Speed reducer problem		
	Case 1 $\mu = 0$ $\alpha = 5000$	Case 2 $\mu = 5$ $\alpha = 25{,}000$	Case 3 $\mu = 10$ $\alpha = 50{,}000$	Case 1 $\mu = 0$ $\alpha = 100$	Case 2 $\mu = 5$ $\alpha = 300$	Case 3 $\mu = 10$ $\alpha = 500$
Best	5935.8659	5949.6481	5889.3216	3000.5738	2997.3158	2994.4824
Mean	6278.2867	6110.8637	6200.6476	3009.2497	3000.8920	2996.6524
Worst	6876.8879	6481.8895	6392.5062	3020.1824	3005.5041	2999.6524
SD	313.56	177.3	160.34	5.29	3.25	1.56

2.7. The steps of MBA

The steps of MBA are summarized as follows:

Step 1: Choose the initial parameters of MBA: N_s, μ, α, and maximum number of iterations.
Step 2: Check the condition of exploration factor (μ).
Step 3: If condition of exploration factor is satisfied, calculate the distance of shrapnel pieces and their locations according to Eqs. (7) and (8), respectively. Otherwise, go to Step 10.
Step 4: Calculate the direction of shrapnel pieces according to Eq. (6).
Step 5: Generate the shrapnel pieces and compute their improved locations using Eq. (3).
Step 6: Check the constraints for generated shrapnel pieces.
Step 7: Save the best shrapnel piece as the best temporal solution.
Step 8: Does the shrapnel piece have the lower function value than the best temporal solution?
Step 9: If true, exchange the position of the shrapnel piece with the best temporal solution. Otherwise, go to Step 10.
Step 10: Calculate the distance of shrapnel pieces and their locations using Eqs. (4) and (5) and return to Step 4.
Step 11: Reduce the distance of the shrapnel pieces adaptively using Eq. (9).
Step 12: Check the convergence criteria. If the stopping criterion is satisfied, the algorithm will be stopped. Otherwise, return to Step 2.

3. Experimental studies

In this section, the performance of MBA is examined by solving several constrained optimization problems. In order to validate the proposed method for constraint problems, first, eight constrained benchmark problems (see Appendix A) have been examined and then, the resulting performance of MBA against eight engineering benchmark design problems (see Appendix B) that are widely used in literatures were tested and the results have been compared with other well-known optimizers. Obtained optimization results were compared in terms of statistical results and NFEs.

The number of function evaluations (NFEs) (computational cost) which is considered as the best NFEs corresponding to the obtained best solution in this paper, is calculated by the product of the number of shrapnel pieces and the number of iterations (i.e. NFEs = $N_s \times$ Iteration number). In other words, the NFEs considered in this paper is the one found for the best optimal solution.

The reported benchmark problems include objective functions and constraints of various types and natures (quadratic, cubic, polynomial and nonlinear) with various number of design variables, and number of inequality and equality constraints.

The proposed algorithm was coded in MATLAB programming software and the simulations and numerical solutions were run on a Pentium V 2.53 GHz with 4 GB RAM. The task of optimizing each constrained benchmark and mechanical design problems was executed using 100 independent runs. Relatively simple constraint functions to complex and nonlinear programming (NLPs) problems were solved in this research.

Table 2
User parameters used for MBA for sixteen constrained and engineering problems.

Problem	N_s	α	μ	Max iteration
Problem 1	20	500	0	500
Problem 2	15	20,000	0	1000
Problem 3	30	1000	0	300
Problem 4	25	300	0	100
Problem 5	15	1000	0	300
Problem 6	50	100,000	10	2500
Problem 7	50	20,000	5	1000
Problem 8	50	5000	0	500
Three-bar truss	40	5000	0	500
Pressure vessel	50	50,000	10	2000
Spring design	50	5000	0	1000
Welded beam	30	150,000	5	2000
Speed reducer	50	500	10	500
Gear train	20	1000	0	500
Belleville spring	50	100,000	10	300
Rolling element bearing	50	5000	10	1000

The maximization problems were transformed into minimization ones using $-f(x)$. All equality constraints were converted into inequality ones, $|h(x)| - \delta \le 0$ using the degree of violation $\delta = 2.2\text{E}{-16}$ that was taken from MATLAB software. The user parameters of MBA which were used for benchmark constrained functions and engineering problems are presented in Table 2.

3.1. Constrained problem 1

This minimization problem, originally introduced by Braken and McCormick [33], is a relatively simple constrained minimization problem. The optimum solution is obtained at $X^* = (0.82288, 0.91144)$ with an objective function value equal to $f(X^*) = 1.393454$. Homaifar and his colleagues [34] solved this problem using the GA.

Fogel [35] compared the result of evolutionary programming (EP) with results obtained using GA. Lee and Geem [1] applied harmony search (HS) method and found their best solution after approximately 40,000 searches. By applying the MBA, the best solution obtained at $X^* = (0.822875, 0.911437)$ with an objective function value equal to $f(X^*) = 1.3934649$ only after 2140 function evaluations. The standard deviation (SD) of the proposed method for this problem is equal to zero, which indicates that the worst, mean, and best solutions are the same.

Table 3 represents the comparison of the best solution and the corresponding design variables among different optimizers. As can

Table 3
Comparison of results obtained from different methods for constrained problem 1.

Method	X_1	X_2	$h(X)$	$g(X)$	$f(X)$
EP	0.8350	0.9125	1.0E−02	−7.0E−02	1.3772
GA	0.8080	0.8854	3.7E−02	5.2E−02	1.4339
HS	0.8343	0.9121	5E−03	5.4E−03	1.3770
MBA	0.822875	0.911437	1.11E−16	0	1.3934649
Optimal	0.82288	0.91144	7.05E−09	1.73E−08	1.393454

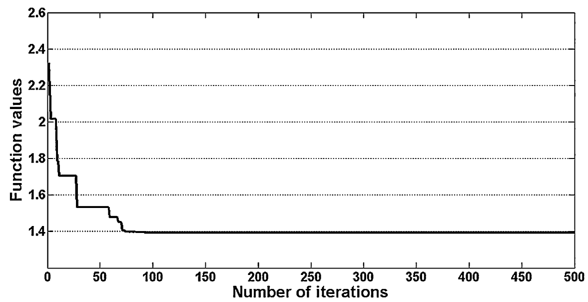

Fig. 3. Function values versus number of iterations for the constrained problem 1.

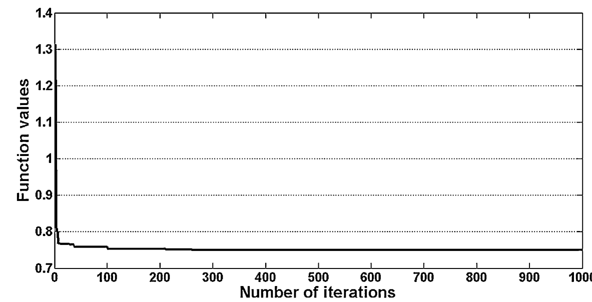

Fig. 4. Function values versus number of iterations for the constrained problem 2.

Table 4
Comparison of best solution for constrained problem 2.

Method	X_1	X_2	$h(X)$	$f(X)$
CULDE	−0.707036	0.500000	1.94E−04	0.749900
MBA	−0.706958	0.499790	8.82E−15	0.750000
Optimal	−0.70711	0.5	0	0.750000

be seen from Table 3, the optimization results obtained by MBA is very close to the optimal solution and MBA outperformed other considered optimizers in terms of the objective function value, constraint accuracy, and NFEs. Fig. 3 shows the function values versus the number of iterations for the constrained problem 1.

3.2. Constrained problem 2

For this minimization problem, MBA is compared with sixteen optimizers: CRGA [14], self adaptive penalty function (SAPF) [36], cultured differential evolution (CULDE) [37], simple multi-membered evolution strategy (SMES) [38], homomorphous mappings (HM) [39], adaptive segregational constraint handling evolutionary algorithm (ASCHEA) [40], PSO, particle swarm optimization with differential evolution (PSO-DE) [41], stochastic ranking (SR) [42], differential evolution (DE) [43], differential evolution with level comparison (DELC) [44], differential evolution with dynamic stochastic selection (DEDS) [45], hybrid evolutionary algorithm and adaptive constraint handling technique (HEAA) [46], improved stochastic ranking (ISR) [47], α constraint simplex method (α Simplex) [48], and artificial bee colony (ABC) [49].

Table 4 represents the comparisons between optimal solutions and related design variables for two methods. The comparison of statistical results for constrained problem 2 is given in Table 5. From Table 5, MBA reached the optimal solution as most of the algorithms considered in this paper. However, except for CRGA, MBA offered better results compared to all considered optimizers in terms of NFEs as shown in Table 5.

Table 5
Comparison of statistical results obtained using different algorithms for constrained problem 2. "N.A" means not available.

Method	Worst	Mean	Best	SD	NFEs
HM	0.75	0.75	0.75	N.A	1,400,000
ASCHEA	N.A	0.75	0.75	N.A	1,500,000
CRGA	0.757	0.752	0.750	2.5E−03	3000
SAPF	0.757	0.751	0.749	2E−03	500,000
CULDE	0.796455	0.757995	0.749900	1.71E−02	100,100
SMES	0.75	0.75	0.75	1.52E−04	75,000
PSO	0.998823	0.860530	0.750000	8.4E−02	70,100
PSO-DE	0.750001	0.749999	0.749999	2.5E−07	70,100
DE	0.74900	0.74900	0.74900	N.A	30,000
SR	0.750	0.750	0.750	8E−05	350,000
DELC	0.750	0.750	0.750	0	50,000
DEDS	0.7499	0.7499	0.7499	0	225,000
HEAA	0.750	0.750	0.750	3.4E−16	200,000
ISR	0.750	0.750	0.750	1.1E−16	137,200
α Simplex	0.7499	0.7499	0.7499	4.9E−16	308,125
ABC	0.75	0.75	0.75	0	240,000
MBA	0.750011	0.750003	0.750000	3.29E−06	6405

Fig. 4 depicts the function values with respect to the number of iterations for the constrained problem 2. By observing Fig. 4, MBA reached the near optimal solution in the early iterations of the algorithm. This advantage is seen in other constrained problems and may be considered as a superiority factor of the proposed method.

3.3. Constrained problem 3

The constrained minimum solution is located at $X^* = (2.246826, 2.381865)$ with an objective function value equal to $f(X^*) = 13.59085$. The MBA was applied and found its optimal solution at $X^* = (2.24683, 2.381997)$ with a corresponding function value $f(X^*) = 13.590842$ using 4560 function evaluations. The HS reached the same optimal value after 15,000 searches. The statistical results (worst, mean, best solution, and SD) are given as 13.592670, 13.591027, 13.590842, and 3.64E−04, respectively.

The MBA's best solution was compared to the previous solutions reported by Deb [50] and HS, as shown in Table 6. Deb solved this problem using the hybrid GA-based method with tournament selection (TS-R method) and with Powell and Skolnick's constraint handling method (PS method), and obtained the best solution of 13.59085 using TS-R method, which showed an excellent agreement with the optimal solution. The MBA superiority on HS is in terms of NFEs, function value, and constrained accuracy. Fig. 5 represents the function values versus the number of iterations for the constrained problem 3.

3.4. Constrained problem 4

This maximization problem was previously solved using HM, SR, ASCHEA, cultural algorithms with evolutionary programming (CAEP) [51], HPSO, hybrid nelder-mead simplex search and particle swarm optimization (NM-PSO) [52], PSO-DE, PSO, GA [11], SMES, CRGA, SAPF, CULDE, DE, DELC, DEDS, ISR, HEAA, ABC, and α Simplex.

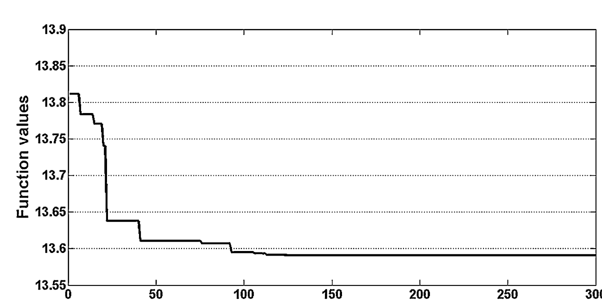

Fig. 5. Function values versus number of iterations for the constrained problem 3.

Table 6
Comparison of optimization results obtained from previous studies for constrained problem 3. "N.A" stands for not available.

D.V	GA with PS (R = 0.01)	GA with PS (R = 1)	GA with TS	HS	MBA	Optimal solution
X_1	N.A	N.A	2.246826	2.246840	2.246833	2.246826
X_2	N.A	N.A	2.381865	2.382136	2.381997	2.381865
$g_1(X)$	N.A	N.A	N.A	−2.09E−06	0	3.52E−07
$g_2(X)$	N.A	N.A	N.A	−0.222181	−0.222183	−0.2221829
$f(X)$	13.58958	13.59108	13.59085	13.590845	13.590842	13.59085

Table 7
Comparison of statistical results obtained from MBA and previous studies for constrained problem 4.

Method	Worst	Mean	Best	SD	NFEs
HM	−0.0291438	−0.0891568	−0.0958250	N.A	1,400,000
ASCHEA	N.A	−0.095825	−0.095825	N.A	1,500,000
PSO	−0.02914408	−0.09449230	−0.09582594	9.4E−03	10,600
PSO-DE	−0.0958259	−0.0958259	−0.0958259	1.3E−12	10,600
SR	−0.0958250	−0.0958250	−0.0958250	2.6E−17	76,200
CAEP	−0.0958250	−0.0958250	−0.0958250	0	50,020
DE	−0.0958250	−0.0958250	−0.0958250	N.A	10,000
HPSO	−0.0958250	−0.0958250	−0.0958250	1.2E−10	81,000
NM-PSO	−0.0958250	−0.0958250	−0.0958250	3.5E−08	2103
CRGA	−0.095808	−0.095819	−0.095825	4.40E−06	64,900
SAPF	−0.092697	−0.095635	−0.095825	1.055E−03	500,000
GA	−0.0958250	−0.0958250	−0.0958250	2.70E−09	4486
SMES	−0.095825	−0.095825	−0.095825	0	240,000
CULDE	−0.095825	−0.095825	−0.095825	1E−07	100,100
DELC	−0.095825	−0.095825	−0.095825	1.0E−17	5000
DEDS	−0.095825	−0.095825	−0.095825	4.0E−17	225,000
HEAA	−0.095825	−0.095825	−0.095825	2.8E−17	200,000
ISR	−0.095825	−0.095825	−0.095825	2.7E−17	160,000
α Simplex	−0.095825	−0.095825	−0.095825	3.8E−13	306,248
ABC	−0.0958250	−0.095825	−0.095825	0	240,000
MBA	−0.0958250	−0.0958250	−0.0958250	0	1600

The comparison of the obtained results for different algorithms is given in Table 7. The MBA obtained its best solution after 1600 function evaluations. From Table 7, although most considered methods reached the optimal solution, however, MBA found its optimal solution with the lowest NFEs and SD equal to zero. Fig. 6 represents the function values versus the number of iterations for this problem. As shown in Fig. 6, the MBA reached close to the optimum value in the early iterations of the algorithm.

3.5. Constrained problem 5

This minimization problem was previously solved using HM, ASCHEA, filter simulated annealing (FSA) [53], GA, NM-PSO, SR, CRGA, CULDE, PSO-DE, PSO, SAPF, SMES, DE, DELC, DEDS, ISR, HEAA, ABC, and α Simplex. The comparison of the obtained results for different algorithms is given in Table 8. Almost all considered optimizers found the best solution, however MBA reached its best solution faster with considerably less NFEs. Fig. 7 shows the function values in terms of the number of iterations for the constrained problem 5. By observing Fig. 7, the proposed algorithm converged to near optimum point in the early iterations.

3.6. Constrained problem 6

For this minimization problem, best solution for a number of optimizers was compared in Table 9. Table 10 represents the comparison of statistical results for the constrained problem 6 obtained using MBA, and the results reported by HM, ASCHEA, IGA, GA [11], GA1 [50], GA2 [54], CRGA, SMES, SAPF, PSO, SR, DE, CULDE, HS, coevolutionary particle swarm optimization using gaussian distribution (CPSO-GD) [55], DELC, DEDS, ISR, HEAA, α Simplex, ABC, particle evolutionary swarm optimization (PESO) [56], co-evolutionary differential evolution (CoDE) [57], and TLBO.

From Table 10 optimizers TLBO, α Simplex, ISR, DEDS, DELC, CULDE, SR, and GA have given more accurate optimal solutions as compared to MBA. However, the MBA requires comparatively less NFEs and offers satisfactory solution. CRGA outperformed MBA with 50,000 function evaluations, while MBA surpassed CRGA in terms of function value (accuracy). Fig. 8 illustrates the function values versus the number of iterations for constrained problem 6.

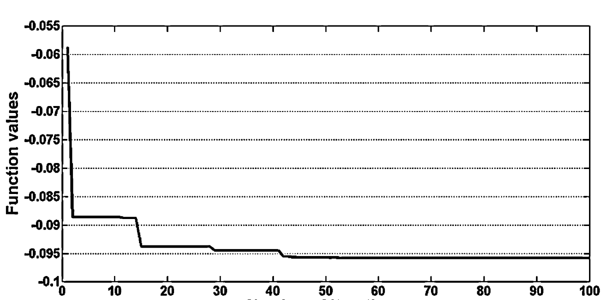

Fig. 6. Function values versus number of iterations for constrained problem 4.

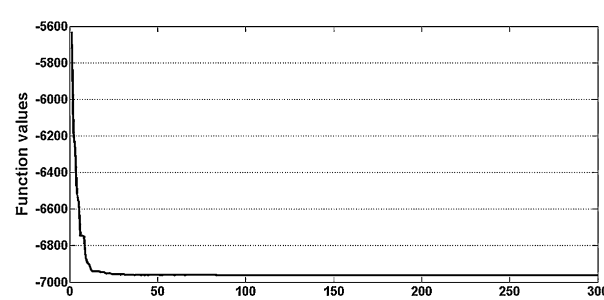

Fig. 7. Function values versus number of iterations for the constrained problem 5.

Please cite this article in press as: A. Sadollah, et al., Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.11.026

Table 8
Comparison of statistical results obtained from MBA and other optimizers for the constrained problem 5.

Method	Worst	Mean	Best	SD	NFEs
HM	−5473.9	−6342.6	−6952.1	N.A	1,400,000
ASCHEA	N.A	−6961.81	−6961.81	N.A	1,500,000
CULDE	−6961.813876	−6961.813876	−6961.813876	1E−07	100,100
DE	−6961.814	−6961.814	−6961.814	N.A	15,000
FSA	−6961.8139	−6961.8139	−6961.8139	0	44,538
GA	−6961.8139	−6961.8139	−6961.8139	0	13,577
CRGA	−6077.123	−6740.288	−6956.251	2.70E+2	3700
NM-PSO	−6961.8240	−6961.8240	−6961.8240	0	9856
PSO-DE	−6961.81388	−6961.81388	−6961.81388	2.3E−09	140,100
PSO	−6961.81381	−6961.81387	−6961.81388	6.5E−06	140,100
SR	−6350.262	−6875.940	−6961.814	160	118,000
SMES	−6962.482	−6961.284	−6961.814	1.85	240,000
SAPF	−6943.304	−6953.061	−6961.046	5.876	500,000
DELC	−6961.814	−6961.814	−6961.814	7.3E−10	20,000
DEDS	−6961.814	−6961.814	−6961.814	0	225,000
ABC	−6961.805	−6961.813	−6961.814	2E−03	240,000
HEAA	−6961.814	−6961.814	−6961.814	4.6E−12	200,000
ISR	−6961.814	−6961.814	−6961.814	1.9E−12	168,800
α Simplex	−6961.814	−6961.814	−6961.814	1.3E−10	293,367
MBA	−6961.813875	−6961.813875	−6961.813875	0	2835

Table 9
Comparison of the best solution given by previous studies and the proposed method for constrained problem 6.

D.V	IGA	HS	MBA	Optimal
X_1	2.330499	2.323456	2.326585	2.330499
X_2	1.951372	1.951242	1.950973	1.951372
X_3	−0.477541	−0.448467	−0.497446	−0.477541
X_4	4.365726	4.361919	4.367508	4.365726
X_5	−0.624487	−0.630075	−0.618578	−0.624487
X_6	1.038131	1.03866	1.043839	1.038131
X_7	1.594227	1.605348	1.595928	1.594227
$g_1(X)$	4.46E−05	0.208928	1.17E−04	4.46E−05
$g_2(X)$	−252.561723	−252.878859	−252.400363	−252.561723
$g_3(X)$	−144.878190	−145.123347	−144.912069	−144.878190
$g_4(X)$	7.63E−06	−0.263414	1.39E−04	7.63E−06
$f(X)$	680.63006	680.6413574	680.6322202	680.6300573

3.7. Constrained problem 7

This minimization problem was previously solved using HM, ASCHEA, SR, CAEP, HPSO, CRGA, DE, CULDE, PSO-DE, PSO, HS, SMES, SAPF, DELC, DEDS, ISR, HEAA, ABC, and α Simplex. Table 11 compares the results for CULDE, HS, GA1, GA2, and MBA with optimal solution. The statistical optimization results for different algorithms are given in Table 12. The MBA reached the optimal solution

Table 10
Comparison of statistical results for various algorithms including MBA for the constrained problem 6.

Method	Worst	Mean	Best	SD	NFEs
HM	683.1800	681.1600	680.9100	4.11E−02	1,400,000
ASCHEA	N.A	680.641	680.630	N.A	1,500,000
IGA	680.6304	680.6302	680.6301	1.00E−05	N.A
GA	680.6538	680.6381	680.6303	6.61E−03	320,000
GA1	680.6508	680.6417	680.6344	N.A	350,070
GA2	N.A	N.A	680.642	N.A	350,070
CRGA	682.965	681.347	680.726	5.70E−01	50,000
SAPF	682.081	681.246	680.773	0.322	500,000
SR	680.763	680.656	680.63	0.034	350,000
HS	N.A	N.A	680.6413	N.A	160,000
DE	680.144	680.503	680.771	0.67098	240,000
CULDE	680.630057	680.630057	680.630057	1E−07	100,100
PSO	684.5289146	680.9710606	680.6345517	5.1E−01	140,100
CPSO-GD	681.371	680.7810	680.678	0.1484	N.A
SMES	680.719	680.643	680.632	1.55E−02	240,000
DELC	680.630	680.630	680.630	3.2E−12	80,000
DEDS	680.630	680.630	680.630	2.9E−13	225,000
HEAA	680.630	680.630	680.630	5.8E−13	200,000
ISR	680.630	680.630	680.630	3.2E−13	271,200
α Simplex	680.630	680.630	680.630	2.9E−10	323,426
PESO	680.630	680.630	680.631	N.A	350,000
CoDE	685.144	681.503	680.771	N.A	248,000
ABC	680.638	680.640	680.634	4E−03	240,000
TLBO	680.638	680.633	680.630	N.A	100,000
MBA	680.7882	680.6620	680.6322	3.30E−02	71,750

Please cite this article in press as: A. Sadollah, et al., Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.11.026

Table 11
Comparison of best solution for the constrained problem 7 given by various optimizers.

D.V	CULDE	HS	GA1	GA2	MBA	Optimal
X_1	78.000000	78.0	80.39	78.0495	78.00000	78.00000
X_2	33.000000	33.0	35.07	33.007	33.00000	33.00000
X_3	29.995256	29.995	32.05	27.081	29.99526	29.99526
X_4	45.000000	45.0	40.33	45.00	44.99999	45.00000
X_5	36.775813	36.776	33.34	44.94	36.77581	36.77581
$g_1(X)$	1.35E−08	4.34E−05	−0.343809	1.283813	1.33E−08	−9.71E−04
$g_2(X)$	−92.00000001	−92.000043	−91.656190	−93.283813	−91.99999	−92
$g_3(X)$	−11.15945	−11.15949	−10.463103	−9.592143	−11.159499	−1.11E+01
$g_4(X)$	−8.840500	−8.840510	−9.536896	−10.407856	−8.84050	−8.87
$g_5(X)$	−4.999999	−5.000064	−4.974473	−4.998088	−4.99999	−5
$g_6(X)$	4.12E−09	6.49E−05	−0.025526	1.91E−03	−3.06E−09	9.27E−09
$f(X)$	−30665.5386	−30665.500	−30005.700	−31020.859	−30665.5386	−30665.539

Table 12
Comparison of statistical results for various optimizers for the constrained problem 7.

Method	Worst	Mean	Best	SD	NFEs
HM	−30645.900	−30665.300	−30664.500	N.A	1,400,000
ASCHEA	N.A	−30665.5	−30665.5	N.A	1,500,000
SR	−30665.539	−30665.539	−30665.539	2E−05	88,200
CAEP	−30662.200	−30662.500	−30665.500	9.3	50,020
PSO	−30252.3258	−30570.9286	−30663.8563	81	70,100
HPSO	−30665.539	−30665.539	−30665.539	1.7E−06	81,000
PSO-DE	−30665.5387	−30665.5387	−30665.5387	8.3E−10	70,100
CULDE	−30665.5386	−30665.5386	−30665.5386	1E−07	100,100
DE	−30665.509	−30665.536	−30665.539	5.067E−03	240,000
HS	N.A	N.A	−30665.500	N.A	65,000
CRGA	−30660.313	−30664.398	−30665.520	1.6	54,400
SAPF	−30656.471	−30655.922	−30665.401	2.043	500,000
SMES	−30665.539	−30665.539	−30665.539	0	240,000
ABC	−30665.539	−30665.539	−30665.539	0	240,000
DELC	−30665.539	−30665.539	−30665.539	1.0E−11	50,000
DEDS	−30665.539	−30665.539	−30665.539	2.7E−11	225,000
HEAA	−30665.539	−30665.539	−30665.539	7.4E−12	200,000
ISR	−30665.539	−30665.539	−30665.539	1.1E−11	192,000
α Simplex	−30665.539	−30665.539	−30665.539	4.2E−11	305,343
MBA	−30665.3300	−30665.5182	−30665.5386	5.08E−02	41,750

faster (less NFEs) than other compared algorithms in this paper as shown in Table 12. The function values versus the number of iterations for the constrained problem 7 are shown in Fig. 9.

3.8. Constrained problem 8

This minimization problem has n decision variables and one equality constraint. The optimal solution of the problem is at $X^* = (1/\sqrt{n}, \ldots, 1/\sqrt{n})$ with a corresponding function value of $f(x) = -1$. For this problem n is considered equal to 10. This problem was previously solved using HM, ASCHEA, PSO-DE, PSO, CULDE, SR, DE, SAPF, SMES, GA [11], CRGA, DELC, DEDS, ISR, HEAA, α Simplex, PESO, CoDE, ABC, and TLBO.

The statistical results of optimization for twenty-one algorithms including MBA are shown in Table 13. From Table 13, TLBO, ABC, HEAA, DELC, SMES, and SR have found the best optimal solution as compared to MBA. However, MBA requires comparatively less NFEs and provides a reasonably accurate solution. Fig. 10 shows the function values with respect to the number of iterations for the constrained problem 8.

3.9. Engineering benchmark constrained and mechanical design problem

3.9.1. Three-bar truss design problem

The three-bar truss problem is one of the engineering minimization test problems for constrained algorithms. The comparison of best solution for MBA, DEDS, and PSO-DE is presented in Table 14. The comparison of statistical results for MBA with previous studies including DEDS, PSO-DE, HEAA, and SC is presented in Table 15.

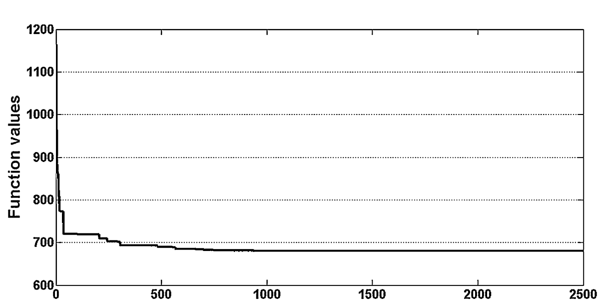

Fig. 8. Function values versus number of iterations for constrained problem 6.

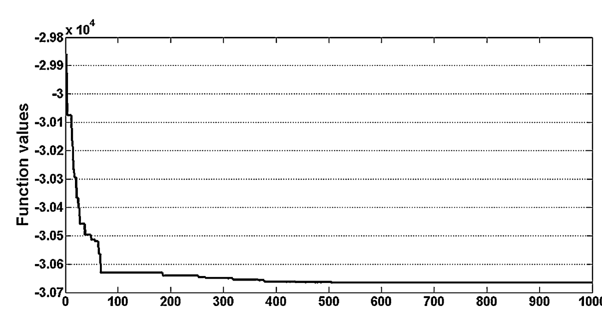

Fig. 9. Function values versus number of iterations for the constrained problem 7.

Table 13
Comparison of statistical results given by various algorithms for the constrained problem 8.

Method	Worst	Mean	Best	SD	NFEs
HM	−0.9978	−0.9989	−0.9997	N.A	1,400,000
ASCHEA	N.A	−0.99989	−1.0	N.A	1,500,000
PSO	−1.0042690	−1.0048795	−1.0049865	1.0E+0	140,100
PSO-DE	−1.0050100	−1.0050100	−1.0050100	3.8E−12	140,100
CULDE	−0.639920	−0.788635	−0.995413	0.115214	100,100
CRGA	−0.9931	−0.9975	−0.9997	1.4E−03	67,600
SAPF	−0.887	−0.964	−1.000	3.01E−01	500,000
SR	−1.0000	−1.0000	−1.0000	1.9E−04	229,000
ISR	−1.001	−1.001	−1.001	8.2E−09	349,200
DE	−1.0252	−1.0252	−1.0252	0	8,000,000
SMES	−1.000	−1.000	−1.000	2.09E−04	240,000
GA	−0.99979	0.99992	0.99998	5.99E−05	320,000
DELC	−1.000	−1.000	−1.000	2.1E−06	200,000
DEDS	−1.0005	−1.0005	−1.0005	1.9E−08	225,000
HEAA	−1.000	−1.000	−1.000	5.2E−15	200,000
α Simples	−1.0005	−1.0005	−1.0005	8.5E−14	310,968
PESO	−0.464	−0.764813	−0.993930	N.A	350,000
ABC	−1	−1	−1	0	240,000
TLBO	−1	−1	−1	0	100,000
MBA	−0.996539	−0.999147	−0.999813	5.44E−04	14,950

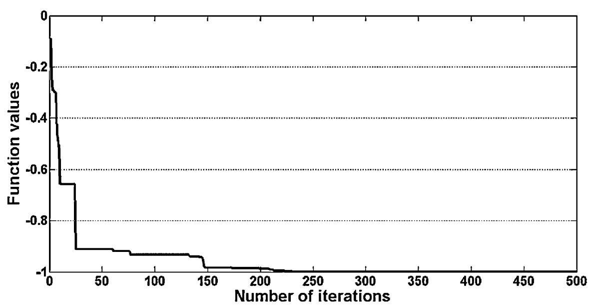

Fig. 10. Function values versus number of iterations for constrained problem 8.

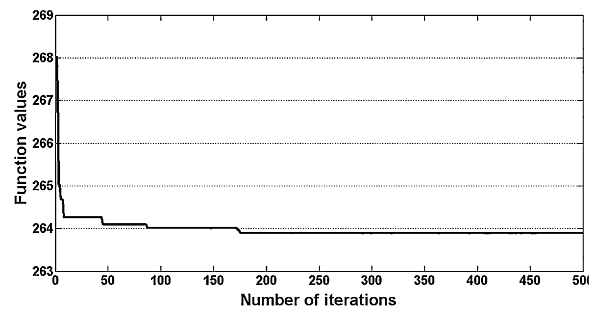

Fig. 11. Function values versus number of iterations for three-bar truss problem.

Table 14
Comparison of best solution obtained by previous algorithms for three-bar truss design problem.

D.V	DEDS	PSO-DE	MBA
X_1	0.78867513	0.7886751	0.7885650
X_2	0.40824828	0.4082482	0.4085597
$g_1(X)$	1.77E−08	−5.29E−11	−5.29E−11
$g_2(X)$	−1.4641016	−1.4637475	−1.4637475
$g_3(X)$	−0.53589836	−0.5362524	−0.5362524
$f(X)$	263.8958434	263.8958433	263.8958522

The optimization results obtained by all considered methods slightly outperformed the results given by MBA in terms of statistical results. However, the proposed MBA offered its best solution with less NFEs as indicated in Table 15. Fig. 11 shows the function values versus the number of iterations for the three-bar truss design problem.

3.9.2. *Pressure vessel design problem*

In the pressure vessel design problem, proposed by Kannan and Kramer [58], the aim is to minimize the total cost, including the cost of material, forming, and welding. A cylindrical vessel is capped at both ends by hemispherical heads as shown in Fig. 12. They are four design variables in this problem: T_s (x_1, thickness of the shell), T_h (x_2, thickness of the head), R (x_3, inner radius), and L (x_4, length of the cylindrical section of the vessel). Among the four design variables, T_s (x_1) and T_h (x_2) are expected to be integer multiples of 0.0625 *in*, and R and L are continuous variables.

Table 16 represents the comparisons of best solution for MBA and other reported methods. This problem has been solved previously using other optimizers including GA based co-evolution model (GA3) [59], GA through the use of dominance-based tour tournament selection (GA4) [60], CPSO, CDE, HPSO, NM-PSO, G-QPSO, QPSO, PSO, PSO-DE, unified particle swarm optimization (UPSO) [61], ABC [62], $(\mu+\lambda)$-ES [63], and TLBO.

The reported optimization results were compared with the proposed method in terms of statistical results as given in Table 17. From this table, MBA outperforms all other algorithms in terms of best solution. However, the NFEs for MBA are higher as

Table 15
Comparison of statistical results obtained using various algorithms for the three-bar truss design problem.

Method	Worst	Mean	Best	SD	NFEs
SC	263.969756	263.903356	263.895846	1.3E−02	17,610
PSO-DE	263.895843	263.895843	263.895843	4.5E−10	17,600
DEDS	263.895849	263.895843	263.895843	9.7E−07	15,000
HEAA	263.896099	263.895865	263.895843	4.9E−05	15,000
MBA	263.915983	263.897996	263.895852	3.93E−03	13,280

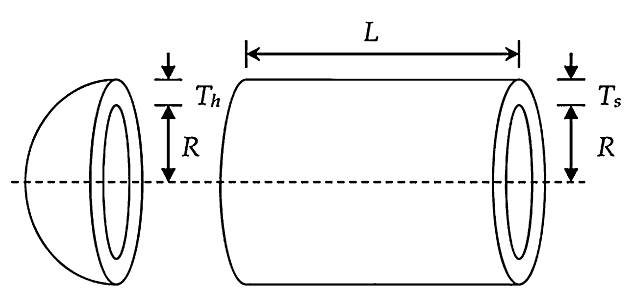

Fig. 12. Pressure vessel design problem.

Table 16
Comparison of best solution obtained from various previous studies for pressure vessel problem.

D.V	GA3	GA4	CPSO	HPSO	NM-PSO	G-QPSO	CDE	MBA
X_1	0.8125	0.8125	0.8125	0.8125	0.8036	0.8125	0.8125	0.7802
X_2	0.4375	0.4375	0.4375	0.4375	0.3972	0.4375	0.4375	0.3856
X_3	40.3239	42.0974	42.0913	42.0984	41.6392	42.0984	42.098411	40.4292
X_4	200.0000	176.6540	176.7465	176.6366	182.4120	176.6372	176.637690	198.4964
$g_1(X)$	−3.42E−02	−2.01E−03	−1.37E−06	−8.80E−07	3.65E−05	−8.79E−07	−6.67E−07	0
$g_2(X)$	−5.28E−02	−3.58E−02	−3.59E−04	−3.58E−02	3.79E−05	−3.58E−02	−3.58E−02	0
$g_3(X)$	−304.4020	−24.7593	−118.7687	3.1226	−1.5914	−0.2179	−3.705123	−86.3645
$g_4(X)$	−400.0000	−63.3460	−63.2535	−63.3634	−57.5879	−63.3628	−63.362310	−41.5035
$f(X)$	6288.7445	6059.9463	6061.0777	6059.7143	5930.3137	6059.7208	6059.7340	5889.3216

Table 17
Comparison of statistical results given by different methods for pressure vessel design problem.

Method	Worst	Mean	Best	SD	NFEs
GA3	6308.4970	6293.8432	6288.7445	7.4133	900,000
GA4	6469.3220	6177.2533	6059.9463	130.9297	80,000
CPSO	6363.8041	6147.1332	6061.0777	86.45	240,000
HPSO	6288.6770	6099.9323	6059.7143	86.20	81,000
NM-PSO	5960.0557	5946.7901	5930.3137	9.161	80,000
G-QPSO	7544.4925	6440.3786	6059.7208	448.4711	8000
QPSO	8017.2816	6440.3786	6059.7209	479.2671	8000
PSO	14076.3240	8756.6803	6693.7212	1492.5670	8000
CDE	6371.0455	6085.2303	6059.7340	43.0130	204,800
UPSO	N.A	9032.55	6544.27	995.573	100,000
PSO-DE	N.A	6059.714	6059.714	N.A	42,100
ABC	N.A	6245.308	6059.714	205	30,000
$(\mu+\lambda)$-ES	N.A	6379.938	6059.7016	210	30,000
TLBO	N.A	6059.71434	6059.714335	N.A	10,000
MBA	6392.5062	6200.64765	5889.3216	160.34	70,650

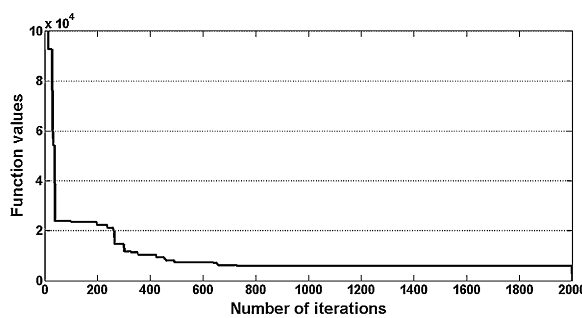

Fig. 13. Function values versus number of iterations for the pressure vessel problem.

compared to TLBO, PSO, QPSO, and G-QPSO algorithms. Fig. 13 shows the function values versus the number of iterations for pressure vessel design problem.

3.9.3. Tension/compression spring design problem

The tension/compression spring design problem is described in Arora [64] for which the aim is to minimize the weight ($f(x)$) of a tension/compression spring (as shown in Fig. 14) subject to constraints on minimum deflection, shear stress, surge frequency, limits on outside diameter, and design variables. The design variables are wire diameter $d(x_1)$, mean coil diameter $D(x_2)$, and number of active coils $P(x_3)$.

The comparison of best solution among several algorithms is given in Table 18. This problem has been used as a benchmark

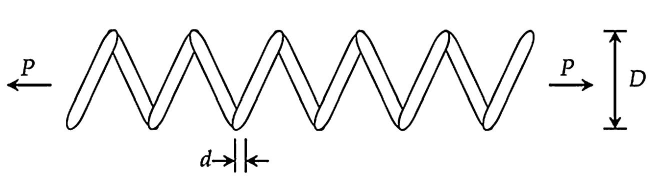

Fig. 14. Tension/compression string design problem.

problem for testing different optimization methods, such as GA3, GA4, CAEP, UPSO, CPSO, HPSO, NM-PSO, G-QPSO, QPSO, PSO-DE, PSO, DELC, DEDS, HEAA, SC, DE, CDE, $(\mu+\lambda)$-ES, ABC, and TLBO.

The obtained statistical results by the considered methods and MBA are given in Table 19. As it can be seen in Table 19, MBA requires only 7650 function evaluations for solving this problem with function value of 0.012665, while G-QPSO, QPSO, and PSO require 2000 function evaluations with function values of 0.012665, 0.012669, and 0.012857, respectively. The best function value is 0.0126302 with 80,000 function evaluations obtained by NM-PSO.

For other algorithms in Table 19, MBA shows superiority in terms of NFEs and the obtained function value. However, TLBO and ABC algorithms have both given the same function value while they require slightly higher NFEs as compared to MBA. Fig. 15 shows the function values versus the number of iterations for tension/compression spring problem.

3.9.4. Welded beam design problem

This design problem, which has been often used as a benchmark problem, was firstly proposed by Coello [59]. In this problem, a welded beam is designed for minimum cost subject to constraints

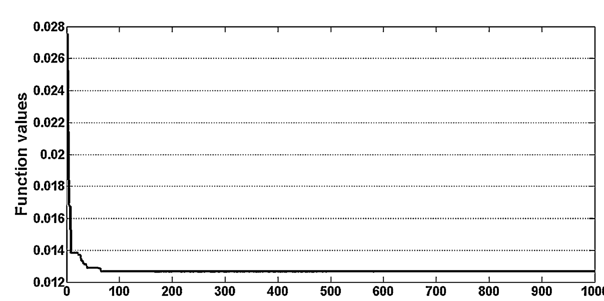

Fig. 15. Function values versus number of iterations for tension/compression spring problem.

Table 18
Comparison of best solution obtained from various algorithms for the tension/compression spring problem.

D.V	DELC	DEDS	CPSO	HPSO	NM-PSO	G-QPSO	HEAA	MBA
X_1	0.051689	0.051689	0.051728	0.051706	0.051620	0.051515	0.051689	0.051656
X_2	0.356717	0.356717	0.357644	0.357126	0.355498	0.352529	0.356729	0.355940
X_3	11.288965	11.288965	11.244543	11.265083	11.333272	11.538862	11.288293	11.344665
$g_1(X)$	−3.40E−09	1.45E−09	−8.25E−04	−3.06E−06	1.01E−03	−4.83E−05	3.96E−10	0
$g_2(X)$	2.44E−09	−1.19E−09	−2.52E−05	1.39E−06	9.94E−04	−3.57E−05	−3.59E−10	0
$g_3(X)$	−4.053785	−4.053785	−4.051306	−4.054583	−4.061859	−4.0455	−4.053808	−4.052248
$g_4(X)$	−0.727728	−0.727728	−0.727085	−0.727445	−0.728588	−0.73064	−0.727720	−0.728268
$f(X)$	0.012665	0.012665	0.0126747	0.0126652	0.0126302	0.012665	0.012665	0.012665

Table 19
Comparisons of the statistical results given by different algorithms for the tension/compression spring problem.

Method	Worst	Mean	Best	SD	NFEs
GA3	0.0128220	0.0127690	0.0127048	3.94E−05	900,000
GA4	0.0129730	0.0127420	0.0126810	5.90E−05	80,000
CAEP	0.0151160	0.0135681	0.0127210	8.42E−04	50,020
CPSO	0.0129240	0.0127300	0.0126747	5.20E−04	240,000
HPSO	0.0127190	0.0127072	0.0126652	1.58E−05	81,000
NM-PSO	0.0126330	0.0126314	0.0126302	8.47E−07	80,000
G-QPSO	0.017759	0.013524	0.012665	0.001268	2000
QPSO	0.018127	0.013854	0.012669	0.001341	2000
PSO	0.071802	0.019555	0.012857	0.011662	2000
DE	0.012790	0.012703	0.0126702	2.7E−05	204,800
DELC	0.012665575	0.012665267	0.012665233	1.3E−07	20,000
DEDS	0.012738262	0.012669366	0.012665233	1.3E−05	24,000
HEAA	0.012665240	0.012665234	0.012665233	1.4E−09	24,000
PSO-DE	0.012665304	0.012665244	0.012665233	1.2E−08	24,950
SC	0.016717272	0.012922669	0.012669249	5.9E−04	25,167
UPSO	N.A	0.02294	0.01312	7.20E−03	100,000
CDE	N.A	0.012703	0.01267	N.A	240,000
$(\mu+\lambda)$-ES	N.A	0.013165	0.012689	3.9E−04	30,000
ABC	N.A	0.012709	0.012665	0.012813	30,000
TLBO	N.A	0.01266576	0.012665	N.A	10,000
MBA	0.012900	0.012713	0.012665	6.30E−05	7650

on shear stress (τ), bending stress (σ) in the beam, buckling load on the bar (P_b), end deflection of the beam (δ), and side constraints. There are four design variables for this problem including $h(x_1)$, $l(x_2)$, $t(x_3)$, and $b(x_4)$ as shown in Fig. 16.

The optimization methods previously applied to this problem include GA3, GA4, CAEP, CPSO, HPSO, NM-PSO, HGA, MGA [65], SC, DE, UPSO, CDE, ABC, $(\mu+\lambda)$-ES, and TLBO. The comparison for the best solution given by such algorithms is presented in Table 20.

The comparison of the statistical results is given in Table 21. Among those studies, the best solution was obtained by NM-PSO with an objective function value of $f(x) = 1.724717$ after 80,000 function evaluations. By applying the proposed method, the best solution of $f(x) = 1.72536$ was obtained. It is worth to mention that the number of function evaluations for the MBA is 42,960. The optimization results obtained by NM-PSO, PSO-DE, HPSO, CAEP, ABC, and TLBO outperformed the results obtained by MBA in terms of the best solution. However, while offering satisfactory result, MBA requires less NFEs as compared to the NM-PSO, PSO-DE, HPSO, and CAEP.

Fig. 17 shows the function values versus the number of iterations for the welded beam design problem. One of the advantages of the MBA over other metaheuristic algorithms is that the function values are reduced to near optimum point at the early iterations. This may be due to the searching criteria and constraint handling approach for MBA where it searches a wide region of problem domain and rapidly focuses on the near optimum solution.

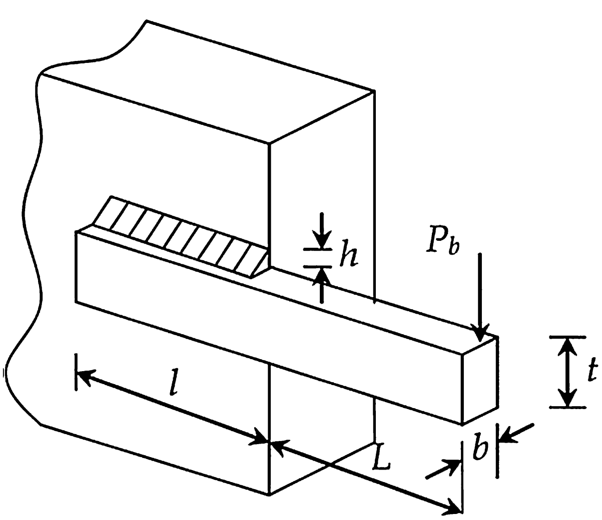

Fig. 16. Welded beam design problem.

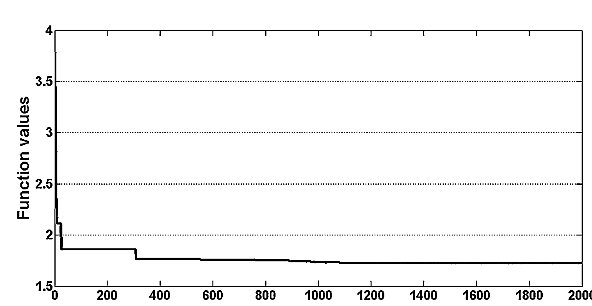

Fig. 17. Function values versus number of iterations for the welded beam problem.

Table 20
Comparison of the best solution obtained from various algorithms for the welded beam problem.

D.V	GA4	HGA	CPSO	CAEP	CPSO	HPSO	NM-PSO	MBA
$X_1(h)$	0.205986	0.2057	0.202369	0.205700	0.202369	0.20573	0.20583	0.205729
$X_2(l)$	3.471328	3.4705	3.544214	3.470500	3.544214	3.470489	3.468338	3.470493
$X_3(t)$	9.020224	9.0366	9.048210	9.036600	9.04821	9.036624	9.036624	9.036626
$X_4(b)$	0.206480	0.2057	0.205723	0.205700	0.205723	0.20573	0.20573	0.205729
$g_1(X)$	−0.103049	1.988676	−13.655547	1.988676	−12.839796	−0.025399	−0.02525	−0.001614
$g_2(X)$	−0.231747	4.481548	−78.814077	4.481548	−1.247467	−0.053122	−0.053122	−0.016911
$g_3(X)$	−5E−04	0	−3.35E−03	0	−1.49E−03	0	0.0001	−2.40E−07
$g_4(X)$	−3.430044	−3.433213	−3.424572	−3.433213	−3.429347	−3.432981	−3.433169	−3.432982
$g_5(X)$	−0.080986	−0.080700	−0.077369	−0.080700	−0.079381	−0.08073	−0.08083	−0.080729
$g_6(X)$	−0.235514	−0.235538	−0.235595	−0.235538	−0.235536	−0.235540	−0.235540	−0.235540
$g_7(X)$	−58.646888	2.603347	−4.472858	2.603347	−11.681355	−0.031555	−0.031555	−0.001464
$f(X)$	1.728226	1.724852	1.728024	1.724852	1.728024	1.724852	1.724717	1.724853

Table 21
Comparison of the statistical results obtained from different optimizers for welded beam problem.

Method	Worst	Mean	Best	SD	NFEs
GA3	1.785835	1.771973	1.748309	1.12E−02	900,000
GA4	1.993408	1.792654	1.728226	7.47E−02	80,000
CAEP	3.179709	1.971809	1.724852	4.43E−01	50,020
CPSO	1.782143	1.748831	1.728024	1.29E−02	240,000
HPSO	1.814295	1.749040	1.724852	4.01E−02	81,000
PSO-DE	1.724852	1.724852	1.724852	6.7E−16	66,600
NM-PSO	1.733393	1.726373	1.724717	3.50E−03	80,000
MGA	1.9950	1.9190	1.8245	5.37E−02	N.A
SC	6.3996785	3.0025883	2.3854347	9.6E−01	33,095
DE	1.824105	1.768158	1.733461	2.21E−02	204,800
UPSO	N.A	2.83721	1.92199	0.683	100,000
CDE	N.A	1.76815	1.73346	N.A	240,000
$(\mu+\lambda)$-ES	N.A	1.777692	1.724852	8.8E−02	30,000
ABC	N.A	1.741913	1.724852	3.1E−02	30,000
TLBO	N.A	1.72844676	1.724852	N.A	10,000
MBA	1.724853	1.724853	1.724853	6.94E−19	47,340

3.9.5. Speed reducer design problem

In this constrained optimization problem (see Fig. 18), the weight of speed reducer is to be minimized subject to constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts, and stresses in the shafts [63]. The variables x_1 to x_7 represent the face width (b), module of teeth (m), number of teeth in the pinion (z), length of the first shaft between bearings (l_1), length of the second shaft between bearings (l_2), and the diameter of first (d_1) and second shafts (d_2), respectively.

This is an example of a mixed integer programming problem. The third variable x_3 (number of teeth) is of integer values while all other variables are continuous. There are 11 constraints in this problem resulting in high complexity of the problem [66] (the solution reported in [66] is infeasible).

The comparison of best solution with previous methods is given in Table 22. The statistical results of nine optimization methods including DELC, DEDS, PSO-DE, ABC, TLBO, modified differential evolution (MDE) [67,68], SC, HEAA, and $(\mu+\lambda)$-ES is compared with the proposed method which is given in Table 23.

From Table 23, among the compared optimization algorithms, DELC and DEDS have found the best solution so far. Although, MBA could not match the best solution obtained by DELC and DEDS, however, it detected its best solution (second best solution) with considerably less NFEs. Fig. 19 depicts the reduction of function values versus the number of iterations for speed reducer design problem.

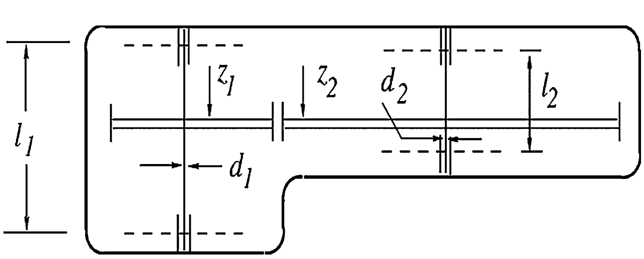

Fig. 18. Speed reducer design problem.

3.9.6. Gear train design problem

Gear train design aims to minimize the cost of the gear ratio of the gear train as shown in Fig. 20. The constraints are only limits on design variables (side constraints). Design variables to be optimized are in discrete form since each gear has to have an integral number of teeth.

Constrained problems with discrete variables may increase the complexity of the problem. The decision variables of the problem are n_A, n_B, n_D, and n_F which are denoted as x_1, x_2, x_3, and x_4, respectively. The lower and upper bounds of integer design variables are 12 and 60, respectively. The gear ratio is defined as n_Bn_D/n_Fn_A. Table 24 shows the comparison of the best solution for ABC and MBA in terms of the value of design variables and function value.

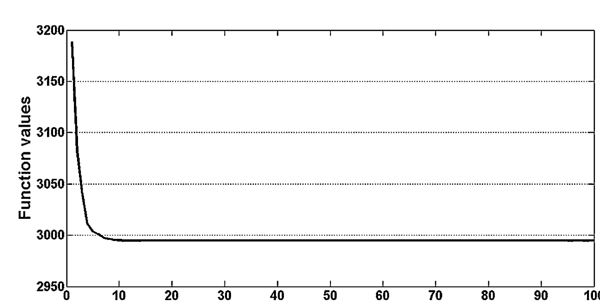

Fig. 19. Function values versus number of iterations for the speed reducer problem.

Table 22
Comparison of best solution obtained using different optimizers for speed reducer design problem.

D.V	DEDS	DELC	HEAA	MDE	PSO-DE	MBA
X_1	3.5 + 09	3.5 + 09	3.500022	3.500010	3.5000000	3.500000
X_2	0.7 + 09	0.7 + 09	0.70000039	0.70000	0.700000	0.700000
X_3	17	17	17.000012	17	17.000000	17.000000
X_4	7.3 + 09	7.3 + 09	7.300427	7.300156	7.300000	7.300033
X_5	7.715319	7.715319	7.715377	7.800027	7.800000	7.715772
X_6	3.350214	3.350214	3.350230	3.350221	3.350214	3.350218
X_7	5.286654	5.286654	5.286663	5.286685	5.2866832	5.286654
$f(X)$	2994.471066	2994.471066	2994.499107	2996.356689	2996.348167	2994.482453

Table 23
Comparison of statistical results using various algorithms for speed reducer design problem.

Method	Worst	Mean	Best	SD	NFEs
SC	3009.964736	3001.758264	2994.744241	4.0	54,456
PSO-DE	2996.348204	2996.348174	2996.348167	6.4E−06	54,350
DELC	2994.471066	2994.471066	2994.471066	1.9E−12	30,000
DEDS	2994.471066	2994.471066	2994.471066	3.6E−12	30,000
HEAA	2994.752311	2994.613368	2994.499107	7.0E−02	40,000
MDE	N.A	2996.367220	2996.356689	8.2E−03	24,000
$(\mu+\lambda)$-ES	N.A	2996.348	2996.348	0	30,000
ABC	N.A	2997.058	2997.058	0	30,000
TLBO	N.A	2996.34817	2996.34817	0	10,000
MBA	2999.652444	2996.769019	2994.482453	1.56	6300

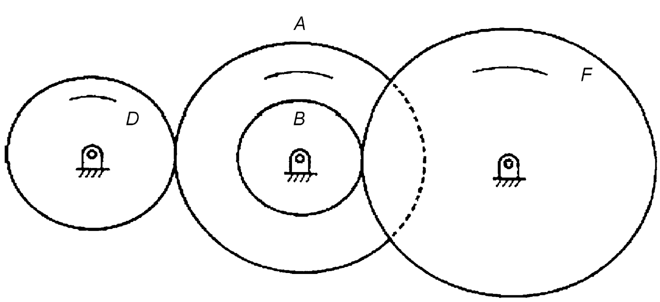

Fig. 20. Great train design problem.

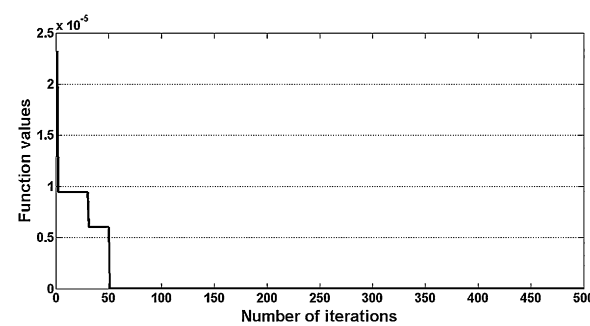

Fig. 21. Function values versus number of iterations for the gear train problem.

MBA found the function value same as ABC, however, with different values for design variables as shown in Table 24.

The statistical results for optimization of gear train problem using different optimizers are given in Table 25. In terms of statistical results, MBA surpassed other reported optimizers with acceptable NFEs as shown in Table 25. However, ABC obtained the best solution faster (less NFEs) than other methods. Fig. 21 demonstrates the reduction of function values (gear ratio cost) with respect to the number of iterations.

3.9.7. *Belleville spring design problem*

The Belleville spring problem, as shown in Fig. 22, is a minimization problem in which one parameter existing in the constraints is to be selected according to the design variable ratios. The objective is to design a Belleville spring having minimum weight and satisfying a number of constraints. This problem has 4 design variables: external diameter of the spring (D_e), internal diameter of the spring (D_i), thickness of the spring (t), and the height (h) of the spring, as shown in Fig. 22.

The subjected constraints concern the compressive stress, deflection, height to deflection, height to maximum height, outer diameter, inner diameter, and slope. The problem for the Belleville spring was studied by Coello [69], Gene AS I, Gene AS II [70], Siddal [71], ABC, and TLBO. Table 26 represents the comparisons of different optimizers for the best obtained solution.

The obtained statistical results for Belleville spring problem were compared in Table 27. It is observed from Table 27 that MBA surpassed ABC and TLBO for offering the best solution in terms of SD and NFEs. The best solution obtained by all three algorithms is same. However, TLBO performs slightly better than MBA in terms

Table 24
Comparison of best solution obtained using ABC and MBA for gear train problem.

D.V	ABC	MBA
X_1	49	43
X_2	16	16
X_3	19	19
X_4	43	49
$f(X)$	2.700857E−12	2.700857E−12

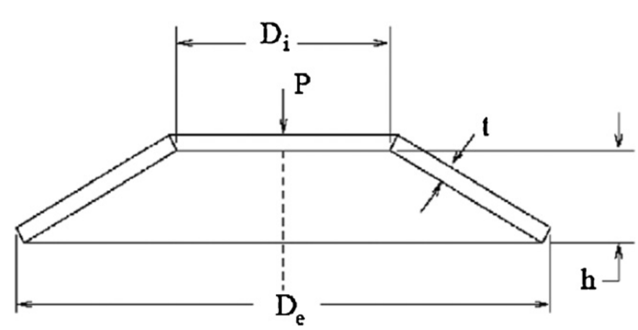

Fig. 22. Belleville spring design problem.

Table 25
Comparison of statistical results using various algorithms for gear train problem.

Method	Worst	Mean	Best	SD	NFEs
UPSO	N.A	3.80562E−08	2.700857E−12	1.09E−07	100,000
ABC	N.A	3.641339E−10	2.700857E−12	5.52E−10	60
MBA	2.062904E−08	2.471635E−09	2.700857E−12	3.94E−09	1120

Table 26
Comparison of best solution obtained using TLBO and MBA for Belleville spring problem.

D.V	Coello	Gene AS I	Gene AS II	Siddal	TLBO	MBA
X_1	0.208	0.205	0.210	0.204	0.204143	0.204143
X_2	0.2	0.201	0.204	0.200	0.2	0.2
X_3	8.751	9.534	9.268	10.030	10.03047	10.0304732
X_4	11.067	11.627	11.499	12.010	12.01	12.01
$g(X_1)$	2145.4109	−10.3396	2127.2624	134.0816	1.77E−06	4.58E−04
$g(X_2)$	39.75018	2.8062	194.222554	−12.5328	7.46E−08	3.04E−07
$g(X_3)$	0.00000	0.0010	0.0040	0.0000	5.8E−11	9.24E−10
$g(X_4)$	1.592	1.5940	1.5860	1.5960	1.595857	1.595856
$g(X_5)$	0.943	0.3830	0.5110	0.0000	2.35E−09	0
$g(X_6)$	2.316	2.0930	2.2310	1.9800	1.979527	1.979526
$g(X_7)$	0.21364	0.20397	0.20856	0.19899	0.198966	0.198965
$f(X)$	2.121964	2.01807	2.16256	1.978715	1.979675	1.9796747

Table 27
Comparison of statistical results using three optimizers for the Belleville spring problem.

Method	Worst	Mean	Best	SD	NFEs
ABC	2.104297	1.995475	1.979675	0.07	150,000
TLBO	1.979757	1.97968745	1.979675	0.45	150,000
MBA	2.005431	1.984698	1.9796747	7.78E−03	10,600

of mean and worst solutions. As shown in Fig. 23a, the convergence rate of ABC and TLBO is nearly same with slight dominance of TLBO over ABC. However, as shown in Fig. 23b, MBA converged to near optimum solution at early iterations compared to other optimizers in this paper.

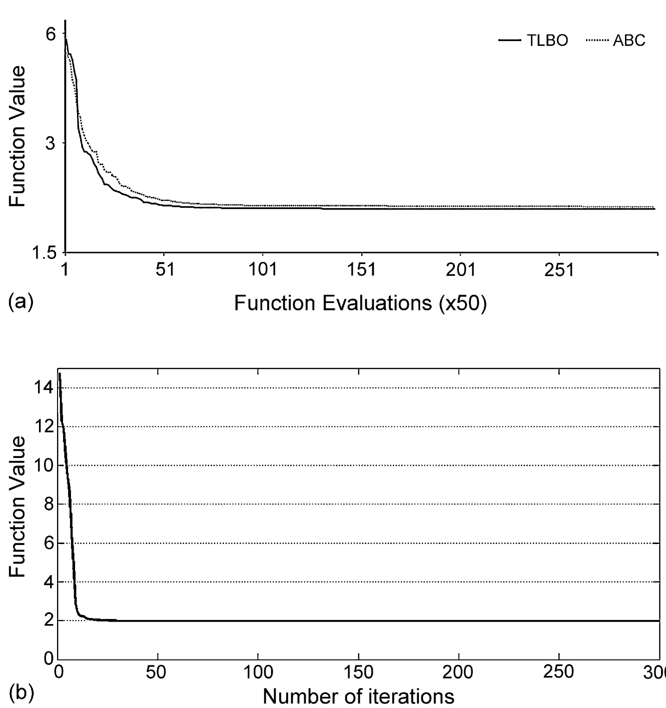

Fig. 23. Comparison of convergence rate for the Belleville spring design problem using: (a) TLBO and ABC, (b) MBA.

3.9.8. *Rolling element bearing design problem*

The objective of this problem is to maximize the dynamic load carrying capacity of a rolling element bearing, as demonstrated in Fig. 24. This problem has 10 decision variables which are pitch diameter (D_m), ball diameter (D_b), number of balls (Z), inner and outer raceway curvature coefficients (f_i and f_o), K_{Dmin}, K_{Dmax}, ε, e, and ζ (see Fig. 24). The five latter variables only appear in constraints and indirectly affect the internal geometry. The number of balls (Z) is the discrete design variable and the remainder are continuous design variables. Constraints are imposed based on kinematic and manufacturing considerations.

The problem of the rolling element bearing was studied by GA [72], ABC, and TLBO. Table 28 shows the comparison of best solution for three optimizers in terms of design variables and function values, and constraints accuracy. The statistical optimization results for reported algorithms were compared in Table 29.

From Table 29, the proposed method detected the best solution with considerable improvement over other optimizers in this

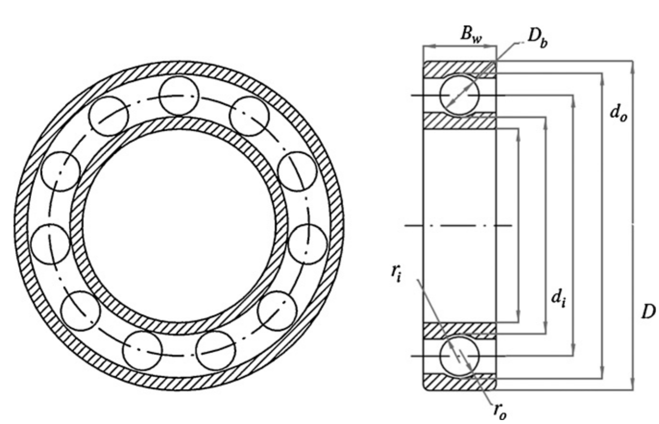

Fig. 24. Rolling element bearing design problem.

Table 28
Comparison of the best solution obtained using three algorithms for the rolling element bearing problem.

D.V	GA	TLBO	MBA
X_1	125.7171	125.7191	125.7153
X_2	21.423	21.42559	21.423300
X_3	11	11	11.000
X_4	0.515	0.515	0.515000
X_5	0.515	0.515	0.515000
X_6	0.4159	0.424266	0.488805
X_7	0.651	0.633948	0.627829
X_8	0.300043	0.3	0.300149
X_9	0.0223	0.068858	0.097305
X_{10}	0.751	0.799498	0.646095
$g(X_1)$	−0.000821	0	0
$g(X_2)$	−13.732999	13.15257	−8.630183
$g(X_3)$	−2.724000	1.5252	−1.101429
$g(X_4)$	3.606000	0.719056	−2.040448
$g(X_5)$	−0.717000	16.49544	−0.715366
$g(X_6)$	−4.857899	0	−23.611002
$g(X_7)$	−0.003050	0	−0.000480
$g(X_8)$	−0.000007	2.559363	0
$g(X_9)$	−0.000007	0	0
$g(X_{10})$	−0.000005	0	0
$f(X)$	81843.3	81859.74	85535.9611

Table 29
Comparison of statistical results using four optimizers for the rolling element bearing problem.

Method	Worst	Mean	Best	SD	NFEs
GA	N.A	N.A	81843.3	N.A	225,000
ABC	78897.81	81496	81859.7416	0.69	10,000
TLBO	80807.8551	81438.987	81859.74	0.66	10,000
MBA	84440.1948	85321.4030	85535.9611	211.52	15,100

paper. In terms of statistical results, MBA offered better results with acceptable NFEs against other considered algorithms. Fig. 25 compares the convergence rate for used optimizers.

From Fig. 25a it is seen that the convergence rate of ABC and TLBO is nearly same with a slightly higher mean searching capability for TLBO. However, MBA reached the best solution at 302 iterations (iteration number for obtained best solution = NFEs/N_s), offering the best solution so far as shown in Fig. 25b.

These overall results may suggest that the proposed MBA may be considered as an effective optimization technique for solving constrained optimization problems. The MBA may be considered as an attractive alternative optimizer for constrained and engineering optimization offering faster convergence and solution quality.

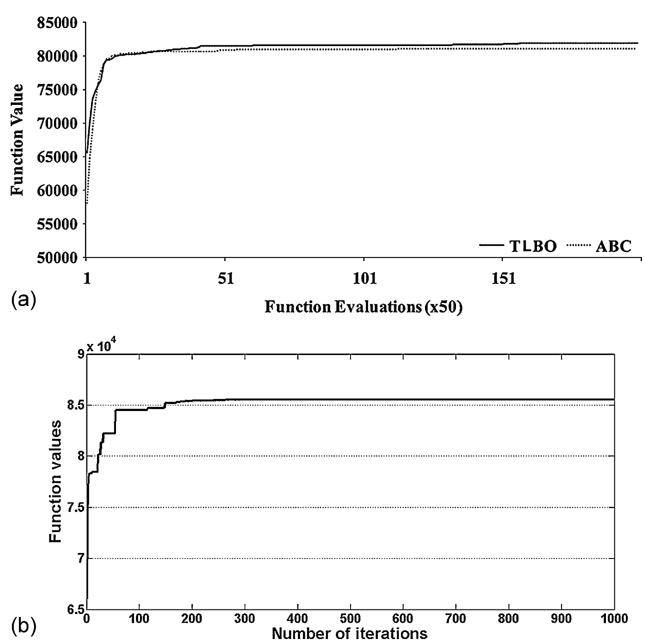

Fig. 25. Comparison of convergence rate for the rolling element bearing design problem using: (a) TLBO and ABC, (b) MBA.

4. Conclusions

This paper presented a new optimization technique, called the mine blast algorithm (MBA). The fundamental concepts and ideas which underlie the method are inspired from the explosion of mine bombs in real world. In this paper, the MBA is proposed for solving sixteen constrained optimization and engineering problems. Computational results in this paper, based on the comprehensive comparative study, illustrate the attractiveness of the proposed method for handling numerous types of constraints. The obtained results show that the proposed algorithm generally offers better solutions than other optimizers considered in this research in terms of objective function values for some problems and the number of function evaluations (computational cost) for almost every problem. Also, the MBA method may be used for solving the real world optimization problems which may require significant computational efforts with acceptable degree of accuracy for the solutions.

Although, the proposed method is capable in finding global optimum point, however, in case of obtaining unsatisfactory results, the tuning of initial parameters (as in other metaheuristic algorithms) is a crucial step. In general, the MBA offers competitive solutions compared with other metaheuristic optimizers based on the experimental results in this research. However, the computational efficiency and quality of solutions given by the MBA depends on the nature and complexity of the underlined problem. This is true for most metaheuristic methods.

Acknowledgments

The authors would like to acknowledge for the Ministry of Higher Education of Malaysia and the University of Malaya, Kuala Lumpur, Malaysia for the financial support under UM.TNC2/IPPP/UPGP/628/6/ER013/2011A.

Appendix A.

A.1. Constrained problem 1

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

subject to:

$$h(x) = x_1 - 2x_2 + 1 = 0$$

$$g(x) = -(x_1^2/4) - x_2^2 + 1 \geq 0$$

$$-10 \leq x_i \leq 10, \quad i = 1, 2$$

A.2. Constrained problem 2

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

subject to:

$$h(x) = x_2 - x_1^2 = 0$$

$$-1 \leq x_i \leq 1, \quad i = 1, 2$$

Please cite this article in press as: A. Sadollah, et al., Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.11.026

A.3. Constrained problem 3

$$f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$

subject to:

$$g_1(x) = 4.84 - (x_1 - 0.05)^2 - (x_2 - 2.5)^2 \geq 0$$

$$g_2(x) = x_1^2 + (x_2 - 2.5)^2 - 4.84 \geq 0$$

$$0 \leq x_i \leq 6, \quad i = 1, 2$$

A.4. Constrained problem 4

$$\max f(x) = \frac{\sin^3(2\pi x_1)\ \sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$$

subject to:

$$g_1(x) = x_1^2 - x_2 + 1 \leq 0$$

$$g_2(x) = 1 - x_1 + (x_2 - 4)^2 \leq 0$$

$$0 \leq x_i \leq 10, \quad i = 1, 2$$

A.5. Constrained problem 5

$$\min f(x) = (x_1 - 10)^3 + (x_2 - 20)^3$$

subject to:

$$g_1(x) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0$$

$$g_2(x) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0$$

$$13 \leq x_1 \leq 100$$

$$0 \leq x_2 \leq 100$$

A.6. Constrained problem 6

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

subject to:

$$g_1(x) = 127 - 2x_1^2 - 3x_2^4 - x_3 - 4x_4^2 - 5x_5 \geq 0$$

$$g_2(x) = 282 - 7x_1 - 3x_2 - 10x_3^2 - x_4 + x_5 \geq 0$$

$$g_3(x) = 196 - 23x_1 - x_2^2 - 6x_6^2 + 8x_7 \geq 0$$

$$g_4(x) = -4x_1^2 - x_2^2 + 3x_1x_2 - 2x_3^2 - 5x_6 + 11x_7 \geq 0$$

$$-10 \leq x_i \leq 10, \quad i = 1, 2, 3, 4, 5, 6, 7$$

A.7. Constrained problem 7

$$\begin{aligned} \min f(x) = {} & 5.3578547x_3^3 + 0.8356891x_1x_5 \\ & + 37.293239x_1 + 40729.141 \end{aligned}$$

subject to:

$$\begin{aligned} g_1(x) = {} & 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 \\ & - 0.0022053x_3x_5 - 92 \leq 0 \end{aligned}$$

$$\begin{aligned} g_2(x) = {} & -85.334407 - 0.0056858x_2x_5 \\ & - 0.0006262x_1x_4 - 0.0022053x_3x_5 \leq 0 \end{aligned}$$

$$\begin{aligned} g_3(x) = {} & 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 \\ & + 0.0021813x_3^2 - 110 \leq 0 \end{aligned}$$

$$\begin{aligned} g_4(x) = {} & -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 \\ & - 0.0021813x_3^2 + 90 \leq 0 \end{aligned}$$

$$\begin{aligned} g_5(x) = {} & 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 \\ & + 0.0019085x_3x_4 - 25 \leq 0 \end{aligned}$$

$$\begin{aligned} g_6(x) = {} & -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 \\ & - 0.0019085x_3x_4 + 20 \leq 0 \end{aligned}$$

$$78 \leq x_1 \leq 102$$

$$33 \leq x_2 \leq 45$$

$$27 \leq x_i \leq 45, \quad i = 3, 4, 5$$

A.8. Constrained problem 8

$$\min f(x) = -(\sqrt{n})^n \cdot \prod_{i=1}^{n} x_i$$

subject to:

$$h(x) = \sum_{i=1}^{n} x_i^2 = 1$$

$$0 \leq x_i \leq 1, \quad i = 1, \ldots, n$$

Appendix B.

B.1. Three-bar truss design problem

$$\min f(x) = (2\sqrt{2}x_1 + x_2) \times l$$

subject to:

$$g_1(x) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0$$

Table B1
Variation of $f(a)$ with a.

a	≤1.4	1.5	1.6	1.7	1.8	1.9	2	2.1	2.2	2.3	2.4	2.5	2.6	2.7	≥2.8
$f(a)$	1	0.85	0.77	0.71	0.66	0.63	0.6	0.58	0.56	0.55	0.53	0.52	0.51	0.51	0.5

$$g_2(x) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0$$

$$g_3(x) = \frac{1}{\sqrt{2}x_2 + x_1}P - \sigma \le 0$$

$$0 \le x_i \le 1, \quad i = 1, 2$$

$$l = 100\,\text{cm}, \quad P = 2\ \text{kN/cm}^2, \quad \sigma = 2\ \text{kN/cm}^2$$

B.2. Pressure vessel design problem

$$\min f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

subject to:

$$g_1(x) = -x_1 + 0.0193x$$

$$g_2(x) = -x_2 + 0.00954x_3 \le 0$$

$$g_3(x) = -\pi x_3^2x_4 - (4/3)\pi x_3^3 + 1,296,000 \le 0$$

$$g_4(x) = x_4 - 240 \le 0$$

$$0 \le x_i \le 100, \quad i = 1, 2$$

$$10 \le x_i \le 200, \quad i = 3, 4$$

B.3. Tension/compression spring design problem

$$f(x) = (x_3 + 2)x_2x_1^2$$

subject to:

$$g_1(x) = 1 - (x_2^3x_3/71,785x_1^4) \le 0$$

$$g_2(x) = (4x_2^2 - x_1x_2/12,566(x_2x_1^3 - x_1^4)) + (1/5108x_1^2) - 1 \le 0$$

$$g_3(x) = 1 - (140.45x_1/x_2^2x_3) \le 0$$

$$g_4(x) = (x_2 + x_1)/1.5 - 1 \le 0$$

$$0.05 \le x_1 \le 2.00$$

$$0.25 \le x_2 \le 1.30$$

$$2.00 \le x_3 \le 15.00$$

B.4. Welded beam design problem

$$f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2)$$

subject to:

$$g_1(x) = \tau(x) - \tau_{\max} \le 0$$

$$g_2(x) = \sigma(x) - \sigma_{\max} \le 0$$

$$g_3(x) = x_1 - x_4 \le 0$$

$$g_4(x) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \le 0$$

$$g_5(x) = 0.125 - x_1 \le 0$$

$$g_6(x) = \delta(x) - \delta_{\max} \le 0$$

$$g_7(x) = P - P_c(x) \le 0$$

$$0.1 \le x_i \le 2, \quad i = 1, 4$$

$$0.1 \le x_i \le 10, \quad i = 2, 3$$

where

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}, \quad \tau' = \frac{P}{\sqrt{2}x_1x_2}, \quad \tau'' = \frac{MR}{J}$$

$$M = P\left(L + \frac{x_2}{2}\right), \quad R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, \quad J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$\sigma(x) = \frac{6PL}{x_4x_3^2}, \quad \delta(x) = \frac{4PL^3}{Ex_3^3x_4}, \quad P_c(x) = \frac{4.013E\sqrt{(x_3^2x_4^6/36)}}{L^2} \times \left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$P = 6000\ \text{lb}, \quad L = 14\,\text{in}, \quad E = 30 \times 10^6\ \text{psi}, \quad G = 12 \times 10^6\ \text{psi}$$

$$\tau_{\max} = 13,600\ \text{psi}, \quad \sigma_{\max} = 30,000\ \text{psi}, \quad \delta_{\max} = 0.25\ \text{in}$$

B.5. Speed reducer design problem

$$\min f(x) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2)$$

subject to:

$$g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \le 0$$

$$g_2(x) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0$$

$$g_3(x) = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \le 0$$

$$g_4(x) = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \le 0$$

$$g_5(x) = \frac{\left[(745(x_4/x_2x_3))^2 + 16.9\times10^6\right]^{1/2}}{110x_6^3} - 1 \le 0$$

$$g_6(x) = \frac{\left[(745(x_5/x_2x_3))^2 + 157.5\times10^6\right]^{1/2}}{85x_7^3} - 1 \le 0$$

$$g_7(x) = \frac{x_2x_3}{40} - 1 \le 0$$

$$g_8(x) = \frac{5x_2}{x_1} - 1 \le 0$$

$$g_9(x) = \frac{x_1}{12x_2} - 1 \le 0$$

$$g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0$$

$$g_{11}(x) = \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0$$

where

$$2.6 \le x_1 \le 3.6,\quad 0.7 \le x_2 \le 0.8,\quad 17 \le x_3 \le 28,\quad 7.3 \le x_4 \le 8.3,\quad 7.3 \le x_5 \le 8.3,\quad 2.9 \le x_6 \le 3.9,\quad 5.0 \le x_7 \le 5.5$$

B.6. Gear train design problem

$$\min f(x) = ((1/6.931) - (x_3x_2/x_1x_4))^2$$

subject to:

$$12 \le x_i \le 60$$

B.7. Belleville spring design problem

$$\min f(x) = 0.07075\pi(D_e^2 - D_i^2)t$$

subject to:

$$g_1(x) = S - \frac{4E\delta_{\max}}{(1-\mu^2)\alpha D_e^2}\left[\beta\left(h - \frac{\delta_{\max}}{2}\right) + \gamma t\right] \ge 0$$

$$g_2(x) = \left(\frac{4E\delta_{\max}}{(1-\mu^2)\alpha D_e^2}\left[(h - \frac{\delta}{2})(h-\delta)t + t^3\right]\right)_{\delta=\delta_{\max}} - P_{\max} \ge 0$$

$$g_3(x) = \delta_1 - \delta_{\max} \ge 0$$

$$g_4(x) = H - h - t \ge 0$$

$$g_5(x) = D_{\max} - D_e \ge 0$$

$$g_6(x) = D_e - D_i \ge 0$$

$$g_7(x) = 0.3 - \frac{h}{D_e - D_i} \ge 0$$

where

$$\alpha = \frac{6}{\pi \ln K}\left(\frac{K-1}{K}\right)^2,\quad \beta = \frac{6}{\pi\ \ln K}\left(\frac{K-1}{\ln K} - 1\right),\quad \gamma = \frac{6}{\pi \ln K}\left(\frac{K-1}{2}\right),\quad P_{\max} = 5400\,\text{lb},$$

$$E = 30e6\ \text{psi}$$

$$\delta_{\max} = 0.2\,\text{in},\quad \mu = 0.3,\quad S = 200\,\text{KPsi},\quad H = 2\,\text{in},\quad D_{\max} = 12.01\,\text{in},\quad K = D_e/D_i,\quad \delta_1 = f(a)a,\quad a = h/t$$

Values for $f(a)$ vary as shown in Table B1.

B.8. Rolling element bearing design problem

$$\max C_d = f_c Z^{2/3} D_b^{1.8} \quad \text{if } D \le 25.4\,\text{mm}$$
$$C_d = 3.647 f_c Z^{2/3} D_b^{1.4} \quad \text{if } D \succ 25.4\,\text{mm}$$

subject to:

$$g_1(x) = \frac{\phi_0}{2\sin^{-1}(D_b/D_m)} - Z + 1 \ge 0$$

$$g_2(x) = 2D_b - K_{D\min}(D-d) \ge 0$$

$$g_3(x) = K_{D\max}(D-d) - 2D_b \ge 0$$

$$g_4(x) = \zeta B_w - D_b \le 0$$

$$g_5(x) = D_m - 0.5(D+d) \ge 0$$

$$g_6(x) = (0.5+e)(D+d) - D_m \ge 0$$

$$g_7(x) = 0.5(D - D_m - D_b) - \varepsilon D_b \ge 0$$

$$g_8(x) = f_i \ge 0.515$$

$$g_9(x) = f_o \ge 0.515$$

where

$$f_c = 37.91\left[1 + \left\{1.04\left(\frac{1-\gamma}{1+\gamma}\right)^{1.72}\left(\frac{f_i(2f_o-1)}{f_o(2f_i-1)}\right)^{0.41}\right\}^{10/3}\right]^{-0.3} \times \left[\frac{\gamma^{0.3}(1-\gamma)^{1.39}}{(1+\gamma)^{1/3}}\right]\left[\frac{2f_i}{2f_i-1}\right]^{0.41}$$

$$\phi_o = 2\pi - 2\cos^{-1}\left(\frac{[\{(D-d)/2 - 3(T/4)\}^2 + \{D/2 - T/4 - D_b\}^2 - \{d/2 + T/4\}^2]}{2\{(D-d)/2 - 3(T/4)\}\{D/2 - T/4 - D_b\}}\right)$$

$$\gamma = \frac{D_b}{D_m},\quad f_i = \frac{r_i}{D_b},\quad f_o = \frac{r_o}{D_b},\quad T = D - d - 2D_b$$

$$D = 160,\quad d = 90,\quad B_w = 30,\quad r_i = r_o = 11.033$$

$$0.5(D+d) \le D_m \le 0.6(D+d),\quad 0.15(D-d) \le D_b \le 0.45(D-d),\quad 4 \le Z \le 50,\quad 0.515 \le f_i \text{ and } f_o \le 0.6,$$

$$0.4 \le K_{D\min} \le 0.5,\quad 0.6 \le K_{D\max} \le 0.7,\quad 0.3 \le e \le 0.4,\quad 0.02 \le e \le 0.1,\quad 0.6 \le \zeta \le 0.85.$$

References

[1] K.S. Lee, Z.W. Geem, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, Computer Methods in Applied Mechanics and Engineering 194 (2005) 3902–3933.
[2] J. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, 1975.
[3] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.

[4] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 1995, pp. 1942–1948.
[5] S. Kirkpatrick, C. Gelatt, M. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[6] C.A.C. Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Computer Methods in Applied Mechanics and Engineering 191 (2002) 1245–1287.
[7] S. Areibi, M. Moussa, H. Abdullah, A comparison of genetic/memetic algorithms and other heuristic search techniques, in: ICAI, Las VeGAs, Nevada, 2001.
[8] E. Elbeltagi, T. Hegazy, D. Grierson, Comparison among five evolutionary-based optimization algorithms, Advanced Engineering Informatics 19 (2005) 43–53.
[9] H. Youssef, S.M. Sait, H. Adiche, Evolutionary algorithms, simulated annealing and tabu search: a comparative study, Engineering Applications of Artificial Intelligence 14 (2001) 167–181.
[10] L. Giraud-Moreau, P. Lafon, Comparison of evolutionary algorithms for mechanical design components, Engineering Optimization 34 (2002) 307–322.
[11] P. Chootinan, A. Chen, Constraint handling in genetic algorithms using a gradient-based repair method, Computers & Operations Research 33 (2006) 2263–2281.
[12] K.Z. Tang, T.K. Sun, J.Y. Yang, An improved genetic algorithm based on a novel selection strategy for nonlinear programming problems, Computers and Chemical Engineering 35 (2011) 615–621.
[13] Q. Yuan, F. Qian, A hybrid genetic algorithm for twice continuously differentiable NLP problems, Computers and Chemical Engineering 34 (2010) 36–41.
[14] A. Amirjanov, The development of a changing range genetic algorithm, Computer Methods in Applied Mechanics and Engineering 195 (2006) 2495–2508.
[15] I.C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Information Processing Letters 85 (2003) 317–325.
[16] Q. He, L. Wang, An effective co-evolutionary particle swarm optimization for engineering optimization problems, Engineering Applications of Artificial Intelligence 20 (2006) 89–99.
[17] H.M. Gomes, Truss optimization with dynamic constraints using a particle swarm algorithm, Expert Systems with Applications 38 (2011) 957–968.
[18] L.D.S. Coelho, Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems, Expert Systems with Applications 37 (2010) 1676–1683.
[19] Q. He, L. Wang, A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization, Applied Mathematics and Computation 186 (2007) 1407–1422.
[20] Y. Wang, Z. Cai, A hybrid multi-swarm particle swarm optimization to solve constrained optimization problems, Frontiers of Computer Science 3 (1) (2009) 38–52.
[21] Y. Wang, Z. Cai, G. Guo, Y. Zhou, Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems, IEEE Transactions on Systems Man and Cybernetics Part B 37 (3) (2007) 560–575.
[22] Y. Wang, Z. Cai, A dynamic hybrid framework for constrained evolutionary optimization, IEEE Transactions on Systems Man and Cybernetics Part B 42 (1) (2012) 203–217.
[23] Y. Wang, Z. Cai, Combining multiobjective optimization with differential evolution to solve constrained optimization problems, IEEE Transactions on Evolutionary Computation 16 (1) (2012) 117–134.
[24] Y. Wang, Z. Cai, Constrained evolutionary optimization by means of $(\mu+\lambda)$-differential evolution and improved adaptive trade-off model, Evolutionary Computation 19 (2) (2011) 249–285.
[25] R.V. Rao, V.J. Savsani, Mechanical Design Optimization Using Advanced Optimization Techniques, Springer-Verlag, London, 2012.
[26] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems, Computer-Aided Design 43 (2011) 303–315.
[27] R.V. Rao, V. Patel, An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems, International Journal of Industrial Engineering Computations 3 (2012) 535–560.
[28] R.V. Rao, V. Patel, Multi-objective optimization of heat exchangers using a modified teaching-learning-based-optimization algorithm, Applied Mathematical Modelling (2012), http://dx.doi.org/10.1016/j.apm.2012.03.043.
[29] T. Ray, K.M. Liew, Society and civilization: an optimization algorithm based on the simulation of social behavior, IEEE Transactions on Evolutionary Computation 7 (2003) 386–396.
[30] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm for optimization of truss structures with discrete variables, Computers & Structures 102–103 (2012) 49–63.
[31] E.M. Montes, C.A.C. Coello, An empirical study about the usefulness of evolution strategies to solve constrained optimization problems, International Journal of General Systems 37 (2008) 443–473.
[32] A. Kaveh, S. Talatahari, A particle swarm ant colony optimization for truss structures with discrete variables, Journal of Constructional Steel Research 65 (2009) 1558–1568.
[33] J. Bracken, G.P. Mccormick, Selected Applications of Nonlinear Programming, John Wiley & Sons, New York, 1968.
[34] A. Homaifar, C.X. Qi, S.H. Lai, Constrained optimization via genetic algorithms, Simulation 62 (1994) 242–253.
[35] D.B. Fogel, A comparison of evolutionary programming and genetic algorithms on selected constrained optimization problems, Simulation 64 (1995) 397–404.
[36] B. Tessema, G.G. Yen, A self adaptive penalty function based algorithm for constrained optimization, IEEE Transactions on Evolutionary Computation (2006) 246–253.
[37] R. Becerra, C.A.C. Coello, Cultured differential evolution for constrained optimization, Computer Methods in Applied Mechanics and Engineering 195 (2006) 4303–4322.
[38] E. Mezura-Montes, C.A.C. Coello, A simple multimembered evolution strategy to solve constrained optimization problems, IEEE Transactions on Evolutionary Computation 9 (2005) 1–17.
[39] S. Koziel, Z. Michalewicz, Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization, IEEE Transactions on Evolutionary Computation 7 (1999) 19–44.
[40] S. Ben Hamida, M. Schoenauer, ASCHEA: new results using adaptive segregational constraint handling, IEEE Transactions on Evolutionary Computation (2002) 884–889.
[41] H. Liu, Z. Cai, Y. Wang, Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization, Applied Soft Computing 10 (2010) 629–640.
[42] T.P. Runarsson, Y. Xin, Stochastic ranking for constrained evolutionary optimization, IEEE Transactions on Evolutionary Computation 4 (2000) 284–294.
[43] J. Lampinen, A constraint handling approach for the differential evolution algorithm, IEEE Transactions on Evolutionary Computation (2002) 1468–1473.
[44] L. Wang, L.P. Li, An effective differential evolution with level comparison for constrained engineering design, Structural and Multidisciplinary Optimization 41 (2010) 947–963.
[45] M. Zhang, W. Luo, X. Wang, Differential evolution with dynamic stochastic selection for constrained optimization, Information Sciences 178 (2008) 3043–3074.
[46] Y. Wang, Z. Cai, Y. Zhou, Z. Fan, Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint handling technique, Structural and Multidisciplinary Optimization 37 (2009) 395–413.
[47] T.P. Runarsson, Y. Xin, Search biases in constrained evolutionary optimization, IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews 35 (2005) 233–243.
[48] T. Takahama, S. Sakai, Constrained optimization by applying the α; constrained method to the nonlinear simplex method with mutations, IEEE Transactions on Evolutionary Computation 9 (2005) 437–451.
[49] D. Karaboga, B. Basturk, Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems, LNAI, vol. 4529, Springer-Verlag, Berlin, 2007, pp. 789–798.
[50] K. Deb, An efficient constraint handling method for genetic algorithms, Computer Methods in Applied Mechanics and Engineering 186 (2000) 311–338.
[51] C.A.C. Coello, R.L. Becerra, Efficient evolutionary optimization through the use of a cultural algorithm, Engineering Optimization 36 (2004) 219–236.
[52] E. Zahara, Y.T. Kao, Hybrid Nelder-Mead simplex search and particle swarm optimization for constrained engineering design problems, Expert Systems with Applications 36 (2009) 3880–3886.
[53] A.R. Hedar, M. Fukushima, Derivative-free filter simulated annealing method for constrained continuous global optimization, Journal of Global Optimization 35 (2006) 521–549.
[54] Z. Michalewicz, Genetic algorithms, numerical optimization, and constraints, in: L. Esheman (Ed.), Proceedings of the Sixth International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, 1995, pp. 151–158.
[55] A.K. Renato, C. Leandro Dos Santos, Coevolutionary particle swarm optimization using gaussian distribution for solving constrained optimization problems, IEEE Transactions on Systems Man and Cybernetics Part B: Cybernetics 36 (2006) 1407–1416.
[56] A.E.M. Zavala, A.H. Aguirre, E.R.V. Diharce, Constrained optimization via evolutionary swarm optimization algorithm (PESO), in: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, 2005, pp. 209–216.
[57] F.Z. Huang, L. Wang, Q. He, An effective co-evolutionary differential evolution for constrained optimization, Applied Mathematics and Computation 186 (1) (2007) 340–356.
[58] B.K. Kannan, S.N. Kramer, An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, Journal of Mechanical Design 116 (1994) 405–411.
[59] C.A.C. Coello, Use of a self-adaptive penalty approach for engineering optimization problems, Computers in Industry 41 (2000) 113–127.
[60] C.A.C. Coello, E. Mezura Montes, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection, Advanced Engineering Informatics 16 (2002) 193–203.
[61] K. Parsopoulos, M. Vrahatis, Unified particle swarm optimization for solving constrained engineering optimization problems Adv. Nat. Computation, LNCS, vol. 3612, Springer-Verlag, Berlin, 2005, pp. 582–591.
[62] B. Akay, D. Karaboga, Artificial bee colony algorithm for large-scale problems and engineering design optimization, Journal of Intelligent Manufacturing (2010), http://dx.doi.org/10.1007/s10845-010-0393-4.
[63] E. Mezura-Montes, C.A.C. Coello, Useful infeasible solutions in engineering optimization with evolutionary algorithms, in: MICAI 2005: Lect. Notes Artif. Int., vol. 3789, 2005, pp. 652–662, http://dx.doi.org/10.1007/11579427_66.
[64] J.S. Arora, Introduction to Optimum Design, McGraw-Hill, New York, 1989.
[65] C.A.C. Coello, Constraint-handling using an evolutionary multiobjective optimization technique, Civil Engineering and Environmental Systems 17 (2000) 319–346.
[66] J.K. Kuang, S.S. Rao, L. Chen, Taguchi-aided search method for design optimization of engineering systems, Engineering Optimization 30 (1998) 1–23.

[67] E. Mezura-Montes, J. Velazquez-Reyes, C.A.C. Coello, Modified differential evolution for constrained optimization, in: Evol. Comput., CEC 2006, IEEE Congress, 2006, pp. 25–32.
[68] E. Montes-Montes, C.A.C. Coello, J. Velazquez-Reyes, Increasing successful offspring and diversity in differential evolution for engineering design, in: Proceedings of the Seventh International Conference on Adaptive Computing in Design and Manufacture, 2006, pp. 131–139.
[69] C.A.C. Coello, Treating constraints as objectives for single-objective evolutionary optimization, Engineering Optimization 32 (3) (2000) 275–308.
[70] K. Deb, M. Goyal, Optimizing engineering designs using a combined genetic search, in: L.J. Eshelman (Ed.), Proceedings of the Sixth International Conference in Generic Algorithms, University of Pittsburgh, Morgan Kaufmann Publishers, San Mateo, CA, 1995, pp. 521–528.
[71] J.N. Siddall, Optimal Engineering Design, Principles and Applications, Marcel Dekker, New York, 1982.
[72] S. Gupta, R. Tiwari, B.N. Shivashankar, Multi-objective design optimization of rolling bearings using genetic algorithm, Mechanism and Machine Theory 42 (2007) 1418–1443.

Please cite this article in press as: A. Sadollah, et al., Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.11.026