Vol. 5(14) Jan. 2015, PP. 1987-1998

Contents list available at IJMEC

International Journal of Mechatronics, Electrical and Computer Technology (IJMEC)

Journal Homepage: www.aeuso.org

PISSN: 2411-6173 - EISSN: 2305-0543

Size and Shape Optimization of Two-Dimensional Trusses Using Hybrid Big Bang-Big Crunch Algorithm

Hossein Lotfi and Ali Ghoddosian

Department of Mechanical Engineering, Semnan Branch, Islamic Azad University, Semnan, Iran

Faculty of Mechanical Engineering, Semnan University, Semnan, Iran

***Corresponding Author's E-mail:** hlotfi@live.com*

Abstract

The Hybrid Big Bang-Big Crunch (HBB-BC) optimization algorithm is employed for size and shape optimal design of two-dimensional truss structures. HBB-BC compared to other new heuristic algorithm in some benchmark test. Numerical results demonstrate the efficiency and robustness of the HBB-BC optimization algorithm.

Keywords: *Size and Shape Optimization, Truss Structures, Hybrid Big Bang-Big Crunch, Heuristic Algorithms*

1. Introduction

Optimization dates back to the first handicrafts of human beings. One of the things made by human in which the greatest amount of interest for optimization can be found is structures. Optimization of trusses is one of the most active branches of structures optimization. Structural optimization is generally divided into three sub-categories. Size optimization includes calculation of legal cross-section area for every element. A great amount of research on optimization has been emphasized on size optimization issues. In this type of optimization, truss members are optimized by changing cross-section area whiles members length is fixed in the process of optimization. Finding the optimum cross-section area of a truss members or finding the optimum thickness of a sheet is one of the size optimization examples. The most important feature of size optimization is that the considered designing area and constraints over it are definite and never change during the optimization. With further engineering developments, finding the optimized borders of structures and shape optimization has been paid attention. Shape optimization means finding the borders of structures which is found by changing size of members and position of nodes. In this type of optimization the number of nodes is fixed and to change structure shape, nodes move so that the general shape of structure does not change. Therefore the nodes connectivity and elements table does not change. Topology optimization is another type of optimization in which length of the structure elements and its places can be changing. Generally, topology optimization includes nodes distribution and changing of elements connectivity. In the recent years, the most important and promising researches in the field of optimization is meta-heuristic methods based on nature. This algorithm is based on steady heuristic methods which had significant results in solving difficult and complicated problems. Nature has two great strategies. Choosing awards for the stronger individuals and punishment for the weaker and mutation which sets the ground for random members introduction. Generally, selection is the basic idea of optimization and mutation is an idea for steady search [1]. In the last decades, different algorithms based on this model have been introduced and used to solve the structural optimization. Meta-heuristic methods such as genetic algorithm [2,3],

Article History:
Received Date: Dec. 03, 2014
Accepted Date: Jan. 26, 2015
Available Online: Feb. 20, 2015

particle swarm algorithm [4,5], ant colony algorithm [6,7], firefly algorithm [8], harmony search algorithm [9,10], simulated annealing algorithm [11,12] and other meta-heuristic methods based on the nature have been used to optimize trusses and structures. Here by using one of the newest optimization method obtained from the nature called Hybrid Big Bang – Big Crunch algorithm, we would like to optimize two-dimension trusses [13,14]. In the end, to show strength and efficiency of this algorithm, performance of this method in some benchmark tests has been compared with other new methods of structural optimization.

2. Big Bang-Big Crunch optimization

With regard to the preventive coefficients used in calculation of optimized designing of structures, designers and industrial owners always like to have optimized structures despite large computation costs for massive structures. Truss optimization is one of the most widely-used branches of structural optimization. As mentioned before, size and shape optimization of truss structures includes calculation of cross-section area optimum values for members of $A_i$and length optimum values for members of $L_i$which finally leads to minimum weight of structures. This minimum design should satisfy inequality constraints that lead to limitation in design variables and structural responses. Optimized design for a truss can be formulated as follows:

Minimize:$W(\{x\}) = \sum_{i=1}^{n} \gamma_i . A_i . L_i$
Subject to:$\delta_{min} \le \delta_i \le \delta_{max}$, $i = 1,2,\dots,m$

$$\sigma_{min} \le \sigma_i \le \sigma_{max}, \quad i = 1,2,\dots,n$$
$$\sigma_i^b \le \sigma_i \le 0, \quad i = 1,2,\dots,ns$$
$$A_{min} \le A_i \le A_{max}, \quad i = 1,2,\dots,ng$$
$$L_{min} \le L_i \le L_{max}, \quad i = 1,2,\dots,nq$$

where$W(\{x\})$ is weight of structure, $n$is the number of members making up the structure, $m$is the number of nodes, $ns$is the number of compression elements, $ng$is the number of the groups or the number of shape designing variables,γ_i is the density of the members$i$, L_i is the length of member$i$which is between L_{min} and L_{max}, A_i is cross-section of member A which is between A_{min} and A_{max}, that is lower band and upper band of cross-section. σ_i and δ_i are respectively stress and nodal deflection and σ_i^b is the value of allowable buckling stress in the member i under compression.

Recently, different meta-heuristic algorithms have been used for structural optimization. These methods include heuristic procedures which incorporate random variation and selection. Random selection and the obtained information for each repetition is used to find new points. These algorithms don't require given function and explicit relationship between objective function and constraints. A new optimization method which is relied on one of the evolution theories of the universe is Big Bang- Big Crunch introduced by Erol and Eksin in 2004. This method has low computation time and high convergence speed. According to this theory, in the phase of Big Bang, dissipation causes disorder and randomly distribution is one of the features of this phase. After this phase, in the phase of Big Crunch, random dissipated particles get ordered. Optimization method of BB-BC provides random points similarly in the phase of Big Bang and these points will get together in one point called representative point which is center of mass in Big Crunch phase. After some consecutive Big Bangs and Big Crunches, search space in each explosion gets smaller and smaller and average point is calculated in each Big Crunch so that algorithm converges toward a solution [15]. As said earlier, BB-BC optimization method contains two phases. The first phase, Big Bang in which the candidates of problem solution similar to other evolutionary algorithms is spread all over the search space. Erol and Eksin associated the random nature of Big Bang and energy dissipation or transforming from order and convergent mood to disordered mood through initiating a new set of

candidates. Next phase is Big Crunch which happens after each Big Bang. This phase is the converging operator which has many inputs but only one output which is called center of mass [16,17]. Here, mass refers to the inverse of merit function. It is a point which shows the center of mass and demonstrates by$X_{\mathrm{i}}^{\mathrm{c(k)}}$and is formulated as:

$$X_i^{c(k)} = \frac{\sum_{j=1}^{M} \frac{1}{Mer^j} \cdot X_i^{(k,j)}}{\sum_{j=1}^{M} \frac{1}{Mer^j}}, \qquad i = 1,2,\dots,N$$

Where$\mathrm{X}_{\mathrm{i}}^{\mathrm{(k,j)}}$is $i$component of jth solution which has been produced in $k$thiteration, M is the size of population in Big Bang and N is the number of design variables. After Big Crunch, algorithm provides a new solution which is used in the next Big Bang. Normal distribution operatesaround the mass center in every direction.If standard deviation from this normal distribution function decreases as the number of repetition of algorithms increases.

$$X_i^{(k+1,j)} = X_i^{c(k)} + \frac{r_j \alpha_1 (X_{max} - X_{min})}{k+1}, i = 1,2,\dots,N$$

In which$r_j$is a random number from standard dissipation which changes for each candidate andα_1 is a parameter for limiting the search space.These consequent explosions and step to step crunches happen sequentially so that they can reach the stopping criterion. We can consider the maximum number of iteration as the index of stop. BB-BC method requires no explicit relationship between objective function and constraints. Instead of that, objective function will consider penalty for a set of designing factors which have been deviated from constraints which is the reflection of violation of designing constraints. To find deviation function, if constraints are in the allowable domain, penalty will be zero otherwise, the value of penalty will be obtained by dividing deviation of allowable domain on its limitation. After analyzing a structure, changing and replacement each node and tension inside each member will be obtained. These values will be compared with allowable limit domain to calculate penalty functions as:

$$\begin{cases} \sigma_i^{min} < \sigma_i < \sigma_i^{max} \rightarrow \varphi_\sigma^{(i)} = 0 & , \quad i = 1,2,\dots,n \\ \sigma_i^{min} > \sigma_i \ or \ \sigma_i^{max} < \sigma_i \ \rightarrow \ \varphi_\sigma^{(i)} = \dfrac{\sigma_i - \sigma_i^{min/max}}{\sigma_i^{min/max}} \end{cases}$$

$$\begin{cases} \sigma_b < \sigma_i < 0 \rightarrow \varphi_{\sigma b}^{(i)} = 0 \ , \ i = 1,2,\dots,ns \\ \sigma_i < 0 \ \wedge \ \sigma_i < \sigma_b \ \rightarrow \ \varphi_{\sigma b}^{(i)} = \dfrac{\sigma_i - \sigma_b}{\sigma_b} \end{cases}$$

$$\begin{cases} \delta_i^{min} < \delta_i < \delta_i^{max} \rightarrow \varphi_\delta^{(i)} = 0 & , \quad i = 1,2,\dots,m \\ \delta_i^{min} > \delta_i \ or \ \delta_i^{max} < \delta_i \ \rightarrow \ \varphi_\delta^{(i)} = \dfrac{\delta_i - \delta_i^{min/max}}{\delta_i^{min/max}} \end{cases}$$

In the optimized structures, the main goal is calculating the minimum amount for merit function. This function is defined as follows:

$$Mer^k = \varepsilon_1 . W^k + \varepsilon_2 . (\varphi_\sigma^k + \varphi_\delta^k + \varphi_{\sigma b}^k)^{\varepsilon_3}$$

The function Mer^k is the merit function for the $k$thcandidate,ε_1, $\varepsilon_2$and ε_3 are the coefficients of this function and φ_σ^k, φ_δ^k, and $\varphi_{\sigma\delta}^k$ are the sum of stresspenalties and nodaldeflectionpenalties and buckling stress penalties for the candidate k [18]. For multi ple direction loading after analyzing the

structure and calculating penalties function for each member of loading, the total penalty function is calculated as follows:

$$Mer^k = \varepsilon_1 . W^k + \varepsilon_2 . \sum_{i=1}^{np} (\varphi_{\sigma(i)}^k + \varphi_{\delta(i)}^k + \varphi_{\sigma b(i)}^k)^{\varepsilon_3}$$

Where,np shows the number of multiple direction loadings. Here, to control better on other parameters, value of $\varepsilon_1$is set to 1. Coefficient $\varepsilon_2$demonstrates the weight of structure and coefficient ε_3 is located in a direction that penalty decreases. In this situation, cross-section area also decreases. Therefore, in the first repetition of search process ε_3 is 1.5 But gradually it increases to 3 [19]. Determining design variables and allowable limit of domains is the first step to initiate BB-BC algorithm parameters, then providing candidates randomly including allowable search space, the next step calculating merit function values for all candidates and obtaining mass center by last step results, then calculating new candidates around mass center, repeating this process until getting to stop criterion.

3. Hybrid Big Bang-Big Crunch Optimization

The advantages of BB-BC algorithm for optimal design of structures are like other evolutionary methods. In this algorithm, there is a random search technique and in each cycle it considers a number of existing points in the search space so that the obtained data is used in each cycle for mass center to find new points. BB-BC method has the ability to manage a mixture of continuous and discrete design variables on the problems having multiple loading. Although BB-BC algorithm has good performance in exploitation, in other words fine search is seen around local optimal point, some problems is observed in the exploration stage. It means, global search has some problems in the search space. For example, if all candidates are gathered in a small part of search space at the beginning of Big Bang, optimized reply is not likely to be found by algorithm, and algorithm is more likely to be trapped in a local optimized point. One of the solutions is that a great number of random variables are used to prevent algorithm from in the local optimized points but it causes calculation time increasing and function evaluation and finally computation cost increasing. Here, it has been attempted by using the ability of particles warm optimization algorithm and mixing it with BB-BC optimization method, the ability of BB-BC algorithm can be improved [18].

Particle swarm optimization algorithm (PSO) has been derived from bird flock and fish group movement that, its movement depends on particle personal experience and population experience. In each repetition, a particle way is a mixture of the best place where that particle has previously moved toward (local best) and the best place where its neighbors have moved toward (global best). [20]. Similarly Hybrid Big Bang-Big Crunch (HBB-BC) algorithm does not use only the center of mass and by using the information of best place of each candidate ($X_i^{lbest(k,j)}$) and the best global place ($X_i^{gbest(k)}$), acts to find the solution as follows:

$$X_i^{(k+1,j)} = \alpha_2 X_i^{c(k)} + (1-\alpha_2)(\alpha_3 X_i^{gbest(k)} + (1-\alpha_3) X_i^{lbest(k,j)}) + \frac{r_j \alpha_1 (X_{max} - X_{min})}{k+1}$$
$$\begin{cases} i = 1,2,\dots,N \\ j = 1,2,\dots,M \end{cases}$$

In which ($X_i^{lbest(k,j)}$) the best point of the particle j until the iteration$k$ and ($X_i^{gbest(k)}$) is the best point among all the random variables until iteration$k$. $\alpha_2$andα_3 are regulating parameters that controlling the influence of global best and local best on the new place of the candidates. Another improvement which is done in the BB-BC algorithm is using Sub-Optimization Mechanism (SOM) as accelerating instrument by updating of designing search space. SOM method is based on finite elements regulations that, like finite elements methods, divide designing area into sub-domains and assign the process of optimization in these sub-domains. Hence, unfavorable parts are omitted and

remaining search space is divided into smaller parts for more searches in the next step. This process continues until the remaining area gets smaller than the required size to satisfy accuracy [19, 21]. This mechanism can be done in the following steps for the certain time nc and $k$iteration. Calculating the bounds of designing variables for each group. If $X_i^{gbest(k_{som}-1)}$is the best global solution which is obtained from the step $(k_{som}-1)$ for the design variable$i$, we have:

$$\begin{cases} X_{min,i}^{(k_{som})} = X_i^{gbest(k_{som}-1)} - \beta_1.(X_{max,i}^{(k_{som}-1)} - X_{min,i}^{(k_{som}-1)}) \geq X_{min,i}^{(k_{som}-1)} \\ X_{max,i}^{(k_{som})} = X_i^{gbest(k_{som}-1)} + \beta_1.(X_{max,i}^{(k_{som}-1)} - X_{min,i}^{(k_{som}-1)}) \leq X_{max,i}^{(k_{som}-1)} \\ \begin{cases} i = 1,2,\dots,N \\ k_{som} = 2,\dots,nc \end{cases} \end{cases}$$

The coefficient$\beta_1$is able to regulate and calculates the remaining designing search space. In this paper, we consider its value as 0.3.The value of$X_{min,i}^{(k_{som})}$ and $\mathrm{X}_{\mathrm{max,i}}^{(\mathrm{k_{som}})}$are minimum and maximum of allowable cross-section in the step ($ksom$, respectively. In the first step, the value of$X_{min,i}^{(1)}$ and$X_{max,i}^{(1)}$are set as follows:

$$X_{min,i}^{i} = X_{min}\ , X_{max,i}^{(1)} = X_{max} \quad , \quad i = 1,2,\dots,N$$

Determining the amount of acceptable increasing of designing variables. In each step, allowable values for each group are obtained for coefficientβ_2. In the HBB-BC optimization algorithm, the value of β_2 is 100.

$$X_i^{*(k_{som})} = \frac{(X_{max,i}^{(k_{som})} - X_{min,i}^{(k_{som})})}{\beta_2 - 1} \quad , \quad i = 1,2,\dots,N$$

Initiating the series of allowable cross-section areas. Set of allowable cross-section areas for group i can be defined as follows:

$$A_{min,i}^{(k_{som})}, A_{min,i}^{(k_{som})} + A_i^{*(k_{som})}, \dots, A_{min,i}^{(k_{som})} + (\beta_2 - 1).A_i^{*(k_{som})} = A_{max,i}^{(k_{som})} \quad , \quad i = 1,2,\dots,N$$

Calculating optimum solution for the stepk_{som}, so that the last step of optimization process is done by stop creation for SOM in the BB-BC algorithm. Stopping creation for SOM described as:

$$X_i^{*(nc)} \leq X^* \quad , \quad i = 1,2,\dots,ng$$

$X_i^{*(nc)}$is equal to acceptable increasing of the last step and $X^*$is the acceptable increasing value of primary problem.

The Sub-Optimization Mechanism (SOM) continues optimization until solution gets to the required process creation. SOM method functions as updating of designing area to improve search process and updates designing search space from one step to another step. Also the proper acts of Sub-Optimization Mechanism help spread the particles in the first Big Bang very well. With increasing repetition, SOM coefficient gets bigger and has bigger effect on the process. In effect, regarding the improper effect of inopportune acts of this process in simultaneous size and shape optimization, time of SOM acts is limited. In a way, random variables are allowed to spread in initial repetitions, and in other repetitions SOM will be assigned gradually. Another advantage of SOM is selection of fewer numbers of random components because of decreasing designing search space, in other repetitions, this advantage has been used and the pace of this process will be increased. Of course, it should be paid attention that the time of using should be completely compatible with the number of unknown design variables. The used coefficients have been obtained by repeated algorithms and compatible

with the number of unknowns for simultaneous size and shape optimization, and they have been applied in the suggested examples.

4. Design Examples

In this paper, three truss structures are optimized with present method. Then the final result compared to other advanced heuristic methods to demonstrate the efficiency of this work.

4.1 Size and shape Optimization of 10-bar truss

One of the benchmark tests for performance presentation of trusses optimization is 10-bar truss which has been represented by Piers in 2004. In the same year Romero suggested all necessary information of the problem for solution by linear elastic rules with using single concentrated load and compatible constraints. Truss has ten members and six nodes with the concentrated load $100kips$is used in nodes 2 and 4. The maximum tension is $\pm 25ksi$. Density is $0.1lb/in^3$and modulus elasticity is $1000ksi$. In the figure 1 the characteristics of 10-bar truss has been showed [22].

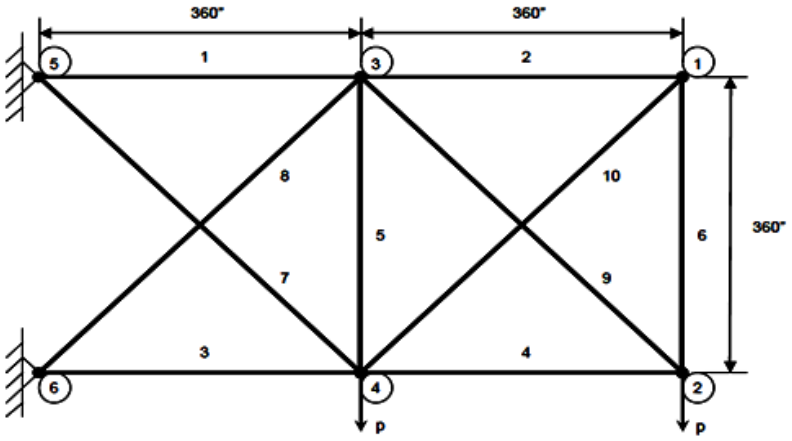

Figure1: 10-bar Truss

Table 1: 10-bar truss specification

Nodes	X (min)	X (max)	Y (min)	Y (max)
1	600	750	50	400
2	720	720	0	0
3	300	600	100	500
4	360	360	0	0
5	0	0	300	750
6	0	0	0	0

The lower nodes of truss are fixed but the upper nodes can be moved. Ten variables of cross-section area designing is minimum $0.1in^2$and maximum $10in^2$ for each member of truss element. In addition, the algorithm includes five variables of designing. Movement limit of each node for shape optimization has been showed in table 1. One of improvements shown in the algorithm is the time of using Sub-Optimization Mechanism. Regarding the high number of designing variable, the process of domain decreasing in the first repetition causes algorithm to be far away from correct results and trapping algorithm in local optimum point. Therefore, to find global optimum more easily, in addition to increasing the number of repetitions, the time of using process of SOM has been put off and after 200 repetitions this mechanism is used. It is remarkable to say that if the time of beginning process of Sub-Optimization mechanism is at the beginning of algorithm, then increasing the number of repetitions is no longer effective and only the time of operation increases. Therefore, using fitness between the beginning time of Sub-Optimization mechanism operation, the number of repetitions and the number of random variables in each step is important based on the number of designing

variables. One of the measures which can help improve algorithm and bring about better results is decreasing the coefficient of decreasing domain through which the results can be improved and decrease the probability of algorithm trapping in local optimization. By improving this coefficient from 0.3 obtained from size optimization to 0.4, the algorithm shows better results. Of course this results improvement has been obtained with higher calculation time. As this coefficient increases more than 0.4 no more improvement is seen in the results, although calculations time increases sharply.

Thus this number is considered as the basis of calculation in other calculations of size and shape optimization by hybrid BB-BC algorithm. The effective components in HBB-BC algorithm process has been considered and shown in the table 2. As seen in lower repetitions, through increasing pace of decreasing domain, finding answer can be done faster with spending enough time, the reason of increasing time with decreasing domain is the random essence of input variables. Another problem offered here is the probability of trapping algorithm in local optimization with decreasing domain coefficient which happens in designing variables with fewer numbers. Therefore using bigger coefficient of domain decreasing and gradual decreasing of designing variables domain, probability of reaching better results is reinforced.

Table 2: Calculation of HBB-BC optimization algorithm coefficient

300 Design variable & 100 Iteration coefficient	**0.2**	**0.3**	**0.4**	**0.5**
Weight (lb)	1249.79	1260.67	1285.70	1301.68
Time (s)	3169	758	719	396
300 Design variable & 300 Iteration coefficient	**0.2**	**0.3**	**0.4**	**0.5**
Weight (lb)	1243.52	1244.47	1245.52	1247.23
Time (s)	19236	11036	10239	8576
500 Design variable & 300 Iteration coefficient	**0.2**	**0.3**	**0.4**	**0.5**
Weight (lb)	1236.22	1235.02	1234.67	1237.17
Time (s)	23123	17235	15236	11032

Optimized 10-bar truss with hybrid BB-BC method has been showed in figure 2. The obtained results for size and shape optimization of 10-bar truss with hybrid BB-BC method with the presented results by Auer and Genesis have been compared in table 3. It should be paid attention Auer and Genesis have used the modified genetic algorithm to optimize 10-bar truss. The obtained weight with 500 random variables and 300 repetitions by using size and shape optimized algorithm of hybrid BB-BC is 1232.27lbthat compared with the weight of 1236.46 lb by Auer and 1232.8 lb by Genesis shows better results. The amount of tension and strain has also been calculated which is in the given range and the conditions of the problem have been considered very well.

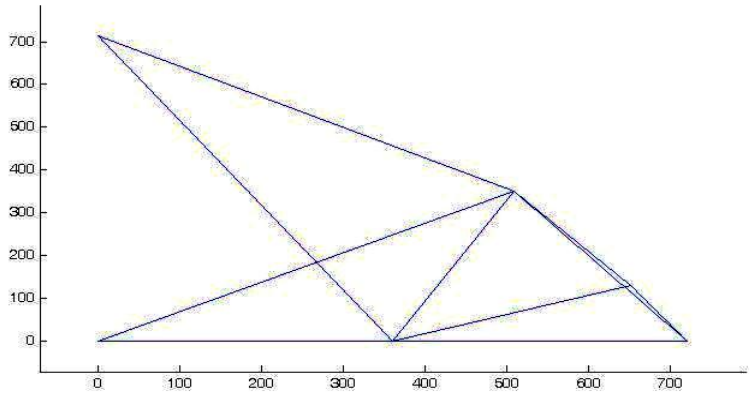

Figure 2: Optimized 10-bar truss by HBB-BC algorithm

Table 3: 10-bar truss optimization result

Design variable	Auer	Genesis	This Work
Sizing variables (in^2)			
A1	4.8847	4.8322	**4.9290**
A2	0.1	0.27127	**2.1703**
A3	4.1691	4.0502	**4.4634**
A4	2.1157	1.9632	**2.2628**
A5	0.1	0.10003	**0.1184**
A6	0.1	0.26833	**2.1566**
A7	4.4825	4.5035	**4.5282**
A8	2.5193	2.3597	**1.9299**
A9	4.4264	4.2159	**2.4610**
A10	0.1	0.10001	**0.2064**
Layout Variables (in)			
X1	642.427	686.284	**651.4798**
X3	523.361	540.459	**508.7414**
Y1	143.557	130.621	**130.7610**
Y3	371.946	355.233	**351.2260**
Y5	694.251	719.865	**713.9121**
Weight (lb)	1236.46	1232.8	**1232.2780**

4.2 Size and shape optimization of 15-bar truss

15-bar truss is one of the design examples used in the benchmark test very much. Truss has 15 members and 8 nodes. In the figure 3 the characteristics of 15-bar truss has been showed. The single $10kips$ vertical load is used in node 8. The nodes 1 and 5 have been considered fixed. The maximum tension is $\pm 25ksi$. The material density is $0.1lb/in^3$ and modulus elasticity is $10000ksi$. The truss has symmetry relative to X axis and in the direction Y there is no symmetry. In the table 4, the movement limit of 15-bar truss has been presented [24].

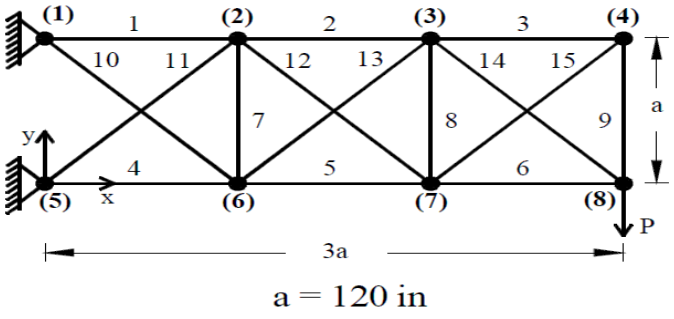

Figure 3: 15-bar truss

Table 4: 15-bar truss specification

Nodes	X (min)	X (max)	Y (min)	Y (max)
1	0	0	120	120
2	100	140	100	140
3	220	260	100	140
4	350	370	50	90
5	0	0	0	0
6	100	140	20-	20
7	220	260	20-	20
8	350	370	20	60

At first, all cross-sections of truss are considered from the same size and horizontal movements of nodes 2, 6, 3 and 7, 4, 8 are the same. So that, they can be independent in vertical direction. In this situation, algorithm has 1 variable for size designing and 8 variables for shape designing. As expected, a proper result has been obtained for shape optimization in short time, but because the elements having less tension have no probability of independent movement against more tension elements, the weight of the obtained truss is slightly different from that of the obtained result in benchmark tests. Therefore, in the main work based on benchmark tests elements are considered to have different cross-sections. In this situation, the truss has 15 variables for size designing and 8 variables for shape designing and totally 23 variables of designing for the algorithm. It should be paid attention that regarding tension considerations inside the truss members with more cross-section variables a better result for the obtained total weight of structure. In this paper, the allowed range of all cross-sections is between $0.1 in^2$and $۲ in^2$. In the figure 4, the optimized 15-bar truss by hybrid BB-BC has been showed. As seen from the table 5, the obtained result by hybrid BB-BC shows a proper improvement compared to other works. The obtained result certainly is better than the result by Chow and Wu and also by Hwang and He. In addition, the obtained result is much better than the results by Tang, Rahami and Kazemzadeh. Here, by using some changes on coefficients and time of using Sub-Optimization Mechanism, some suitable results have been obtained for simultaneous size and shape optimization of two-dimension trusses which shows better results compared to other works. It is remarkable to say that the above results have satisfied allowable strain and tension limit in elements of truss very well.

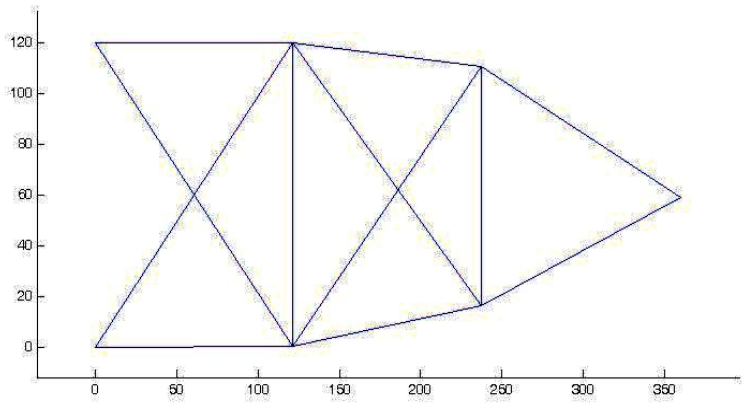

Figure4: Optimized 15-bar truss by HBB-BC algorithm

Table 5: 15-bar truss optimization result

Design variable	Wu & Chow	Tang & al.	Hwang & He	Rahami el al.	Azad	This Work
			Sizing variables (in^2)			
A1	1.174	1.081	0.954	1.081	0.954	**0.974**
A2	0.954	0.539	1.081	0.539	0.539	**0.671**
A3	0.44	0.287	0.44	0.287	0.111	**0.259**
A4	1.333	0.954	1.174	0.954	0.954	**1.020**
A5	0.954	0.954	1.488	0.539	0.539	**0.657**
A6	0.174	0.22	0.27	0.141	0.347	**0.300**
A7	0.44	0.111	0.27	0.111	0.111	**0.100**
A8	0.44	0.111	0.347	0.111	0.111	**0.101**
A9	1.081	0.287	0.22	0.539	0.111	**0.943**
A10	1.333	0.22	0.44	0.44	0.44	**0.317**
A11	0.174	0.44	0.347	0.539	0.44	**0.266**
A12	0.174	0.44	0.22	0.27	0.174	**0.175**
A13	0.347	0.111	0.27	0.22	0.174	**0.201**
A14	0.347	0.22	0.44	0.141	0.347	**0.307**
A15	0.44	0.347	0.22	0.287	0.111	**0.252**

			Layout Variables (in)			
X2	123.18	133.61	118.46	101.57	105.78	**121.4**
X3	231.18	234.75	225.20	227.91	258.59	**237.2**
Y2	107.18	100.44	119.04	134.79	133.62	**120.4**
Y3	119.17	104.73	105.08	128.22	105.00	**110.9**
Y4	60.46	73.76	63.37	54.86	54.45	**59.06**
Y6	-16.72	-10.06	-20	-16.44	-19.92	**0.093**
Y7	15.56	-1.33	-20	-13.30	3.62	**16.48**
Y8	36.64	50.40	57.72	54.857	54.447	**59.05**
Weight (lb)	120.528	79.82	104.573	76.6854	72.5152	**72.4938**

4.3 Size and shape optimization of 18-bar truss

18-bar truss and its characteristics have been showed in the figure 5. Five 20$kips$ loads are used in the nodes 1,2,4,6, and 8. The material density is 0.1lb/in^3 and modulus elasticity is 10000ksi. The maximum tension is $\pm$25ksi. In the table 6, the movement limit of 18-bar truss has been presented [25]. The cross-section areas of truss members were linked into four groups, as follows:

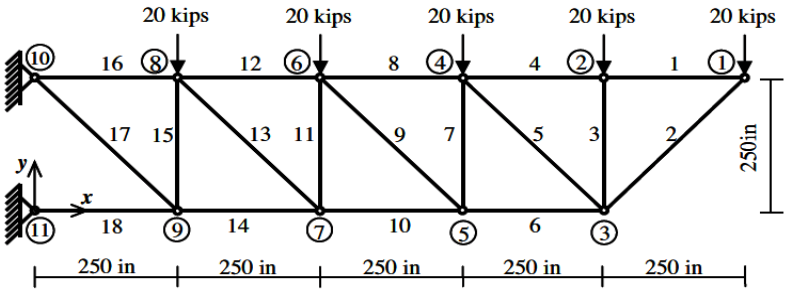

Figure 5: 18-bar truss

Table 6: 18-bar truss specification

Nodes	X (min)	X (max)	Y (min)	Y (max)
1	1250	1250	250	250
2	1000	1000	250	250
3	775	1225	-225	245
4	750	750	250	250
5	525	975	-225	245
6	500	500	250	250
7	275	725	-225	245
8	250	250	250	250
9	25	475	-225	245
10	250	250	0	0
11	0	0	0	0

First group: $A_1 = A_4 = A_8 = A_{12} = A_{16}$,
Second group:$A_2 = A_6 = A_{10} = A_{14} = A_{18}$,
Third group:$A_3 = A_7 = A_{11} = A_{15}$,
Fourth group:$A_5 = A_9 = A_{13} = A_{17}$

The problem has four variables of size designing, and eight variables of shape designing. Generally12 variables of designing can be considered for solving problem by algorithm. The optimized 18-bar truss by big hybrid BB-BC algorithm has been showed in the figure 6.

The results of weight and cross-section of truss members have been showed in the table 7. The obtained result is unexpectedly better than other works in all benchmark tests [26].

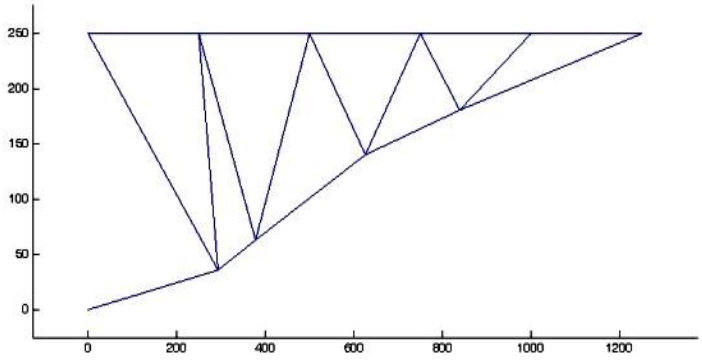

Figure 6: Optimized 18-bar truss by HBB-BC algorithm

Table 7: 18-bar truss optimization result

Design variable	Hasancebi & Erbatur	Rahami et al.	Gholizadeh	This Work
Sizing variables (in^2)				
A1	12.25	12.75	12.50	**12.214**
A2	17.50	18.50	17.50	**17.455**
A3	5.75	4.75	5.75	**5.128**
A5	4.25	3.25	3.75	**4.261**
Layout Variables (in)				
X3	910.0	917.4475	907.2491	**879.1027**
X5	638.0	654.3243	636.7873	**605.0881**
X7	408.0	424.4821	407.9442	**421.1024**
X9	198.0	208.4691	198.7897	**328.8813**
Y3	179.0	193.7899	179.8671	**193.8769**
Y5	141.0	159.9436	141.8271	**138.7172**
Y7	91.0	108.5779	94.0559	**82.5168**
Y9	24.0	37.6349	29.5157	**55.8138**
Weight (lb)	4533.24	4530.68	4512.365	**4509.548**

Conclusions

Hybrid BB-BC algorithm with some changes has been used to solve three size and shape optimization design examples of two-dimension trusses. The changes have been used in the optimized Hybrid BB-BC algorithm by using the feature of Particle Swarm Optimization algorithm point finding and implementing time range in the upgrading domain through using the feature of finite element sub-domain making. In all benchmark problems, the goal is finding the optimum weight of the truss, the characteristics of nodes and the size of elements cross-section in a way that the conditions of tension, strain and nodal deflection have been met and the designing variables should be remained in the given limit. It should be paid attention that the conditions of the compared problems are considered the same. Therefore, the solved problems have been synchronized regarding resources and results have been evaluated with the same conditions for input information and domains. The obtained results certainly show that HBB-BC algorithm were better than the compared algorithms in solving two-dimension trusses. In the problem of 10-bar truss which was with evaluation of coefficients and finding optimum numbers for decreasing domain coefficient and the time of using process of Sub-Optimization Mechanism, the problem has been solved in different situations so that the number of random variables required and the number of the optimized repetitions should be extracted. Finally, the results of solving problems with the obtained coefficients were better than other works. In this problem, the result was better than both researches done on the benchmark tests. In the problem of 15-bar truss first truss has been solved with the same elements which showed a good result in a shorter time then to find a comparable

optimum answer with the benchmark test, the elements have been considered separately. In this situation, the obtained results were better than other works. In another problem, 18-bar truss has been examined. The obtained result for this truss which has been optimized by HBB-BC algorithm is evidently better than other compared works on benchmark test. Generally the results show that Hybrid BB-BC algorithm which has first been used here for simultaneous size and shape optimization of two-dimension trusses demonstrated good results in a short time in all mentioned problems and have been comparable with other advanced heuristic algorithms and has potentially the ability to be used in other fields of structural optimization.

References

[1] D. Befogged, Evolutionary Computation: Toward a New Philosophy of Machine Intelligence, IEEE Press, Third Edition, Piscataway, NJ, 2006.
[2] J. H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor,1975.
[3] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Kluwer Academic Publishers, Boston, MA, 1989.
[4] J. Kennedy, R. Eberhart, Particle Swarm Optimization, Proceedings of IEEE International Conference on Neural Networks IV in Australia, (1995) 1942–1948.
[5] Y. Shi, R. C. Eberhart, A modified particle swarm optimizer, Proceedings of IEEE International Conference on Evolutionary Computation, (1998) 69–73.
[6] M. Dorigo, Optimization: Learning and Natural Algorithms, PhD thesis, Politecnico di Milano Italy, 1992.
[7] M. Dorigo,L. M. Gambardella, Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem, IEEE Transactions on Evolutionary Computation, Volume 1 (1997) 53-66.
[8] L. F. F. Miguel, R. H. Lopez, L. F. F. Miguel, Multimodal size, shape, and topology optimization of truss structures using the Firefly algorithm, Advances in Engineering Software, Volume 56 (2013) 23–37.
[9] K. S. Lee, Z. W. Geem, A new structural optimization method based on the harmony search algorithm, Computers & Structures, Volume 82 (2004) 781–798.
[10] S.O. Degertekin, Improved harmony search algorithms for sizing optimization of truss structures, Computers & Structures, Volumes 92–93(2012) 229–241.
[11] L. Lamberti, An efficient simulated annealing algorithm for design optimization of truss structures, Computers & Structures, Volume 86(2008) 1936–1953.
[12] O. Hasancebi, F. Erbatur, Layout optimization of trusses using simulated annealing, Advances in Engineering Software, Volume 33 (2002) 681–696.
[13] Bilal Alatas,Uniform Big Bang–Chaotic Big Crunch optimization, Communications in Nonlinear Science and Numerical Simulation, Volume 16 (2011) 3696–3703.
[14] O. Hasancebi, S. Kazemzadeh Azad, An exponential big bang-big crunch algorithm for discrete design optimization of steel frames, Computers & Structures, Volumes 110–111 (2012) 167–179.
[15] O. K. Erol, I. Eksin, New optimization method: Big Bang–Big Crunch, Advances in Engineering Software, Volume 37(2006) 106–111.
[16] B. Stuckman,A Global search method for optimizing nonlinear systems, IEEE Transaction on Systems, Volume 18 (1988) 965–77.
[17] O. Tigli, I. Eksin, S. Ertem, O. Boz, A global optimization in controlling a plant, Electrical Power and Energy Systems, Volume30(1994) 257–62.
[18] A. Kaveh, S. Talatahari ,Size optimization of space trusses using Big Bang – Big Crunch algorithm, Computers and Structures, Volume 87 (2009) 1129-1140.
[19] A. Kaveh, A. B. Farahmand, S. Talatahari, Ant colony optimization for design of space trusses, International Journal of Space Structures, Volume 23(2008) 167–81.
[20] J. Kennedy, R. C. Eberhart, Swarm Intelligence, Morgan Kaufmann, ISBN 1-55860-595-9, 2001.
[21] A. Kaveh, S. Talatahari, An improved ant colony optimization for constrained engineering design problems, Engineering Computations in press, 2008.
[22] L. F. F. Miguel, L. F. F. Miguel, Shape and size optimization of truss structures considering dynamic constraints through modern metaheuristic algorithms, Expert Systems with Applications, Volume 39 (2012) 9458–9467.
[23] B. J. Auer, Size and shape optimization of frame and truss structures throuth evolutionary methods, Presented in Partial Fulfillment of the Requirements for the Degree of Master of Science, Major in Mechanical Engineering in the College of Graduate Studies University of Idaho, 2005.
[24] S. Kazemzadeh Azad and A. J. Kulkarni, Structural optimization using amutation-based genetic algorithm, International journal of optimization in civil engineering, Volume 2(2012) 81-101.
[25] A.Ahrari, A. A. Atai, Fully Stressed Design Evolution Strategy for Shape and Size Optimization of Truss Structures, Computers and Structures, Volume 123 (2013) 58–67.
[26] S. Gholizadeh, Layout optimization of truss structures by hybridizing cellular automata and particle swarm optimization, Computers and Structures, Volume 125 (2013) 86–99.