9th International Congress on Civil Engineering, May 8-10, 2012
Isfahan University of Technology (IUT), Isfahan, Iran

# An Improved Big Bang – Big Crunch Algorithm For Size Optimization of Trusses

**Behrooz Hassani [1], Mostafa Assari [2]*, Morteza Kazemi Torbaghan [3]**

**1- Associate Professor of Civil Engineering, Shahrood University of Technology, Shahrood, Iran**
**2,3- Faculty of Civil Engineering , Islamic Azad University Kashmar Branch, Kashmar, Iran**

assari@iaukashmar.ac.ir

**Abstract**

The Big Bang–Big Crunch (BB–BC) optimization algorithm is a new optimization method that relies on the Big Bang and Big Crunch theory, one of the theories of the evolution of the universe. This method is among the heuristic population-based search procedures that incorporate random variation and selection, such as genetic algorithm (GA) and simulated annealing (SA). Alongside the main advantages of these methods, the problems resulting from the improper distribution of candidate solutions cannot be ignored, especially for high-dimensional functions. In this paper a method, namely Audze-Eglais' approach, has been applied to produce population that increases accuracy via homogeneous candidate solutions. Numerical results demonstrate the efficiency of the improved BB-BC method compared to other heuristic algorithm.

**Keywords: Big Bang-Big Crunch, Size optimization, Truss structures, Heuristic algorithms**

## 1. INTRODUCTION

In general, the optimization techniques used in structural design can be categorized into classical and heuristic search methods. Classical optimization methods such as linear programming, nonlinear programming and optimality criteria often require substantial gradient information. In these methods the final results depend on the initially selected points and the number of computational operations increases with the size of the structure. The solution in these methods does not necessarily correspond to the global optimum. Many engineering design problems are too complex to be handled with mathematical programming methods; while heuristic search methods do not require the data as in the conventional mathematical programming and have better global search abilities than the classical optimization algorithms [1].

A new optimization heuristic method relied on one of the theories of the evolution of the universe namely, the Big Bang and Big Crunch theory is introduced by Erol and Eksin [2] which has a low computational time and high convergence speed. According to this theory, in the Big Bang phase energy dissipation produces disorder and randomness is the main feature of this phase; whereas, in the Big Crunch phase, randomly distributed particles are drawn into an order. The Big Bang–Big Crunch (BB–BC) Optimization method similarly generates random points in the Big Bang phase and shrinks these points to a single representative point via a center of mass in the Big Crunch phase. After a number of sequential Big Bangs and Big Crunches where the distribution of randomness within the search space during the Big Bang becomes smaller and smaller about the average point computed during the Big Crunch, the algorithm converges to a solution.

In this study, an Improved Big Bang–Big Crunch optimization (IBB–BC) is implemented to solve the truss optimization problems. The IBB–BC method consists of two phases: a Big Bang phase where candidate solutions are randomly distributed over the search space, and a Big Crunch phase working as a convergence operator where the center of mass is generated. Then new solutions are created by using the center of mass to be used as the next Big Bang. These successive phases are carried repeatedly until a stopping criterion has been met.

## 2. PROBLEM FORMULATION

The optimization problem is the minimization the weight of the structure subject to stress, displacement and minimum member size constrains. The objective function is:

$$W\left(\{x\}\right)=\sum_{i=1}^{n}\gamma_i\,L_i\,A_i \tag{1}$$

where $L_i$ is the length, $\gamma_i$ is the material density of member, and $A_i$ is the cross sectional of the $i$-th bar. Note that the objective function is linear on the design variables $A_i$. The problem is subject to tensile and compressive stress constraints, bounds on displacements, and side constraints on the areas, as follows:

$$\begin{array}{ll}
\sigma_{\min}\le\sigma_i\le\sigma_{\max} & i=1,\dots,n\\
\sigma_i^{\,b}\le\sigma_i\le 0 & i=1,\dots,ns\\
\delta_{\min}\le\delta_i\le\delta_{\max} & i=1,\dots,m\\
A_{\min}\le A_i\le A_{\max} & i=1,\dots,ng
\end{array}$$

where $n$ = number of members making up the structure; $ns$ = number of compression elements; $m$ = number of nodes; $ng$ = number of groups (number of design variables); $\sigma_i$ and $\delta_i$ = the stress and nodal deflection, respectively; $\sigma_i^b$ allowable buckling stress in member $i$ when it is in compression. $A_{min}$ and $A_{max}$= minimum and maximum cross section area.

BB–BC does not require an explicit relationship between the objective function and constraints. Instead, the objective function for a set of design variables can be penalized to reflect any violation of the design constraints. In utilizing the penalty functions, if the constraints are between the allowable limits, the penalty will be zero; otherwise, the amount of penalty is obtained by dividing the violation of allowable limit to the limit itself [3]. After analyzing a structure, the deflection of each node and the stress in each member are obtained. These values are compared to the allowable limits to calculate the penalty functions as:

$$\begin{cases}\sigma_i^{\min}<\sigma_i<\sigma_i^{\max}\Rightarrow\Phi_\sigma^{(i)}=0\\ \sigma_i^{\min}>\sigma_i\ \ or\ \ \sigma_i^{\max}<\sigma_i\Rightarrow\Phi_\sigma^{(i)}=\dfrac{\sigma_i-\sigma_i^{\min/\max}}{\sigma_i^{\min/\max}}\end{cases}\qquad i=1,2,\dots,n \tag{2}$$

$$\begin{cases}\sigma_b<\sigma_i<0\Rightarrow\Phi_{\sigma b}^{(i)}=0\\ \sigma_i<0\wedge\sigma_i<\sigma_b\Rightarrow\Phi_{\sigma b}^{(i)}=\dfrac{\sigma_i-\sigma_b}{\sigma_b}\end{cases}\qquad i=1,2,\dots,ns \tag{3}$$

$$\begin{cases}\delta_i^{\min}<\delta_i<\delta_i^{\max}\Rightarrow\Phi_\delta^{(i)}=0\\ \delta_i^{\min}>\delta_i\ \ or\ \ \delta_i^{\max}<\delta_i\Rightarrow\Phi_\delta^{(i)}=\dfrac{\delta_i-\delta_i^{\min/\max}}{\delta_i^{\min/\max}}\end{cases}\qquad i=1,2,\dots,m \tag{4}$$

In optimizing structures, the main objective is to find the minimum amount of the merit function. This function is defined as [4]:

$$f^{\,k}=\varepsilon_1.W^{\,k}+\varepsilon_2.\left(\Phi_\sigma^k+\Phi_{\sigma b}^k+\Phi_\delta^k\right)^{\varepsilon_3} \tag{5}$$

$f^{\,k}$ = objective function for the $k$-$th$ candidate; $\varepsilon_1$, $\varepsilon_2$ and $\varepsilon_3$ = coefficients of objective function. $\Phi_\sigma^{\,k}$ , $\Phi_\delta^{\,k}$ and $\Phi_{\sigma b}^{\,k}$ = summation of stress penalties; summation of nodal deflection penalties and summation of buckling stress penalties for candidate k; respectively. In this paper, for a better control on other parameters, $\varepsilon_1$ is set to 1. The coefficient $\varepsilon_2$ is taken as the weight of the structure and the coefficient $\varepsilon_3$ is set in a way that the penalties decrease. The cross-sectional areas can also be reduced. Therefore, in the first iterations of the search process, $\varepsilon_3$ is set to 1.5 but gradually it is increased to 3, Ref. [4].

## 3. BIG BANG – BIG CRUNCH OPTIMIZATION ALGORITHM

The Big Bang-Big Crunch (BB-BC) optimization method developed by Erol and Eksin [2] consists of two main steps: The first step is the Big Bang phase where candidate solutions are randomly distributed over the search space and the next step is the Big Crunch phase where a contraction procedure calculates a center of mass for the population.

The initial Big Bang population is randomly generated over the entire search space similar to any other evolutionary search algorithm. All subsequent Big Bang phases are randomly distributed around the center of mass or the best fit individual in a similar fashion. In [2], the working principle of the Big Bang phase is explained as energy dissipation or the transformation from an ordered state (a convergent solution) to a disordered or chaotic state (new set of candidate solutions).

After the Big Bang phase, a contraction procedure is applied during the Big Crunch. In this phase, the contraction operator takes the current positions of each candidate solution in the population and its associated cost function value and computes a centre of mass according to (6),

$$X_{COM} = \frac{\sum_{i=1}^{N} \frac{1}{f^{i}} x_{i}}{\sum_{i=1}^{N} \frac{1}{f^{i}}} \tag{6}$$

where $X_{COM}$ is the position vector of the center of mass, $x_i$ is the position vector of the *i-th* candidate, $f^i$ is the cost function value of the *i-th* candidate, and $N$ is the population size. The new generation for the next iteration Big Bang phase is normally distributed around $X_{COM}$ . The new candidates around the centre of mass are calculated by adding or subtracting a normal random number whose value decreases as the iterations elapse. This can be formalized as

$$X^{new} = X_{COM} + \frac{r\alpha(x_{\max} - x_{\min})}{k} \tag{7}$$

where $r$ is a normal random number, $\alpha$ is a parameter limiting the size of the search space, $x_{max}$ and $x_{min}$ are the upper and lower limits, and $k$ is the iteration step. Since normally distributed numbers can be exceeding $\pm 1$, it is necessary to limit the population to the prescribed search space boundaries. This narrowing down restricts the candidate solutions into the search space boundaries.

Instead of the centre of mass, other points like the best fit individual can also be chosen as the starting point in the Big Bang phase. In the experiments reported in this paper we apply an elitist strategy introduced by Camp [5]. The positions of new candidate solutions at the beginning of each Big Bang are normally distributed around a new point located between the center of mass and the best solution,

$$X^{new} = \beta . X_{COM} + (1-\beta) . X_{BEST} + \frac{r\alpha(x_{\max} - x_{\min})}{k} \tag{8}$$

where $\beta$ is the parameter controlling the influence of the global best solution $X_{BEST}$ on the location of new candidate solutions. This modification of generating the new solution can be viewed as to be an elitist strategy, where the best solution influences the direction of the search.

These successive explosion and contraction steps are carried out repeatedly until a stopping criterion has been met. A maximum number of iterations is utilized as a stopping criterion.

## 4. IMPROVED BB-BC (IBB-BC) METHOD FOR STRUCTURAL DESIGN

Although BB–BC performs well in the exploitation (the fine search around a local optimum), there are some problems in the exploration (global investigation of the search place) stage. If all of the candidates in the initial Big Bang are collected in a small part of search space, the BB–BC method may not find the optimum solution and with a high probability, it may be trapped in that sub-domain. One can consider a large number for candidates to avoid this defect, but it causes an increase in the function evaluations as well as the computational costs [3].

This paper uses the Audze-Eglais' approach capacities to improve the exploration ability of the BB–BC algorithm. There is wide literature on elaboration of plan (design) of experiment. Good results can be obtained using experiment designs proposed in [6], where for each variable $x_i$ the number of different values is equal to the number of experiments. Usually, it is assumed that numerical values of $x_i$ are uniformly distributed in the interval [+1, -1]. For these designs the points are distributed according to the criterion of minimum of potential energy of repulsive forces, which are defined for a set of points of unit mass. The magnitude of these repulsive forces is inversely proportional to the cubed distance between the points:

$$U = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \frac{1}{L_{ij}^{2}} \Rightarrow \min \tag{9}$$

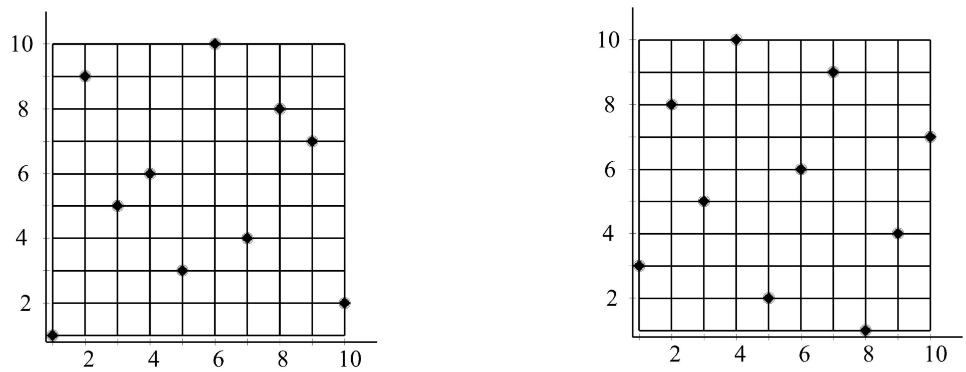
Here $L_{ij}$ is a distance between the reference points having numbers $i$ and $j$ ($i \neq j$) and $N$ is number of points.

The optimization problem consists of finding an Latin hypercube sample which minimizes $U$. For this purpose the columnwise-pairwise (CP) algorithm [7] is employed.

According to the method due to Audze-Eglais, we select the Latin hypercube sample that minimizes the quantity $U$ as given above. This is expected to distribute the experiment points as uniformly distributed as possible within the design variable domain. For example, for a number of factors (design variables) $N = 2$ and $P = 10$, the matrix is

$$\begin{vmatrix} 8 & 10 & 4 & 6 & 2 & 3 & 9 & 5 & 7 & 1 \\ 1 & 7 & 10 & 6 & 8 & 5 & 4 & 2 & 9 & 3 \end{vmatrix} \tag{10}$$

The plan (10) is represented in Figure 1 and compared with a LH for two design variables with 10 runs. The advantages of this method are the space-filling property as shown in Figure 1 and the presentation of the data as tabulated designs.




**(a)** **(b)**

**Figure 1. Comparison between (a) Latin Hypercube Design (b) and Audze-Eglias Design**

In the Big Crunch step, we can use Audze-Eglais' approach for filling up the input space in a uniform fashion. This algorithm not only considers the center of mass as the average point in the beginning of each Big Bang, but also avoids the candidates that to be collected in a small part of search space. Another advantage of Audze-Eglais' approach is to select a small number of candidates because of selecting the best candidates with reasonable distribution. Thus, the pseudo-code of the IBB–BC algorithm can be summarized as follows:

**Step 1 (Big Bang Phase):** An initial generation of N candidates is generated randomly in the search space.
**Step 2 :** The cost function values of all the candidate solutions are computed.
**Step 3 (Big Crunch Phase):** The center of mass is calculated. Either the best fit individual or the center of mass is chosen as the point of Big Bang Phase.
**Step 4 :** New candidates are calculated around the new point calculated in Step 3 by adding or subtracting a random number whose value decreases as the iterations elapse.

**Step 5 :** A set of candidates is chosen which minimizes *U* (Eq.(9)).
**Step 6 :** Return to Step 2 until stopping criteria has been met.

## 5. NUMERICAL EXAMPLES

In order to verify the proposed IBB-BC, we carried out the structural design optimization for two benchmarking design problems of 10-bar and 25-bar truss structures. The final results are compared to the solutions of other methods to demonstrate the efficiency of the present approach. The algorithms are coded in FORTRAN and the structures are analyzed using the direct stiffness method.

5.1. Weight optimization of a 10 bars plane truss
The geometry of 10-bar plane truss structure is show in Figure 2. This optimization problem has been studied by many researchers, and solutions by many different optimization approaches are available in the literature. The objective function of the problem is to minimize the weight of the structure. The input data for this problem are Young's modulus, E = 6.895 × $10^4$ MPa ($10^4$ Ksi), material density, ρ = 2767.990 kg/$m^3$ (0.1 lb/$in^3$) and vertical downward loads of 445.374 kN (100 Kips) at joints 2 and 4. The allowable displacement is limited to 5.08 cm (2 in) in both x and y directions at all nodes, and the allowable stress = ± 172.375 MPa (25 ksi) for all members. Design parameters: $A_i$ $(cm^2)$ ∈ {10.45 , 11.61 , 12.84 , 13.74 , 15.35 , 16.9 , 18.58 , 19.94 , 20.19 , 21.81 , 22.39 , 22.90 , 23.42 , 24.77 , 24.97 , 25.03 , 26.97 , 27.23 , 28.97 , 29.61 , 30.97 , 32.06 , 33.03 , 38.32 , 46.58 , 51.4 , 74.19 , 87.1 , 89.68 , 91.61 , 99.99 ,103.23 , 109.03 , 121.29 , 128.39 , 141.94 , 147.74 , 170.93 , 183.87 , 193.5 , 216.13}

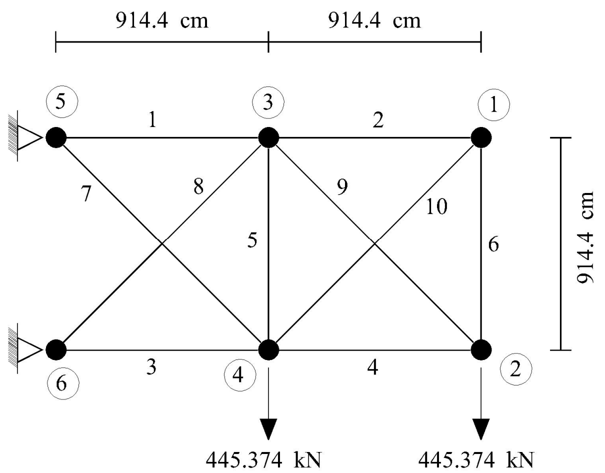



**Figure 2. Configuration of 10-Bar Truss**

**Table 1. Designs for 10-Bar Truss Using Discrete Variables**

| Method | Optimum Weight (kg) | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | A10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2490.72 | 216.13 | 10.45 | 147.74 | 99.99 | 10.45 | 10.45 | 51.4 | 141.94 | 141.94 | 10.45 |
| 2 | 2546.37 | 216.13 | 10.45 | 141.94 | 99.99 | 10.45 | 10.45 | 91.61 | 128.39 | 128.39 | 16.90 |
| 3 | 2626.31 | 193.55 | 16.9 | 147.74 | 147.74 | 16.9 | 16.9 | 51.4 | 147.74 | 141.94 | 10.45 |
| 4 | 2479.38 | 216.13 | 10.45 | 141.9 | 87.10 | 11.61 | 10.45 | 74.19 | 141.94 | 128.39 | 10.45 |

Notes
1- Cai and Thiereu [8].
2- Rajeev and Krishnamoorthy [9].
3- Jang et al. [10].
4- Present Work (IBB-BC)

The optimal parameters of each design variables are given in Table 1. The maximum calculated stress and deflection, in IBB-BC Method, are 94.2 MPa and $5.08\times10^{-2}$ m, respectively. These results satisfy the constraint conditions. The convergence history of the optimum solution for the weight of the truss is illustrated in Figure 3.

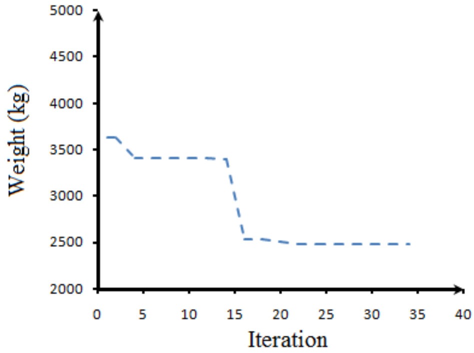



**Figure 3. Convergence history of weight for 10 bar truss**

5.2. Weight optimization of a 25 bars space truss

The second example is the 25 bars space truss, shown in Figure 4. Twenty five members are categorized into eight groups. The member groupings are given in Table 2. Loading conditions for this space truss are given in Table 3. The assumed data are: Young's modulus, E = $6.895 \times 10^4$ MPa ($10^4$ Ksi), material density, ρ = 2767.990 kg/m$^3$ (0.1 lb/in$^3$). The allowable displacement is limited to 8.89 mm (0.35 in) in x, y and z directions at all nodes, and the allowable stress = ± 275.8 MPa (40 ksi) for all members. Design parameters: $G_i$ $(cm^2)$ ∈ {0.645, 1.29, 1.935, 2.58, 3.225, 3.87, 4.515, 5.16, 5.805, 6.45, 7.095, 7.74, 8.385, 9.03, 9.675, 10.32, 10.965, 11.61, 12.255, 12.9, 13.545, 14.19, 14.835, 15.48, 16.125, 16.77, 18.06, 19.35, 20.64, 21.93}

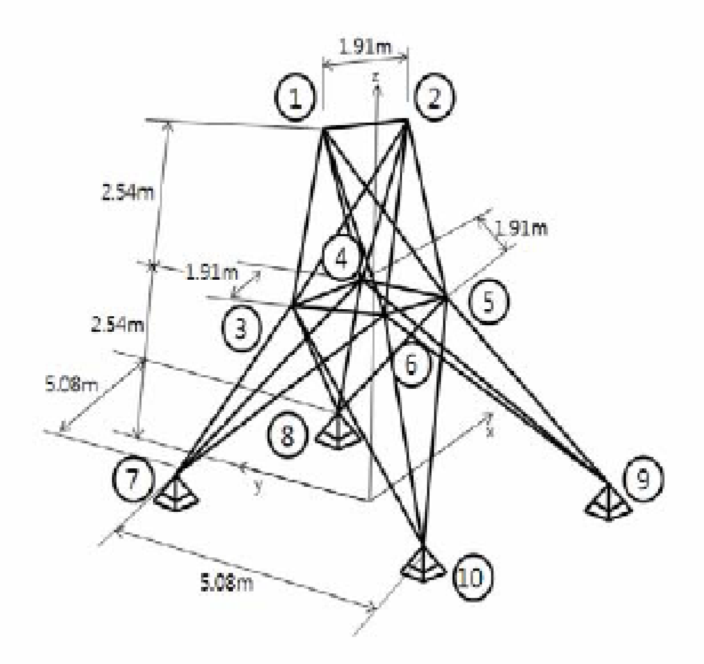



**Figure 4. Configuration of 25-Bar Truss**

**Table 2. Design variable of the 25 bars truss**

| Design Variable Number | End Nodes of members |
|---|---|
| 1 | (1,2) |
| 2 | (1,4),(1,5),(2,3),(2,6) |
| 3 | (1,3),(1,6),(2,4),(2,5) |
| 4 | (3,6),(4,5) |
| 5 | (3,4),(5,6) |
| 6 | (3,10),(4,9),(5,8),(6,7) |
| 7 | (3,8),(4,7),(5,10),(6,9) |
| 8 | (3,7),(4,8),(5,9),(6,10) |

**Table 3. Loading Data**

| Nodal Number | Px (kN) | Py (kN) | Pz (kN) |
|---|---|---|---|
| 1 | 4.454 | -44.53 | -44.53 |
| 2 | 0 | -44.53 | -44.53 |
| 3 | 2.227 | 0 | 0 |
| 4 | 2.672 | 0 | 0 |

**Table4. Designs for 25-Bar Truss Using Discrete Variables**

| Method | Optimum Weight (kg) | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 221.09 | 0.645 | 0.645 | 21.93 | 0.645 | 12.9 | 6.45 | 4.515 | 21.93 |
| 2 | 247.67 | 0.645 | 11.61 | 14.835 | 1.29 | 0.645 | 5.16 | 11.61 | 19.35 |
| 3 | 247.91 | 14.19 | 1.29 | 21.94 | 9.68 | 4.52 | 4.52 | 9.68 | 21.94 |
| 4 | 219.16 | 0.645 | 12.255 | 14.835 | 2.58 | 1.29 | 5.16 | 5.16 | 19.35 |

Notes

1- Cai and Thiereu [8].
2- Rajeev and Krishnamoorthy [9].
3- Jang et al. [10].
4- Present Work (IBB-BC)

The optimal parameters of each design variables are given in Table 4. The maximum calculated stress and deflection, in IBB-BC Method, are 65.3 MPa and $8.9\times10^{-3}$ m, respectively. These results satisfy the constraint conditions. The convergence history of the optimum solution for the weight of the truss is illustrated in Figure 5.

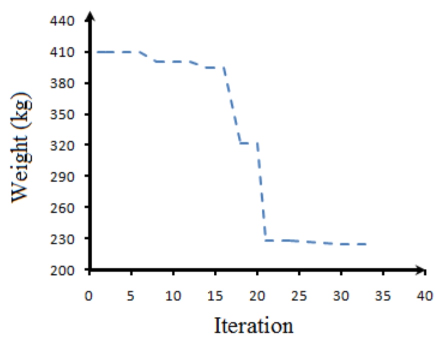



**Figure 5. History of the weight for 25-bar truss**

## 6. CONCLUSIONS

In this paper, it was shown a new heuristic population-based search relied on the Big Bang and Big Crunch theory (BB–BC) of the evolution of the universe. The proposed Improved BB-BC (IBB-BC) method presented an optimization method, combining the BB-BC algorithm with Audze-Eglais' approach.

The convergence speed and accuracy of this method can be accelerated by selecting the best position of all candidates in the beginning of each Big Bang step. On the other hand, other heuristic techniques present convergence difficulty or get trapped at a local optimum in large size structures, the proposed method solved this problem by using Audze-Eglais' approach which opts the best distribution of candidates for filling up the search-space and reduces the number of analyses for convergence.

From the results of this study it can be seen that the IBB–BC method has better performance than other heuristic algorithms.

## 7. REFERENCES

1. Coello CA. (2002), "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**(11-12), pp. 1245–87.

2. Erol OK, Eksin I. (2006), "New optimization method: Big Bang–Big Crunch", Advances in Engineering Software, **37**, pp. 106–11.

3. Kaveh A, Talatahari S. (2009), "Size optimization of space trusses using big bang–big crunch algorithm", Computers and Structures, **87**(17-18), pp. 1129–40.

4. Kaveh A, Farahmand Azar B, Talatahari S. (2008), "Ant colony optimization for design of space trusses", Int J Space Struct, **23**(3), pp. 167–81.

5. C. V. Camp. (2007), "Design of space trusses using big bang-big crunch optimization", Journal of Structural Engineering, **133**(7), pp. 999-1008.

6. Audze P, Eglais V. (1977), "New approach to planning out of experiments", Problems of Dynamics and Strength, pp. 5104-7.

7. Ye KQ, Li W, Sudjianto A. (2000), "Algorithmic construction of optimal symmetric Latin hypercubes", Journal of Statistical Planning and Inference, **90**, pp. 145-59.

8. Cai, J. B., and Thiereut, G. (1993), "Discrete Optimization of Structures Using an Improved Penalty Function Method", Engineering Optimization, **21**(4), pp.293-306.

9. Rajeev, S., and Krishnamoorthy, C. S. (1992), "Discrete Optimization of Structures Using Genetic Algorithms", Journal of Structural Engineering, ASCE, **118(**5), pp. 1233-1250.

10. Jang, S., Jung, W., Kwon, B., and Choi, Y. (2009), "A Study on Structural Design Optimization using TMSA in Discrete Searching Space" , World Academy of Science, Engineering and Technology, **56**, pp. 626-630.