Paper 52

©Civil-Comp Press, 2012
Proceedings of the Eighth International Conference
on Engineering Computational Technology,
B.H.V. Topping, (Editor),
Civil-Comp Press, Stirlingshire, Scotland

Improving the Big Bang-Big Crunch Algorithm for Optimum Design of Steel Frames

O. Hasançebi and S. Kazemzadeh Azad
Department of Civil Engineering
Middle East Technical University, Ankara, Turkey

Abstract

This paper presents an improved version of the big bang-big crunch (BB-BC) algorithm namely exponential BB-BC algorithm (EBB-BC) for optimum design of steel frames according to ASD-AISC provisions. It is shown that the standard version of the algorithm sometimes is unable to provide reasonable solutions for problems from discrete design optimization of steel frames. Therefore, by investigating the shortcomings of the BB-BC algorithm, it is aimed to enhance the algorithm for solving complicated steel frame optimization problems. In order to evaluate the performance of the proposed algorithm, the optimization results attained using the EBB-BC algorithm are compared to those of other well known metaheuristics. The numerical results demonstrate the efficiency and robustness of the proposed algorithm in practical design optimization of steel frames.

Keywords: steel frames, practical design, metaheuristics, big bang-big crunch algorithm, discrete optimization.

1 Introduction

In the recent years, drawbacks of traditional structural optimization techniques namely mathematical programming [1] and optimality criteria [2, 3] methods, such as their gradient based formulations and inefficiency in handling discrete design variables, have led to an increasing tendency to the non-traditional stochastic search algorithms or metaheuristics. Typically, metaheuristic algorithms such as genetic algorithm (GA), particle swarm optimization (PSO), ant colony optimization (ACO), etc., borrowing their working principles from natural phenomena [4], follow non-deterministic search strategies to locate the optimum solutions of complicated optimization problems. Reputation of metaheuristics in structural optimization can be attributed to their superior solutions, robust performances, ease of implementation/use, independency to gradient information, and capability of handling both continuous and discrete design variables. However, required high computational effort can be considered as the main shortcoming of metaheuristic algorithms. Today's trend is to provide a remedy for this problem by developing more efficient approaches which need less computational time to locate the optima. Hopefully, this burden will be removed through development of high performance computational systems in the near future. The state-of-the-art reviews of metaheuristic algorithms considering their applications in the structural optimization problems are provided in [5, 6].

In 2006 Erol and Eksin [7] proposed a novel optimization algorithm so called big bang–big crunch (BB-BC) algorithm as an efficient metaheuristic optimization method based on the BB-BC theory of the universe evolution. Due to its efficient performance and ease of use, BB-BC algorithm quickly became one of the most popular algorithms of the recent years and many researchers employed it for solving practical optimization problems so far [8-17]. Afshar and Motaei [8] used the BB-BC algorithm to find the optimal solution of large scale reservoir operations problem. Tang et al. [9] used the algorithm for parameter estimation in structural systems. Camp [10] seems to be the first who applied BB-BC algorithm for optimum design of skeletal structures. In his work the optimum design of planar and spatial truss structures was performed using a modified variant of the algorithm. To enhance the efficiency of the BB–BC algorithm, Camp [10] proposed a weighting parameter to control the influence of both the center of mass and the current global best solution on new candidate solutions. Additionally, a multiphase search strategy was utilized to increase the quality of solutions. The study revealed the efficiency of the BB–BC algorithm in comparison to the GA, PSO, and ACO based techniques previously reported in the literature. Kaveh and Abbasgholiha [11] used the Camp's strategy of generating new candidate solutions for optimum design of planar steel sway frames. Recently, Lamberti and Pappalettere [12] introduced an improved BB-BC algorithm for weight minimization of truss structures and reported promising results using four benchmark truss optimization instances. Kaveh and Talatahari [13-15] developed hybrid variants of the BB-BC algorithm for design optimization of different types of skeletal structures. In addition, Kaveh et al. [16] employed a hybrid BB-BC algorithm for optimum seismic design of gravity retaining walls. A recent performance evaluation of BB-BC algorithm carried out by Kazemzadeh Azad et al. [17] revealed the efficiency of this method in benchmark engineering optimization problems.

Although the BB-BC algorithm has been utilized for solving a wide range of engineering optimization problems so far, there are only a few studies in the literature including the BB-BC algorithm based optimization of steel frame structures [11, 15]. Since the available studies cover only the planar steel frames, the present study aims to provide a more realistic application of the BB-BC algorithm through a three dimensional practical steel frame example. In spite of good reputation of the BB-BC algorithm in solving design optimization problems, the present study reveals the shortcomings of the standard algorithm in providing acceptable solutions for practical steel frame design optimization problems. Hence, an exponential BB-BC algorithm (EBB-BC) is proposed to enhance the capability of the standard algorithm in locating global optimum or a reasonable near optimum solution. The remaining sections of the paper are as follows. The second section includes a statement of the steel frame optimization problem based on discrete design variables. In the third section, the BB-BC algorithm as well as the proposed formulation to enhance its performance in design optimization of steel frames are discussed. The performance evaluation of the considered methodologies is carried out in the fourth section. The last section of the study covers the concluding remarks.

2 Practical design optimization of steel frames

In real practice the frame members are typically adopted from a set of available steel sections which yields a discrete optimization problem. For a steel frame consisting of N_m members grouped into N_d design groups, the design optimization problem according to ASD-AISC [18] code becomes as follows. The optimization objective is to find an integer vector **I** (Equation (1)) which reflects the sequence numbers corresponding to a table of available standard steel sections

$$\mathbf{I}^T = [I_1, I_2, ..., I_{N_d}] \tag{1}$$

to produce a vector of cross-sectional areas **A** (Equation (2)) for N_m frame members

$$\mathbf{A}^T = [A_1, A_2, ..., A_{N_m}] \tag{2}$$

such that **A** minimizes the total weight (W) of the structure

$$W = \sum_{m=1}^{N_m} \rho_m L_m A_m \tag{3}$$

where ρ_m, L_m, A_m are unit weight, length, and cross-sectional area of the *m-th* member, respectively. Detailed formulations of the optimization problem are given in Ref. [22].

3 Optimization method

Big bang-big crunch optimization method has first appeared in Erol and Eksin's study [7]. It is emerged from the Big Bang and Big Crunch theories of the universe evolution. As its name implies, the method is based on the continuous application of two successive stages, namely big bang and big crunch phases. During big bang phase, new solution candidates, which are the parameters that affect the fitness function, are randomly generated around a "center of mass", that is later calculated in the big crunch phase with respect to their fitness values.

3.1 Standard BB-BC algorithm

The steps in the implementation of a standard BB-BC algorithm can be outlined as follows.

Step1. Initial population: Form an initial population by randomly spreading individuals (solutions) over all the search space (first big bang) in a uniform manner. This step is applied once.

Step 2. Evaluation: The initial population is evaluated, where structural analyses of all individuals are performed with the set of steel sections selected for design variables, and force and deformation responses are obtained under the loads. The objective function values of the feasible individuals that satisfy all problem constraints are directly calculated from Equation (3). However, infeasible individuals that violate some of the problem constraints are penalized using an external penalty function approach, and their objective function values are calculated according to Equation (4) [20].

$$\phi = W\left[1 + p\left(\sum_i c_i\right)\right] \tag{4}$$

In Equation (4), ϕ is the constrained objective function value, c_i is the *i*-th problem constraint and p is the penalty coefficient used to tune the intensity of penalization as a whole. This parameter is generally set to an appropriate static value of $p = 1$. The fitness scores of the individuals are then calculated by taking the inverse of their objective function values (i.e. fitness = $1/W$ or $1/\phi$ for feasible and infeasible solutions, respectively). The fitness scores are assigned as the mass values for the individuals.

Step 3. Big-crunch phase: Calculate the "center of mass" by taking the weighted average using the coordinates (design variables) and the mass values of every single individual or choose the fittest individual among all as their center of mass (the latter approach is adopted in the present study).

Step 4. Big bang phase: Generate new individuals by using normal distribution (big bang phase). For a continuous variable optimization problem, Equation (5) is used at each iteration to generate new solutions around the center of mass.

$$x_i^{new} = x_i^c + \alpha r_i \frac{(x_i^{\max} - x_i^{\min})}{k} \tag{5}$$

where x_i^c is the value of *i*-th continuous design variable in the fittest individual, $x_i^{\min}$ and $x_i^{\max}$ are the lower and upper bounds on the value of *i*-th design variable, respectively, r_i is a random number generated according to a standard normal distribution with mean (μ) zero and standard deviation (σ) equal to one, k is the iteration number, and α is a constant.

In the present study, however, a discrete list of ready sections is used for sizing members in a structure. Hence, Equation (6) is employed instead to round off the real values to nearest integers representing the sequence number of ready sections in a given section list.

$$I_i^{new} = I_i^c + round\left[\alpha r_i \frac{(I_i^{\max} - I_i^{\min})}{k}\right] \tag{6}$$

where I_i^c is the value of i-th discrete design variable in the fittest individual, and $I_i^{\min}$ and $I_i^{\max}$ are its lower and upper bounds respectively.

Step 5. Elitism: Keep the fittest individual found so far in a separate place or as a member of the population.

Step 6. Termination: Go to Step 2 until a stopping criterion is satisfied, which can be imposed as a maximum number of iterations or no improvement of the best design over a certain number of iterations.

It is important to mention that metaheuristic search techniques offer a general solution methodology for solving a wide range of optimization problems from different disciplines. On the other hand, each optimization problem has unique features of its own, and in most cases a problem-wise reformulation is necessary to achieve the best performance of the algorithm for a particular class of problems. In the following section, the observed deficiencies of the standard BB-BC algorithm in the discrete design optimization of frame structures are discussed in detail. The poor performance of the standard algorithm is attributed to ineffective manipulations of the two search parameters; namely search dimensionality and step size.

3.2 Search dimensionality and step size

Search dimensionality (SD) is defined as the number of design variables that are perturbed to generate a new design through Equation (6). Perhaps a more general term to quantify the degree of search dimensionality irrespective of problem size is search dimensionality rate (SDR), which is computed by proportioning the number of perturbed design variables to the total number of design variables used in a problem, Equation (7).

$$SDR = \frac{N_p}{N_d} \tag{7}$$

Accordingly, the average search dimensionality rate for a population, $(SDR)_{ave}$, is obtained by averaging *SDR* values of all designs (Equation (8)), where $(SDR)_j$ is search dimensionality rate for design *j* and N_{pop} is the population size referring to the total number of designs in the population.

$$(SDR)_{ave} = \frac{\sum (SDR)_j}{N_{pop}} \tag{8}$$

For continuous optimization problems $(SDR)_{ave}$ will always have a value equal or close to 1.0, since all variables -except those on the value bounds- are subjected to perturbation during generation of a new design. That is to say an N-dimensional search is performed by the algorithm at any time during the search process. However, for discrete optimization problems some design variables will remain unchanged owing to the fact that the second term is rounded off to zero when the random number r_i generated by normal distribution is too small, which implicitly drive $(SDR)_{ave}$ to low values especially when the iteration number *k* increases.

On the other hand, the step size for a single design variable is equal to $I_i^{new} - I_i^c$. Hence one can define average step size for an individual design and entire population at any iteration as formulated in Equations (9) and (10).

$$(SS)_{j,ave} = \frac{\sum (I_i^{new} - I_i^c)_j}{N_d} \tag{9}$$

$$(SS)_{ave} = \frac{\sum (SS)_{j,ave}}{N_{pop}} \tag{10}$$

where $(SS)_{j,ave}$ and $(SS)_{ave}$ are average step size for *j*-th design and entire population respectively.

3.3 Numerical investigations with BB-BC algorithm

While using the standard BB-BC algorithm for discrete optimization, it is noted that both average search dimensionality rate $(SDR)_{ave}$ and average step size $(SS)_{ave}$ parameters will tend to become zero after a certain number of iterations. Once this happens, no new designs are generated; i.e. the subsequent designs simply replicate the former ones. As a remedy to this situation, the following routine is integrated into the algorithm to make sure that a new design will differ from the former one at least by one variable.

```
Set σ := 1.0;
Quitloop:= False;
Repeat
    Generate I^new from I^c using Equation (6)
    If I^new = I^c then Quitloop := true
              else  σ := σ + 1.0
Until Quitloop;
```

When applying this routine, if all the design variables in a new design remain unchanged after applying Equation (6), the generation process is iterated in the same way by increasing the standard deviation of normal distribution σ by one every time, and this is repeated until a different design is produced. Apparently, the increased standard deviation facilitates occurrence of larger step sizes and increases probability of design variable change.

The typical results obtained from numerical investigations with BB-BC algorithm on discrete frame-sizing optimization problems are reflected in Figures 1a and 1b, which show variations of $(SDR)_{ave}$ and $(SS)_{ave}$ parameters in the course of search process. It is noted that the average search dimensionality of a population in the first iterations is generally in the order of 0.9, which results in extreme changes in the individuals. Although this provides a diverse population, this amount of diversity is more likely to result in convergence difficulties in case of discrete design optimization of frame structures. It is stated in Hasançebi [21] that a useful starting value of $(SDR)_{ave}$ will be in the range of [0.25, 0.50] based on experiments with an evolution strategies integrated search process. On the other hand, towards the later stages, search is implemented mostly in a single direction per design. Although a somewhat decreased search dimensionality towards the latest stage might be useful in the sense that it boosts more exploitative search in the design space, the search capability of the algorithm is significantly restricted when it is limited too much.

3.4 The reformulations of BB-BC algorithm

Noticing the drawbacks of the standard algorithm discussed above, the use of Equation (11) is proposed in place of Equation (6) to improve the efficiency of the BB-BC algorithm in discrete design optimization of frame structures.

$$I_i^{new} = I_i^c + round\left[\alpha d_i^3 \frac{(I_i^{\max} - I_i^{\min})}{k}\right] \tag{11}$$

The difference between the standard formulation and this reformulation is that the third power of a random number (d_i) generated using a chosen distribution is favored in the new formulation for producing the new candidate solutions around the center of mass. Recently, the authors demonstrated the satisfactory performance of the third power formulation (i.e. Equation (11)), using a normally distributed random number for parameter d_i, in discrete design optimization of truss structures through a modified BB-BC (MBB-BC) algorithm [23].

It should be noted that the rationale behind the use of third power formulation is to achieve a satisfactory trade-off or compromise between the following two conflicting requirements needed to eliminate the shortcomings of the standard formulation: (i) diminishing search dimensionality in the beginning of the search process and increasing it somewhat towards the latest stage and (ii) enabling large step size from time to time at later optimization stages to facilitate design transitions to new design regions and thereby preventing entrapment of the search in local optima. In fact, at times when a random number is sampled at values below 1, taking the third power of this number makes is even much smaller, which helps to fulfill the first requirement. On the other hand, at times when a normally distributed number above 1 is sampled, it is amplified to large values by taking its third power, helping to satisfy the second requirement.

In the present study the use of an exponentially distributed random number for parameter d_i is proposed and investigated in discrete design optimization of steel frame structures. It is worth mentioning that the probability density function for a standard exponential distribution is as follows:

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{12}$$

where λ is a real, positive constant. The mean and variance of the exponential distribution are $1/\lambda$ and $1/\lambda^2$, respectively. In the study, λ is set to 1.0.

During the optimization of steel frames it is observed that similar to MBB-BC algorithm, however more effectively, the exponential version i.e. EBB-BC algorithm eliminates the aforementioned two shortcomings of the standard formulation through adjusting the search dimensionality during the optimization process as well as providing occasional large step size values during the optimization process.

Although the proposed refinement looks rather simple, it results in appreciable improvement in the performance of the standard algorithm. Figures 1c to 1f show typical variations of $(SDR)_{ave}$ and $(SS)_{ave}$ parameters during a search captured during numerical investigations with the MBB-BC and EBB-BC algorithms. It is observed from Figures 1c and 1e that using the MBB-BC and EBB-BC algorithms the average search dimensionality ratio in the initial iterations is mostly in the order of 0.5, which leads to an appropriate diversity in the population and provides a more suitable search mechanism for discrete design optimization problems. Furthermore, the EBB-BC algorithm usually provides greater $(SDR)_{ave}$ values in comparison to the standard version as well as the MBB-BC algorithm, which results in a more efficient exploration due to increased dimensionality of the search during the optimization process.

The other advantage of the MBB-BC and EBB-BC variants lies in their ability to accommodate large step sizes even in the last iterations of optimization. This is clearly seen in Figures 1d and 1f. These occasional large step sizes are quite useful for steering the search towards new design regions when the search gets stuck in local optima. This characteristic of the refined formulations provide an efficient mechanism to avoid local optima while the standard algorithm is likely to get trapped in local optima due to the constant small step size values (mostly an average step size of 1) of last iterations. Numerical investigations demonstrate the superiority of the EBB-BC algorithm on the MBB-BC version in adjusting an efficient balance between the exploration and exploitation properties of the standard algorithm which results in a more effective search in the design space towards the optimum design.

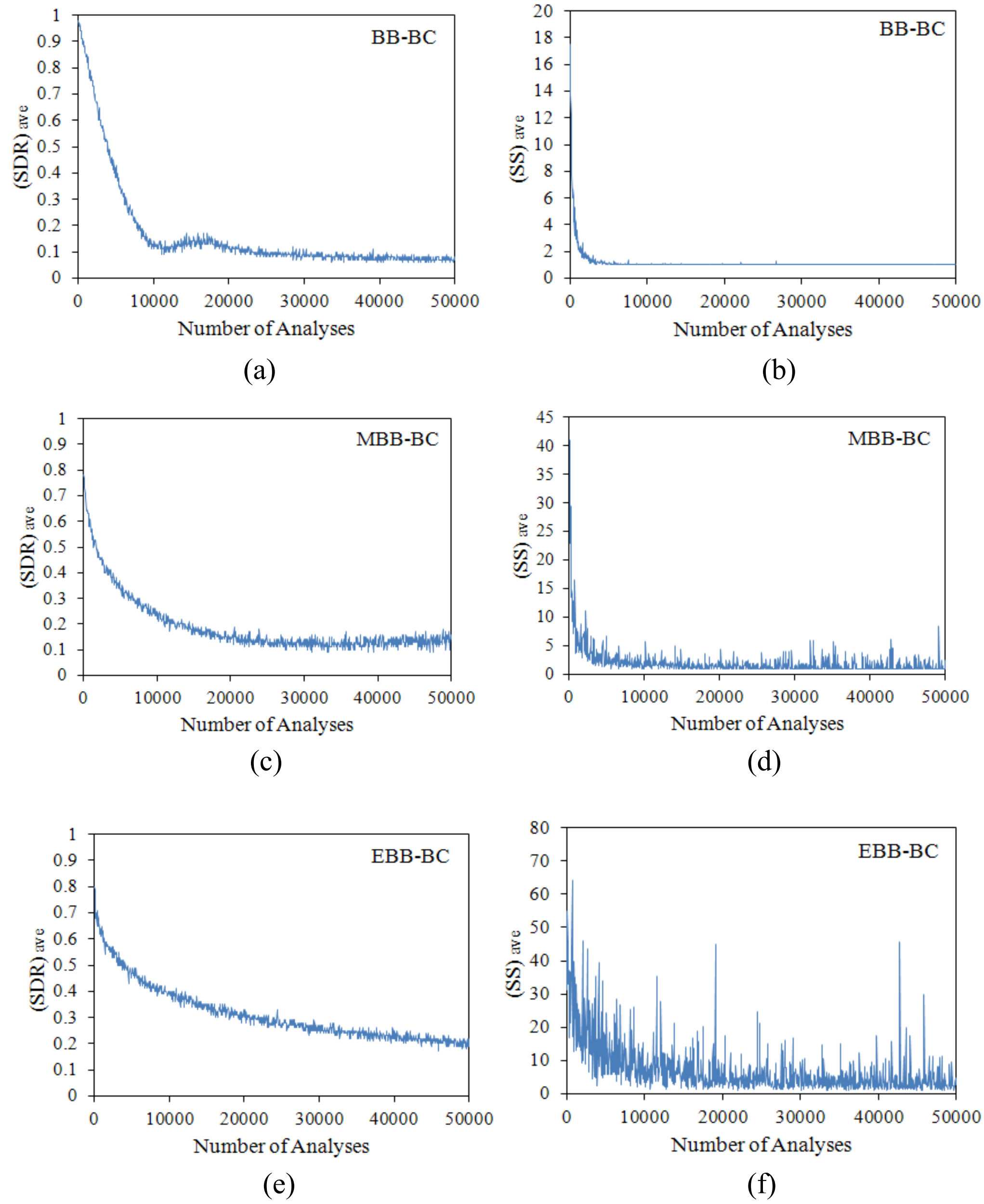

Figure 1: $(SDR)_{ave}$ and $(SS)_{ave}$ variations in a typical run of: (a, b) BB-BC, (c, d) MBB-BC, and (e, f) EBB-BC

4 Design example: 132-member unbraced steel frame

In this section, a 132-member unbraced steel frame, which has been previously studied in the literature, is adopted to investigate the efficiency of BB-BC algorithm and its two variants i.e. MBB-BC and EBB-BC. The optimization results attained based on the MBB-BC and EBB-BC approaches are compared to those achieved using the standard version of the BB-BC algorithm as well as other metaheuristic techniques. Additionally, the effect of new formulations on the average mutation probability and average step size is presented. For a fair comparison of results, the maximum number of structural analyses is limited to the previously reported values in the literature. Thus, the maximum number of structural analyses for the investigated example is set to 50,000. Here, the value of parameter α in Equations (6) and (11) is taken as 0.25 and a population size of 50 is used for the BB-BC, MBB-BC and EBB-BC algorithms. The material properties of steel used for the example is as follows: modulus of elasticity (E) = 29,000 ksi (203,893.6 MPa) and yield stress (F_y) = 36 ksi (253.1 MPa).

The design example depicted in Figure 3 is a three dimensional unbraced (swaying) steel frame composed of 70 joints and 132 members that are grouped into 30 independent sizing variables (Figure 3b) to satisfy practical fabrication requirements. The columns are adopted from the complete W-shape profile list consisting of 297 ready sections, whereas a discrete set of 171 economical sections selected from W-shape profile list based on area and inertia properties is used to size beam members. Here, both gravity and lateral loads are considered in designing the structure. Gravity loads (G) consisting dead, live and snow loads are calculated according to ASCE 7-05 [19] based on the following design values: a design dead load of 60.13 lb/ft^2 (2.88 kN/m^2), a design live load of 50 lb/ft^2 (2.39 kN/m^2), and a ground snow load of 25 lb/ft^2 (1.20 kN/m^2). This yields the uniformly distributed loads on the outer and inner beams of the roof and floors given in Table 1. As for lateral forces, earthquake loads (E) are considered. These loads are calculated based on the equivalent lateral force procedure outlined in ASCE 7-05 [19], resulting in the values given in Table 1 that are applied at the center of gravity of each story as joint loads. Gravity (G) and earthquake (E) loads are combined under two loading conditions for the frame: (i) 1.0G + 1.0E (in x-direction), and (ii) 1.0G + 1.0E (in y-direction). The combined stress, stability and geometric constraints are imposed as explained in Ref. [22]. The joint displacements in x and y direction are limited to 1.53 in (3.59 cm) which is obtained as height of frame/400. Additionally, story drift constraints are applied to each storey of the frame which is equal to height of each story/400.

Here, the BB-BC, MBB-BC and EBB-BC algorithms are tested to minimize the weight of the 132-member steel frame. In Table 2 the minimum weight designs of the structure obtained by the BB-BC, MBB-BC, and EBB-BC algorithms are compared to the previously reported results by Hasançebi [22] based on different metaheuristic techniques. The EBB-BC algorithm produces a design weight of 60804.31 kg (134050.55 lb) for the frame which is the best solution of this problem reported so far. Relatively higher design weights have been obtained for the frame with other metaheuristic algorithms; namely 62993.55 kg (138874.67 lb) by iSA, 64733.69 kg (142710.96) by TS, 64926.17 kg (143135.29) by HS and 63056.72 kg (139016.28 lb) by MBB-BC. The BB-BC algorithm shows a poor performance and produces a final design weight of 87468.21 kg (192834.39 lb). Such a significant difference between the results clearly demonstrates the usefulness of the proposed refinement on the performance of the standard algorithm.

The variations of the best feasible design obtained so far in the search processes with BB-BC, MBB-BC and EBB-BC algorithms are plotted in Figure 2. Figures 4a to 4f show the variations of the $(SDR)_{ave}$ and $(SS)_{ave}$ parameters in the implementations of the algorithms. According to the results, EBB-BC algorithm follows the most successful strategy in searching the optimum design by adjusting the balance between exploration and exploitation. In comparison to BB-BC and MBB-BC algorithms, EBB-BC makes greater changes in the $(SS)_{ave}$ values while decreasing the $(SDR)_{ave}$ parameter in a reasonable way resulting in the minimum weight design of the 132-member steel frame.

Gravity Loads				
Beam Type	Uniformly Distributed Load			
	Outer Span Beams		Inner Span Beams	
	(lb/ft)	(kN/m)	(lb/ft)	(kN/m)
Roof beams (Dead + Snow Loads)	1011.74	14.77	1193.84	17.42
Floor beams (Dead + Live Loads)	1468.40	21.49	1732.70	25.29
Lateral Loads				
Floor Number	Earthquake Design Load			
	(kips)		(kN)	
1	6.57		29.23	
2	12.43		55.28	
3	18.52		82.35	
4	24.76		110.15	

Table 1: The gravity and lateral loading on 132-member space steel frame

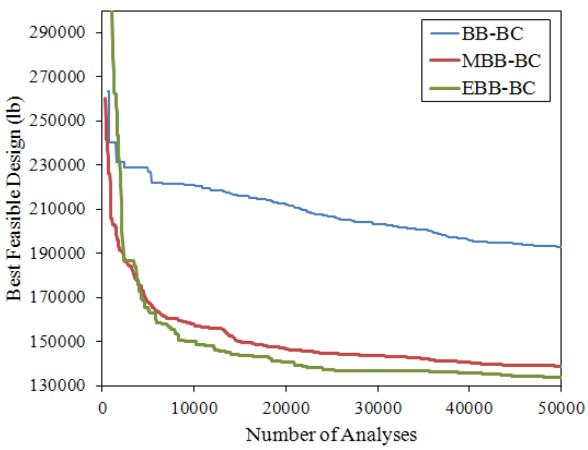

Figure 2: Optimization histories for 132-member space steel frame

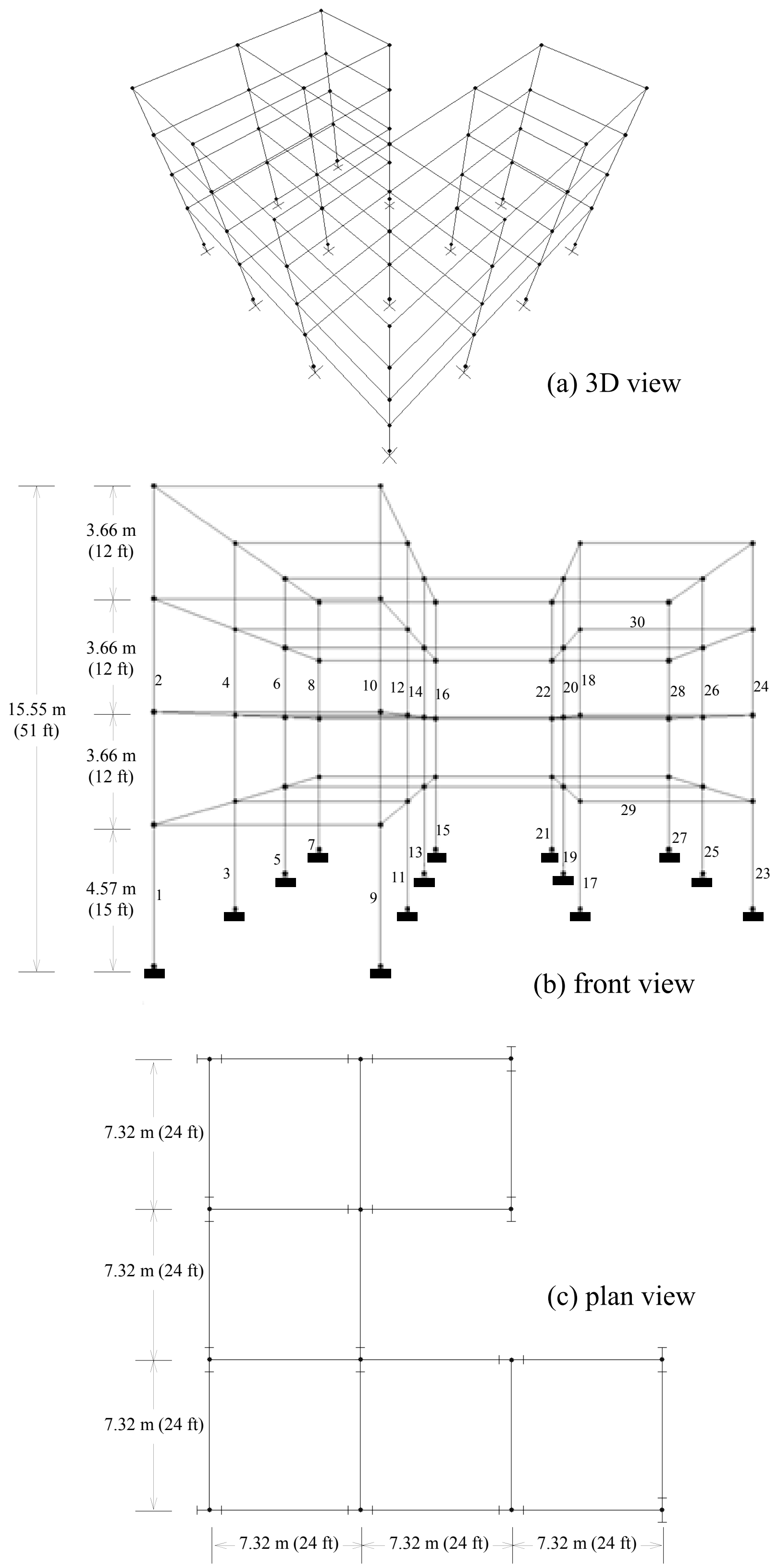

Figure 3: 132-member space steel frame (a) 3D view, (b) front view, (c) plan view

Sizing variables	iSA	TS	HS	BB-BC	MBB-BC	EBB-BC
1	W8X35	W8X31	W14X53	W24X176	W12X58	W10X33
2	W18X86	W12X65	W12X120	W21X132	W14X109	W12X79
3	W12X79	W27X129	W30X48	W27X336	W10X100	W40X167
4	W18X65	W8X58	W16X77	W24X279	W10X54	W12X65
5	W12X65	W12X79	W18X119	W14X193	W12X96	W14X120
6	W27X161	W12X106	W24X104	W14X109	W14X90	W14X109
7	W24X117	W18X97	W30X148	W12X87	W36X182	W14X99
8	W10X54	W8X58	W10X68	W27X94	W12X65	W14X90
9	W18X86	W12X72	W18X158	W30X292	W18X130	W10X100
10	W12X96	W14X90	W12X120	W18X283	W14X90	W12X106
11	W10X60	W36X135	W36X150	W10X49	W12X58	W33X152
12	W10X49	W10X49	W16X67	W21X62	W30X99	W12X53
13	W12X87	W12X96	W10X112	W18X311	W44X224	W14X90
14	W12X50	W10X49	W24X117	W33X141	W40X192	W36X160
15	W24X55	W24X55	W18X40	W18X40	W16X40	W18X40
16	W24X55	W10X33	W14X61	W12X210	W8X35	W12X53
17	W12X58	W18X76	W12X65	W16X67	W12X65	W21X111
18	W12X67	W21X83	W18X119	W12X65	W12X96	W12X65
19	W12X40	W8X40	W14X82	W14X211	W12X65	W14X43
20	W10X49	W14X61	W18X86	W14X211	W12X65	W10X60
21	W12X72	W18X76	W14X90	W40X277	W12X72	W12X106
22	W12X79	W12X72	W18X97	W33X141	W12X72	W10X88
23	W8X48	W12X40	W21X73	W12X65	W14X43	W8X48
24	W24X68	W24X76	W12X87	W30X326	W18X119	W27X84
25	W14X61	W10X77	W18X71	W12X72	W12X152	W14X61
26	W21X50	W16X50	W27X102	W8X28	W12X53	W10X39
27	W8X40	W10X49	W8X48	W30X124	W12X45	W12X40
28	W8X67	W14X61	W24X117	W24X94	W12X72	W18X76
29	W10X39	W18X97	W18X97	W16X89	W12X58	W24X68
30	W21X44	W16X45	W16X40	W21X44	W16X40	W18X40
Weight, lb (kg)	138874.67 (62993.55)	142710.96 (64733.69)	143135.29 (64926.17)	192834.39 (87468.21)	139016.28 (63056.72)	134050.55 (60804.31)

Table 2**:** Comparison of results for 132-member space steel frame

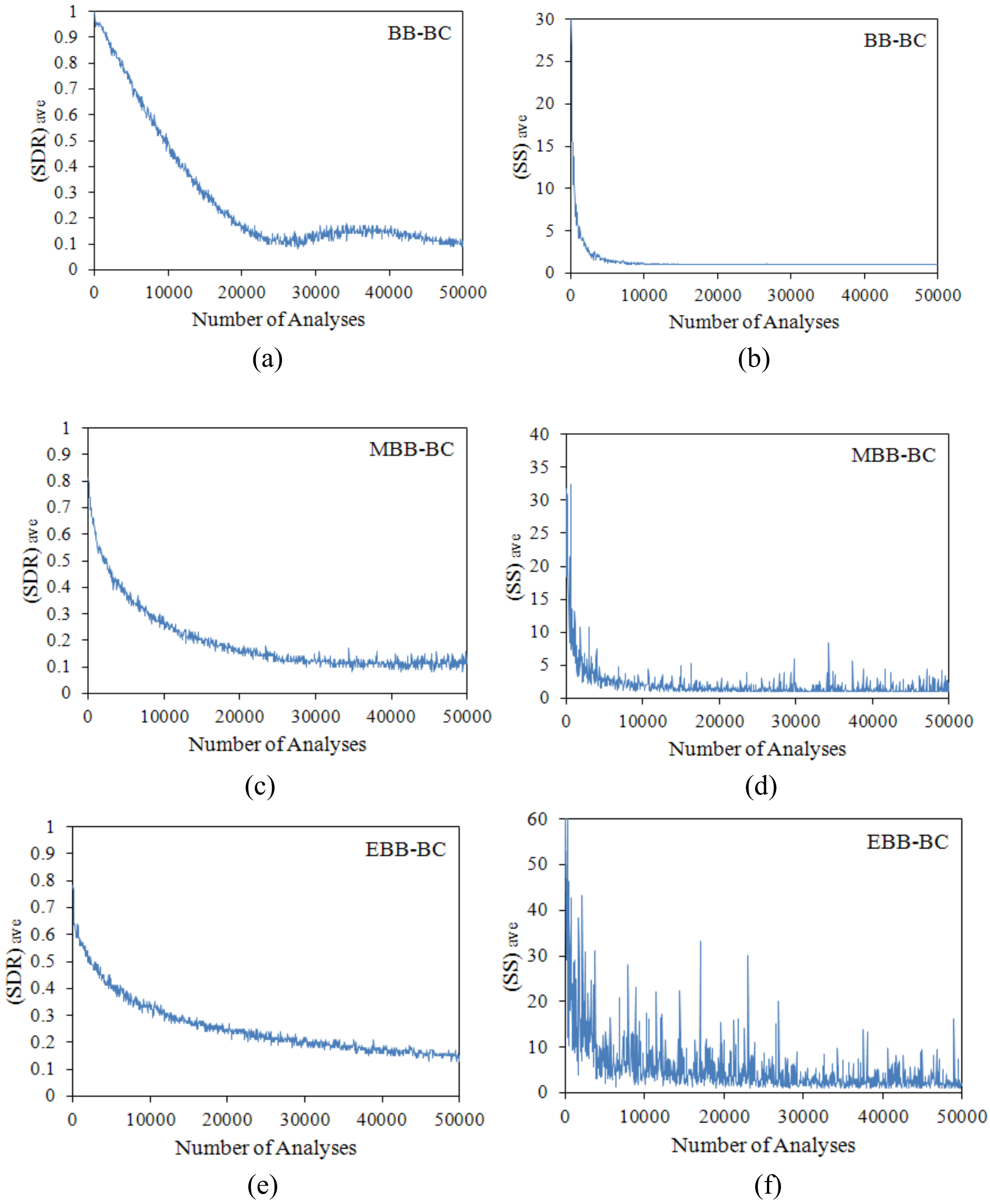

Figure 4: $(SDR)_{ave}$ and $(SS)_{ave}$ variations for 132-member space steel frame example; (a, b) BB-BC, (c, d) MBB-BC, and (e, f) EBB-BC variants

5 Concluding remarks

Performance enhancement of the existing metaheuristic algorithms for tackling specific optimization problems has become one of the most frequent strategies in the recent years. In the present study, an enhanced variant of the BB-BC algorithm, namely EBB-BC, is developed for code based design optimization of steel frame structures. The key issue in creating a new metaheuristic optimization technique as well as modification of an existing algorithm is how to adjust a balance between the exploration and exploitation characteristics of the technique. In this regard, the main concern is to investigate the effect of changes in the structure of the algorithm on the quality of final solutions. Regarding the optimum designs attained based on the BB-BC, EBB-BC and the priviously proposed MBB-BC algorithm, it is concluded that exploitation characteristic of the standard algorithm is more dominant which results in getting stuck in local optima in case of complicated practical design optimization problems of steel frames. However, MBB-BC algorithm shows a better performance in comparison to the BB-BC, by increasing the exploration properties of the method through employing more reasonable $(SDR)_{ave}$ as well as $(SS)_{ave}$ values. Finally, EBB-BC version of the algorithm which utilizes an exponential distribution to generate new candidate solutions, is more successful in adjusting the balance between exploration and exploitation and demonstrates a better performance than those of other two approaches and attains the best design for the investigated example. Further, comparison of the optimization results with thoes of different metaheuristic approaches clearly indicate the robustness and efficiency of the EBB-BC algorithm as a novel version of the well known BB-BC technique in tackling practical design optimization instances of steel frames.

References

[1] F. Erbatur, M.M. Al-Hussainy, “Optimum design of frames”, Computers and Structures, 45(5–6), 887–891, 1992.

[2] E.I. Tabak, P.M. Wright, “Optimality criteria method for building frames”, Journal of Structural Division, ASCE, 107(7), 1327–1342, 1981.

[3] M.P. Saka, “Optimum design of steel frames with stability constraints” Computers and Structures, 41(6), 1365–1377, 1991.

[4] X-S. Yang, “Nature-inspired metaheuristic algorithms” Luniver Press, 2008.

[5] L. Lamberti, C. Pappalettere, "Metaheuristic design optimization of skeletal structures: a review", Computational Technology Reviews, 4, 1-32, 2011. doi:10.4203/ctr.4.1

[6] M.P. Saka, "Optimum Design of Steel Frames using Stochastic Search Techniques Based on Natural Phenomena: A Review", in B.H.V. Topping, (Editor), "Civil Engineering Computations: Tools and Techniques", Saxe-Coburg Publications, Stirlingshire, UK, Chapter 6, pp 105-147, 2007. doi:10.4203/csets.16.6

[7] O.K. Erol, I. Eksin, “New optimization method: big bang-big crunch”, Advances in Engineering Software, 37, 106–11, 2006.

[8] M.H. Afshar, I. Motaei, “Constrained big bang-big crunch algorithm for optimal solution of large scale reservoir operation problem”, International Journal of Optimization in Civil Engineering, 1(2), 357–375, 2011.

[9] H. Tang, J. Zhou, S. Xue, L. Xie, “Big bang-big crunch optimization for parameter estimation in structural systems”, Mechanical Systems and Signal Processing, 24(8), 2888–2897, 2010.

[10] C.V. Camp, “Design of space trusses using big bang-big crunch optimization”, Journal of Structural Engineering, ASCE, 133, 999–1008, 2007.

[11] A. Kaveh, H. Abbasgholiha, “Optimum design of steel sway frames using big bang-big crunch algorithm”, Asian Journal of Civil Engineering, 12(3), 293–317, 2011.

[12] L. Lamberti, C. Pappalettere, "A Fast Big Bang-Big Crunch Optimization Algorithm for Weight Minimization of Truss Structures", in Y. Tsompanakis, B.H.V. Topping, (Editors), "Proceedings of the Second International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering", Civil-Comp Press, Stirlingshire, UK, Paper 11, 2011. doi:10.4203/ccp.97.11

[13] A. Kaveh, S. Talatahari, “Size optimization of space trusses using big bang–big crunch algorithm”, Computers and Structures, 87(17–18), 1129–1140, 2009.

[14] A. Kaveh, S. Talatahari, “Optimal design of Schwedler and ribbed domes via hybrid big bang-big crunch algorithm”, Journal of Constructional Steel Research, 66 (3), 412–419, 2009.

[15] A. Kaveh, S. Talatahari, “A discrete big bang-big crunch algorithm for optimal design of skeletal structures”, Asian Journal of Civil Engineering, 11(1), 103–123, 2010.

[16] A. Kaveh, S. Talatahari, R. Sheikholeslami, "Optimum Seismic Design of Gravity Retaining Walls using the Heuristic Big Bang-Big Crunch Algorithm", in Y. Tsompanakis, B.H.V. Topping, (Editors), "Proceedings of the Second International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering", Civil-Comp Press, Stirlingshire, UK, Paper 4, 2011. doi:10.4203/ccp.97.4

[17] S. Kazemzadeh Azad, O. Hasançebi, O.K. Erol, “Evaluating efficiency of big-bang big-crunch algorithm in benchmark engineering optimization problems”, International Journal of Optimization in Civil Engineering, 1(3), 495–505, 2011.

[18] ASD-AISC. “Manual of steel construction-allowable stress design”, 9th ed., Chicago, Illinois, USA, 1989.

[19] ASCE 7-05, “Minimum design loads for building and other structures”, American Society of Civil Engineering, 2005.

[20] O. Hasançebi, S. Çarbas, E. Dogan, F. Erdal, M.P. Saka, “Performance evaluation of metaheuristic search techniquesin the optimum design of real size pin jointed structures”, Computers and Structures, 87(5–6), 284–302, 2009.

[21] O. Hasançebi, “Adaptive evolution strategies in structural optimization: enhancing their computational performance with applications to large-scale structures”, Computers and Structures, 86 (1–2), 119–132, 2008.

[22] O. Hasançebi, S. Çarbaş, M.P. Saka, Improving the performance of simulated annealing in structural optimization, Structural and Multidisciplinary Optimization, 41, 189–203, 2010.
[23] O. Hasançebi, S. Kazemzadeh Azad, "Discrete Size Optimization of Steel Trusses using a Refined Big Bang–Big Crunch Algorithm", Structural and Multidisciplinary Optimization, (under review).