S. Kazemzadeh Azad and S. Kazemzadeh Azad; Structures, 58 (2023) 105409.
This is a draft version. For the final published paper, visit the following link:
https://doi.org/10.1016/j.istruc.2023.105409

A standard benchmarking suite for structural optimization algorithms: ISCSO 2016-2022

Saeid Kazemzadeh Azad[1,*], Sina Kazemzadeh Azad[2]

*[1]Department of Civil Engineering, Atilim University, Ankara, Turkey (*Corresponding author: saeid.azad@atilim.edu.tr, Tel: +90-312-586-8353)*

[2]School of Civil Engineering, The University of Sydney, Sydney, NSW 2006, Australia (sina.kazemzadehazad@sydney.edu.au)

Abstract

Benchmarking is an essential part of developing efficient structural optimization techniques. Despite the advent of numerous metaheuristic techniques for solving truss optimization problems, benchmarking new algorithms is often carried out using a selection of classic test examples which are indeed unchallenging for contemporary sophisticated optimization algorithms. Besides, limited optimization results available in the literature, on new test examples, are usually not accurately comparable. This is typically attributed to the lack of sufficient data on the performance of the investigated algorithms as well as the inconsistencies between the studies in terms of adopted test examples for benchmarking, optimization problem formulation, maximum number of objective function evaluations etc. Accordingly, there exists a need for new standard test suites composed of easily reproducible challenging test examples with rigorous and comparable performance evaluation results of algorithms on these test suites. To this end, the present work aims to propose a new baseline for benchmarking structural optimization algorithms, using a set of challenging sizing and shape optimization problems of truss structures selected from the international student competition in structural optimization (ISCSO) instances. The most recent six structural optimization examples from the ISCSO are tackled using a representative metaheuristic structural optimization algorithm. The statistical results of all the optimization runs using the proposed benchmarking suite are provided to pave the way for more rigorous benchmarking of structural optimization algorithms.

Keywords: Structural optimization; benchmarking suite; metaheuristic algorithms; steel trusses; sizing optimization; shape optimization.

1. Introduction

The environmental and economic significance of structural optimization has been more recognized in the recent years [1]. In the context of structural optimization, design of truss systems for minimum weight or cost has been always an active field of research. This is not only because of the widespread use of these structures in practice but also due to their computational efficiency in providing reasonable conceptual designs for their more complex counterparts. Another reason for the large number of discrete truss optimization studies in the literature is the capability of the optimized instances to represent the complexities of challenging combinatorial optimization problems and thus being good candidates for performance evaluation of optimization algorithms in different disciplines related to engineering optimization, applied mathematics, computer science, etc. The advantages of adopting truss instances for discrete optimization have been comprehensively addressed in [2].

Over the past decades numerous variants of evolutionary optimization algorithms or metaheuristics have been developed and employed for structural optimization purposes [3-5]. In the literature, typically, performance evaluation of newly developed techniques for truss optimization has been accomplished using a selection of popular conventional benchmark instances. The popularity of conventional truss optimization benchmarks can be attributed to the availability of data on these examples as well as the large number of studies in the literature that provide performance evaluation results of previous optimization algorithms on these test examples [6-13]. Although conventional benchmarks form a reliable and unbiased platform for comparison of the optimization algorithms, owing to the growing increase in computational power and efficiency of search strategies, it can easily be observed from the literature that most of the recently

GCRIS

developed metaheuristic algorithms perform well on these test examples. Consequently, often the difference between the solutions obtained using different contemporary algorithms on standard truss optimization benchmarks is insignificant, and it does not contribute to detecting more successful methods among the investigated algorithms.

The use of trivial optimization examples, merely to show the successful performance of a specific algorithm – referred to as an algorithm-seeks-problem approach, contrary to a problem-seeks-algorithm approach [14] – has been criticized in Cohn and Dinovitzer [15]. Accordingly, using more challenging test examples that better reflect the complexities of real-world design instances has been suggested in the literature [16]. In the critical review on truss optimization with discrete variables covering the studies between 1968–2014, Stolpe [2] highlights the urgent need for development of a publicly available benchmark library to promote development and evaluation of truss optimization techniques.

Optimization competitions have long been recognized as fruitful platforms to examine and verify the efficiency of optimization algorithms [17]. For example, optimization test suites proposed in the IEEE Congress on Evolutionary Computation (CEC) (e.g., CEC'2005 [18] benchmark suite) have become well-known benchmark suites adopted by researchers in the literature [19-21]. More information about the CEC benchmark suites can be found in Molina et al. [17] where a decade of competitions on evolutionary optimization algorithms is rigorously outlined. In contrast, in the field of structural optimization, despite pioneering attempts to include more realistic test examples for performance evaluation of truss optimization algorithms [22], still there is a need for establishing standard test suites. Indeed, lack of standard platforms or benchmark libraries for referring to specific collections of truss optimization benchmarks

GCRIS

in structural optimization could be an important yet often overlooked source of inconsistency between the studies. Typically, referring to conventional truss optimization examples using only the term "classic benchmarks" without specifically mentioning the names of included examples, does not imply sufficient information about the adopted collection of test examples. Consequently, authors of different papers, intending to use the classic benchmarks in their studies, often choose different subsets of available benchmarks to demonstrate the performance of the developed optimization algorithms. This leads to difficulties in comparisons of truss optimization algorithms among different studies since often algorithms are verified on different test suites. Borrowing the idea of labeling specific test suits from the IEEE CEC benchmark suites, a practical solution can be specifically denoting the benchmarking suites in structural optimization as well (e.g., ISCSO 2016-2022 benchmark suite). This could yield a simple, standard, and more comprehensible terminology when referring to a specific collection of truss optimization problem instances.

The international student competition in structural optimization (ISCSO) [23] is an annual online competition which began over a decade ago. Beside encouraging students and researchers to solve engineering optimization problems, the competition also forms an international platform for researchers to readily examine the performance of their optimization algorithms using the provided test problems. The early test examples of the competition have been already gathered in [24], some of which, i.e., ISCSO 2011 and 2012 optimization problems, have been optimized by Kale and Kulkarni [25] using a hybrid metaheuristic with self-adaptive penalty function. There has been a growing interest in the use of ISCSO optimization problems in the recent literature which is reviewed in the following.

GCRIS

Recent studies on evolutionary optimization algorithms that use the ISCSO optimization problems [26, 28-30, 32] clearly demonstrate the challenging nature of the ISCSO optimization problems and render them potential benchmarks for performance evaluation of structural optimization techniques. Albert and Zhang [26] proposed SpartaPlex, as a novel black-box optimization algorithm, and employed two challenging optimization problems of the ISCSO 2018 and 2019 for performance evaluation of their technique versus different optimization algorithms. The authors concluded that best solutions known for the ISCSO 2018 and 2019 examples i.e., 14425.0973 kg and 12329.1302 kg (winner design weights of ISCSO 2018 and 2019), respectively, are approximately 24% and 29% lighter than the best results obtained across all trials and formulations presented in [26]. It is stated that although the proposed SpartaPlex outperforms the other optimization algorithms, the best solutions known are significantly lighter.

To generate a practical large-scale optimization test example, Ghosh et al. [27] used a modified version of the ISCSO 2019 truss optimization problem with a different loading condition than that of the original problem. The objective of the study was to integrate user knowledge into the optimization process to reduce the computational cost of multi-objective design optimization problems. Although the foregoing modified version of the ISCSO 2019 problem forms a large-scale practical test example, it does not fully reflect the difficulty of the original version basically due to the reduced number of loading conditions as well as the smaller number of design variables in the modified version. Etaati et al. [28] proposed an optimization framework composed of 12 different metaheuristic algorithms and performed a comparative study using the ISCSO 2018 and 2019 truss optimization problems, namely 314 and 260-member truss structures, respectively. The authors highlighted that the complex nature of the constraints as well as the large number of

GCRIS

design variables, involved in the foregoing examples, have resulted in challenging test cases for optimization algorithms. Moreover, the proposed comparative optimization framework revealed that, to deal with the investigated test examples, the metaheuristics used in the study need to be further enhanced in terms of constraint handling.

Dehkordi et al. [29] developed an adaptive metaheuristic technique for large-scale design optimization and compared its performance versus 14 modern metaheuristic optimization algorithms. For this purpose, the authors used well-known mathematical test functions as well as three design optimization problems of the ISCSO 2018, 2019, and 2021, namely 314-member, 260-member, and 345-member truss structures, respectively. In their study, since the allowable maximum number of objective function evaluations for the algorithms was set to a different value than that of the original ISCSO test examples, further comparison of the obtained results with the best feasible solutions known for the ISCSO examples is not possible.

For structural optimization applications, Kaveh and Biabani Hamedani [30] proposed an improved version of a newly developed metaheuristic method called arithmetic optimization algorithm (AOA) [31]. The authors demonstrated successful performance of the developed improved variant of the arithmetic optimization algorithm (IAOA) on conventional structural optimization benchmarks. However, in further performance evaluation of the proposed IAOA using the ISCSO 2017 test instance, it was observed that the performance of the algorithm deteriorated. The authors reported that, for the ISCSO 2017 problem instance, the metaheuristic methods used in the study, namely AOA, IAOA, and PSO, were unable to converge to a feasible design at the end of optimization process.

A recent state-of-the-art review by Lagaros et al. [32] on the performance evaluation

GCRIS

of metaheuristic optimization algorithms in structural optimization clearly shows the complexity of the ISCSO test examples versus conventional structural optimization benchmarks. The authors employed twenty-four metaheuristic algorithms for solving eleven single objective structural optimization test examples. First, the authors investigated the performance of metaheuristics on six well-known benchmark structural optimization problems from the literature including three well-known benchmark truss sizing optimization problems i.e., 10-bar truss, 25-bar truss, and 72-bar truss structures with 10, 8, and 16 design variables, respectively. Next, the performance evaluation of metaheuristics was carried out using five challenging optimization test examples of ISCSO 2015-2019 with design variables ranging from 54 to 328. The study revealed that although most of the investigated twenty-four metaheuristic algorithms were successful in handling the well-known conventional structural optimization benchmarks, their performances were not promising in the case of five test problems taken from the ISCSO 2015-2019.

The inferior performance of majority of the contemporary metaheuristics on the ISCSO problem instances reported in the recent literature indicates the complexity of these examples and highlights the importance of algorithm selection and parameter tuning in metaheuristics. Obviously, with development of more competitive standard benchmarking suites for performance assessment, further hyperparameter tuning and improvements in the formulations of the selected algorithms, better results could be achieved. On the other hand, typically there exists some difficulties when replicating new test examples for performance evaluation of structural optimization algorithms. In contrast to concise formulations of standard mathematical functions commonly used as real-parameter function optimization benchmarks [18], geometrical modelling, analysis, and design of structural systems needs further data associated with geometry, material,

GCRIS

loading, support conditions, strength and serviceability constraints, etc. that should be rigorously treated to prevent any inconsistency in the comparison of the results. To alleviate the burden of modelling, simulation, and design of the structural optimization examples, ready-to-use MATLAB [33] functions are publicly available to readily regenerate the investigated ISCSO test examples [23]. These coded functions can reduce the user error in replicating the structural optimization examples and provide unique platforms for performance evaluation of truss optimization algorithms.

In the present work, it is aimed to form a challenging, easily reproducible, and accurately comparable benchmarking suite for structural optimization algorithms using the ISCSO 2016-2022 test examples. To this end, the exponential big bang-big crunch (EBB-BC) algorithm [34] is used as a representative metaheuristic method to solve the aforementioned optimization test examples. Three different configurations of the algorithm have been run numerous times and the obtained statistical results are presented for future comparisons with existing and new optimization algorithms. To facilitate later comparisons, the statistical results are presented for sufficiently large number of 30 independent optimization runs. Finally, based on the numerical experiments, certain concluding remarks are provided for future studies. It is worth mentioning that, in the present work, a comparison of the performance of the EBB-BC algorithm versus other metaheuristics is not intended. The logic behind presenting the statistical results of the EBB-BC algorithm is to provide a baseline for benchmarking structural optimization algorithms using the proposed standard test suite.

This paper is organized as follows. Section 2 outlines the design optimization problem formulation. Section 3 briefly describes the utilized representative structural optimization algorithm. Section 4 elaborates the ISCSO structural analysis, design, and

GCRIS

visualization functions. Section 5 presents the numerical experiments and obtained results using the ISCSO 2016-2022 problem instances. The concluding remarks are provided in the last section.

2. Problem formulation

In practical applications, structures shall be designed in accordance with the regulations and design constraints stipulated by standard design codes. This section provides the general mathematical formulation for single-objective truss optimization problem with respect to AISC-LRFD [35]. The objective of the optimization process is to find a solution vector ***X*** (Eq. 1) denoting the involved *n* design variables,

$$X^T = \left[x_1, x_2, \ldots, x_n\right] \tag{1}$$

such that ***X*** minimizes the corresponding weight of the truss structure:

$$W = \sum_{i=1}^{N_m} \rho_i L_i A_i \tag{2}$$

where W is the net weight of the truss, ρ_i, L_i, A_i are unit weight, length, and cross-sectional area of the *i-th* member, respectively. The above-mentioned weight minimization problem shall be solved subject to the following strength and displacement constraints. With respect to AISC-LRFD [35], the following relation must be satisfied for the strength requirement of each truss member.

$$\left[\frac{P_u}{\phi P_n}\right]_i - 1 \le 0 \tag{3}$$

where P_u and P_n are the required and nominal axial (tensile or compressive) strengths of

GCRIS

the *i-th* truss member, respectively. Here, ϕ is the resistance factor for axial strength, taken as 0.85 for compression and 0.9 for tension.

According to AISC-LRFD [35], the nominal tensile strength of a truss member, based on yielding in the gross cross section, is computed using the following equation:

$$P_n = F_y A_g \tag{4}$$

where F_y is the member's specified yield stress and A_g is the gross cross section of the member. Moreover, the nominal compressive strength of members with compact and/or non-compact elements, for the limit state of flexural buckling is determined as follows:

$$P_n = F_{cr} A_g \tag{5}$$

where F_{cr} is the critical stress based on flexural buckling of the member calculated as:

$$\text{for} \quad \lambda_c = \frac{K l}{r \pi}\sqrt{\frac{F_y}{E}} \le 1.5 \qquad F_{cr} = (0.658^{\lambda_c^2})F_y \tag{6}$$

$$\text{for} \quad \lambda_c = \frac{K l}{r \pi}\sqrt{\frac{F_y}{E}} > 1.5 \qquad F_{cr} = \left[\frac{0.877}{\lambda_c^2}\right]F_y \tag{7}$$

In the foregoing equations, l is the laterally unbraced length of the member, K is the effective length factor, r is the governing radius of gyration about the buckling axis and E is the modulus of elasticity. Moreover, for handling the displacement constraints, the following displacement criterion shall be satisfied:

$$\frac{d_{j,k}}{(d_{j,k})_{all}} - 1 \le 0 \tag{8}$$

where j = 1, 2, ... , N_j is the joint number, N_j is the total number of joints, $d_{j,k}$, and $(d_{j,k})_{all}$,

GCRIS

are the displacements computed in the *k-th* direction of the *j-th* joint and the corresponding maximum allowable value, respectively. It is worth mentioning that beside the above-mentioned general problem statement, further details on optimization problem formulation for each test example can be found in [23].

3. Structural optimization algorithm

To provide statistical results for future comparisons of structural optimization algorithms, a representative metaheuristic optimization algorithm, namely EBB-BC [34], is selected in the present work. The standard version of the big bang-big crunch (BB-BC) optimization method has first appeared in Erol and Eksin's study [36]. Dut to its efficiency and algorithmic simplicity, the algorithm has found plenty of applications in the literature of structural optimization [37]. An enhanced variant of the BB-BC algorithm, namely EBB-BC [34], demonstrated a promising performance in discrete structural design optimization problems. The EBB-BC algorithm is a population-based metaheuristic technique which uses an elitist evolutionary scheme where at every iteration, the current best design, I^c, is used to generate a new population of candidate designs using the following relation:

$$I_i^{new} = I_i^c \pm round\left[\alpha \cdot Exp(\lambda = 1)_i^3 \frac{(I_i^{\max} - I_i^{\min})}{iter}\right] \tag{9}$$

where I_i^{new} and I_i^c are the value of *i*-th discrete design variable in the new and fittest candidate solutions, and $I_i^{\min}$ and $I_i^{\max}$ are the corresponding lower and upper bounds, respectively. Here, *iter* denotes the iteration number, and α is a scaling constant set to 0.25 in the present study. As can be seen from the equation, in the formulation of the EBB-BC algorithm the third power of a random number, generated using an exponential distribution (*Exp*), is favored. Moreover, for handling the design constraints a simple

GCRIS

static penalty function is used with a constant penalty coefficient of unity [34]. Detailed descriptions of the main steps for implementation of the EBB-BC optimization algorithm can be found in Ref. [34].

In the present work, owing to the promising performance of the EBB-BC, specifically in design optimization of steel structures, it is selected to optimize the ISCSO 2016-2022 challenging design examples. For each problem instance, with respect to the allowed maximum number of objective function evaluations, three different configurations of the EBB-BC algorithm, namely EBB-BC$_{(25)}$, EBB-BC$_{(50)}$, and EBB-BC$_{(100)}$, are implemented with population sizes of 25, 50, and 100, respectively. The corresponding parameter setting for three different configurations of the EBB-BC algorithm is presented in Table 1.

Table 1: Parameter setting for different configurations of the EBB-BC algorithm

Test example	Optimization algorithm	Population size	Maximum number of iterations	Objective function evaluations
ISCSO (2016)	EBB-BC$_{(25)}$	25	800	20000
	EBB-BC$_{(50)}$	50	400	20000
	EBB-BC$_{(100)}$	100	200	20000
ISCSO (2017)	EBB-BC$_{(25)}$	25	6000	150000
	EBB-BC$_{(50)}$	50	3000	150000
	EBB-BC$_{(100)}$	100	1500	150000
ISCSO (2018)	EBB-BC$_{(25)}$	25	8000	200000
	EBB-BC$_{(50)}$	50	4000	200000
	EBB-BC$_{(100)}$	100	2000	200000
ISCSO (2019)	EBB-BC$_{(25)}$	25	8000	200000
	EBB-BC$_{(50)}$	50	4000	200000
	EBB-BC$_{(100)}$	100	2000	200000
ISCSO (2021)	EBB-BC$_{(25)}$	25	8000	200000
	EBB-BC$_{(50)}$	50	4000	200000
	EBB-BC$_{(100)}$	100	2000	200000
ISCSO (2022)	EBB-BC$_{(25)}$	25	8000	200000
	EBB-BC$_{(50)}$	50	4000	200000
	EBB-BC$_{(100)}$	100	2000	200000

4. ISCSO structural analysis, design, and visualization functions

Typically, for benchmarking new structural optimization algorithms on conventional truss optimization examples, one needs to gather and rigorously implement the associate data with modelling, loading, design variables and constraints. This procedure inherently is prone to inconsistencies between the original instances and the regenerated models due to either the user induced errors or lack of sufficient data related to geometry, material, loading, design constraints, etc. An alternative to constructing a test example from scratch using the available data in the literature, is to provide the user with ready to use functions for each test example wherein all structural data associated with geometry, material, loading, analysis, and design constrains are implemented. For this purpose, ISCSO website [23] provides simple MATLAB functions which can be readily used integrated with new optimization techniques (to facilitate the access to the functions, they are also provided as supplementary material). For example, analysis and design of the ISCSO 2018 optimization test example, i.e., 314-member truss structure, anywhere in an optimization code, can easily be carried out using the following command:

`[Weight, Const_Vio_Stress, Const_Vio_Disp] = ISCSO_2018(Sections, Coordinates, Flag)` (10)

Here, the terms inside the parentheses on the right-hand side of the function represent the input values which should be provided to the function while the terms inside the brackets on the left-hand side of the function indicate the output variables which will be computed by the function. In the above function, “Sections” and “Coordinates” denote the sizing and shape design variables, respectively. “Flag” has only two possible values i.e., either 0 or 1. The value of zero indicates that no graphical output is requested from the function. For computational efficiency, this option is commonly used during the optimization iterations where graphical output is not necessary. In contrast, if Flag is set

GCRIS

to 1, the function will plot the undeformed geometry of the given structure. In the above function, as the terms imply, “Weight”, “Const_Vio_Stress”, and “Const_Vio_Disp” represent the net weight, stress and displacement constraint violations, respectively.

Ensuring the feasibility of the optimized final designs by satisfying all the imposed design constraints is an important criterion in structural optimization. One of the typical problems, often encountered in the literature of truss optimization, is the existence of infeasibilities in the reported final designs basically due to the inconsistencies between the problem formulations of the generated test examples and the original benchmarks. This sometimes results in erroneous comparisons between feasible and infeasible solutions achieved by different optimization algorithms for a specific problem instance. To overcome this difficulty, the foregoing ISCSO functions, on the one hand, provide a unique problem formulation that eliminates the probable inconsistencies in formulating the optimization constraints and, on the other hand, help to readily detect any kind of constraint violation through visualizing the infeasibilities of the candidate designs. As shown in Figure 1, in the graphical output of the ISCSO functions, the nodes and members that violate the displacement and stress constraints, respectively, are colored in red.

GCRIS

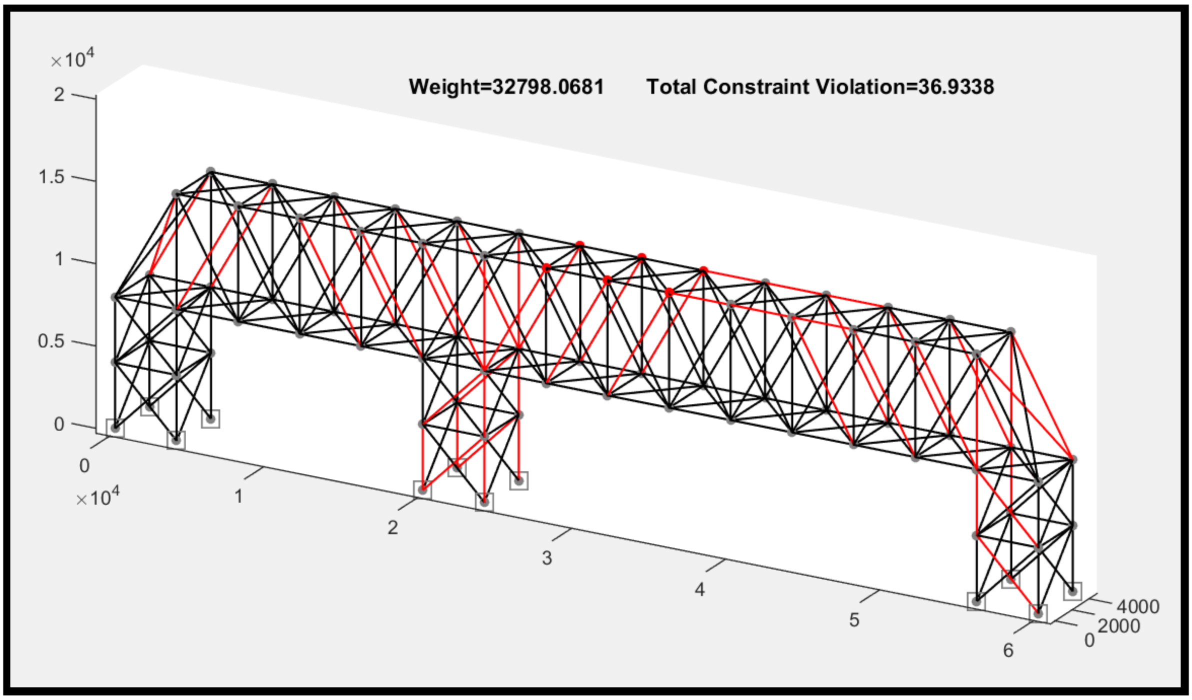


Figure 1: Visualization of the strength and serviceability constraint violations for an arbitrary design using the ISCSO 2018 function

5. Numerical experiments

To examine the performance of the EBB-BC metaheuristic algorithm, six challenging optimization problems, namely 117-member truss structure (ISCSO 2016), 198-member truss structure (ISCSO 2017), 314-member truss structure (ISCSO 2018), 260-member truss structure (ISCSO 2019), 345-member truss structure (ISCSO 2021), and 336-member free-form truss structure (ISCSO 2022), are chosen from ISCSO 2016-2022 instances. To ease the future comparisons, it is aimed to provide statistical results based on sufficiently large number of 30 independent runs considering the computational cost of the problems. All the test examples are solved with respect to the original problem statements and using the same number of maximum objective function evaluations of the original test examples stated in [23]. A summary of the design optimization examples is tabulated in Table 2. Although a brief description of each test example is provided in the

GCRIS

following, more information about the design optimization examples can be found in [23].

Table 2: Summary of the design optimization test examples

Test example	Design variables	Sizing variables	Shape variables	Objective function evaluations	*ISCSO winner solution (kg)
ISCSO (2016)	124	117	7	20000	2816.0281
ISCSO (2017)	211	198	13	150000	44090.5356
ISCSO (2018)	328	314	14	200000	14425.0973
ISCSO (2019)	270	260	10	200000	12329.1302
ISCSO (2021)	345	345	0	200000	3977.2609
ISCSO (2022)	336	336	0	200000	5976.5057

*The design weight reported by the winner of the ISCSO [23].

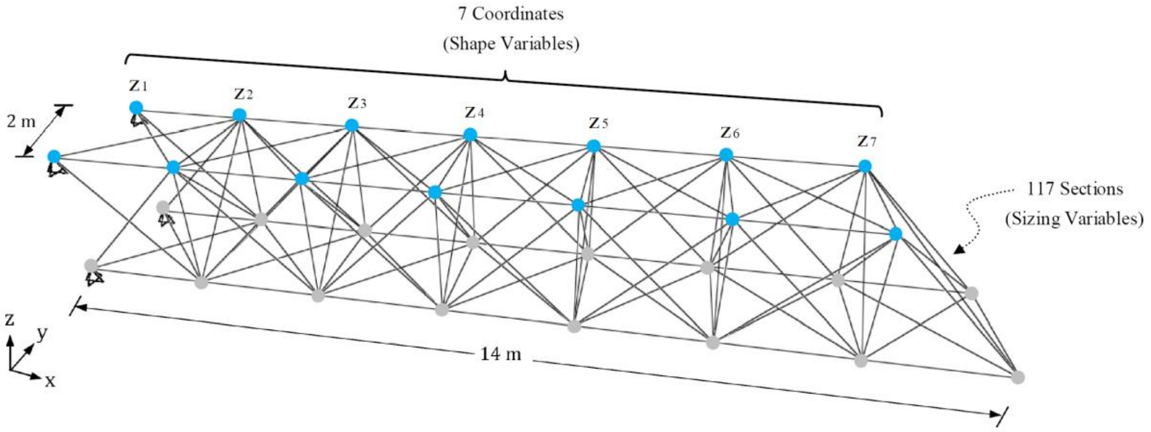


Figure 2: 117-member cantilever truss (ISCSO 2016)

The 117-member truss structure shown in Figure 2 is adopted from the ISCSO 2016. This problem includes 117 sizing variables (which represent the cross-sections of the truss members) as well as 7 shape variables (which are the vertical coordinates of the top blue nodes of the structure) resulting in a total of 124 design variables. It should be emphasized that each pair of the blue nodes with the same x-coordinate will have the same height. For this optimization problem, the maximum number of objective function evaluations performed by the algorithms is limited to 20000. The best feasible design weight which is reported by the winner of the ISCSO 2016 for this optimization problem is 2816.0281 kg [23].

Simultaneous sizing and geometry optimization of the 117-member truss structure is carried out using three different configurations of the EBB-BC algorithm, namely EBB-BC$_{(25)}$, EBB-BC$_{(50)}$, and EBB-BC$_{(100)}$ and the obtained best feasible solutions over 30 independent runs are presented in Table 3. As can be seen from Table 3, for the ISCSO 2016 test example EBB-BC$_{(25)}$ obtains a design weight of 2882.7 kg over 30 optimization runs which is lighter than those of EBB-BC$_{(50)}$, and EBB-BC$_{(100)}$ which are 3081.3 and 3403.1 kg, respectively. In order to better demonstrate the solution quality of the achieved optimization results, the obtained best design weights are normalized (denoted as normalized solution quality) through dividing the best solution known for each test example (i.e., the corresponding design weight reported by the winner of the ISCSO [23]) by the best solution found here by the optimization algorithm. Accordingly, as presented in Table 3, for the ISCSO 2016 problem instance, a reference design weight of 2816.0281 kg [23] is used to normalize the results of all the employed three configurations of the EBB-BC algorithm. Consequently, for this test example, EBB-BC$_{(25)}$ yields a better normalized solution quality of 0.98 than those of EBB-BC$_{(50)}$, and EBB-BC$_{(100)}$ which are 0.91, and 0.83, respectively. The optimized geometry obtained using EBB-BC$_{(25)}$ for the ISCSO 2016 test problem is depicted in Figure 3. The convergence histories of all the independent runs, using EBB-BC$_{(25)}$ for ISCSO 2016 test problem, are depicted in Figure 4. It is worth mentioning that the average convergence curve shown by black dashed line in the figure – which represents the variation of the average feasible design weight of all runs – may not necessarily be an average of all the 30 runs in the early iterations because different optimization runs may reach the first feasible solution at different iterations.

GCRIS

Table 3: Feasible design weights obtained over independent optimization runs (ISCSO 2016-2017)

Run no.	**ISCSO (2016)**			**ISCSO (2017)**		
	EBB-BC$_{(25)}$	EBB-BC$_{(50)}$	EBB-BC$_{(100)}$	EBB-BC$_{(25)}$	EBB-BC$_{(50)}$	EBB-BC$_{(100)}$
1	3081.7	3226.9	3867.2	49127.4	51416.7	52628.3
2	3108.4	3219.1	3704.8	53034.5	47627.8	54551.4
3	3050.0	3135.7	3946.0	58041.2	53671.6	50227.3
4	2882.7	3144.4	3471.9	56794.9	53605.9	49934.5
5	2946.0	3210.5	3793.2	52740.8	48583.4	52456.7
6	3199.3	3240.4	3596.7	59049.5	49023.4	53469.5
7	3216.8	3300.3	3809.7	56580.6	50696.6	53170.7
8	3474.9	3380.3	4003.4	50986.6	53612.4	47461.7
9	3067.7	3132.8	3713.8	51197.3	59162.4	56727.3
10	2969.3	3498.5	3984.6	52578.0	50622.5	52825.3
11	3166.6	3213.8	3817.2	49903.4	56943.1	48299.4
12	3185.0	3320.2	3829.1	54446.0	47740.1	47486.5
13	3139.9	3552.6	3808.2	52728.1	49292.9	50634.3
14	3026.2	3279.3	3661.4	54700.9	53499.6	48849.3
15	3138.3	3342.9	4462.0	47746.4	51996.6	48858.9
16	3122.2	3377.6	3757.4	50448.9	53380.6	49894.4
17	3080.8	3309.3	3631.9	53749.6	47683.3	49983.4
18	2930.0	3488.9	4271.0	53794.8	54841.6	49420.0
19	3021.7	3290.3	3910.5	53635.1	47466.7	47768.0
20	3051.7	3221.5	3603.3	52066.8	49817.7	47800.3
21	2979.3	3081.3	3671.1	56782.3	49637.5	54866.1
22	3031.6	3259.5	3757.1	56984.7	49321.8	56060.2
23	3157.8	3373.3	3962.6	52101.3	52428.7	48226.3
24	3068.1	3536.7	3596.7	51576.0	48173.9	52191.2
25	2982.6	3438.0	3582.8	53070.1	50599.5	53219.7
26	3055.6	3459.6	3532.5	51634.9	51246.0	60238.8
27	3073.4	3196.8	3893.7	50873.8	47537.8	54836.9
28	2918.5	3605.6	3776.6	51379.0	62456.6	50840.3
29	3066.2	3251.8	3842.2	49821.4	47689.7	51451.9
30	3014.1	3305.1	3403.1	54038.9	49938.9	50381.3
Best weight (kg)	2882.7	3081.3	3403.1	47746.4	47466.7	47461.7
Worst weight	3474.9	3605.6	4462.0	59049.5	62456.6	60238.8
Mean weight	3073.5	3313.1	3788.7	53053.8	51323.8	51492.0
Standard deviation	113.7	134.8	219.6	2730.0	3607.5	3112.9
Coefficient of variation (%)	3.7	4.1	5.8	5.1	7.0	6.0
No. analyses	20000	20000	20000	150000	150000	150000
*ISCSO winner solution (kg)	2816.0281	2816.0281	2816.0281	44090.5356	44090.5356	44090.5356
Normalized solution quality	**0.98**	**0.91**	**0.83**	**0.92**	**0.93**	**0.93**

*The design weight reported by the winner of the ISCSO [23].

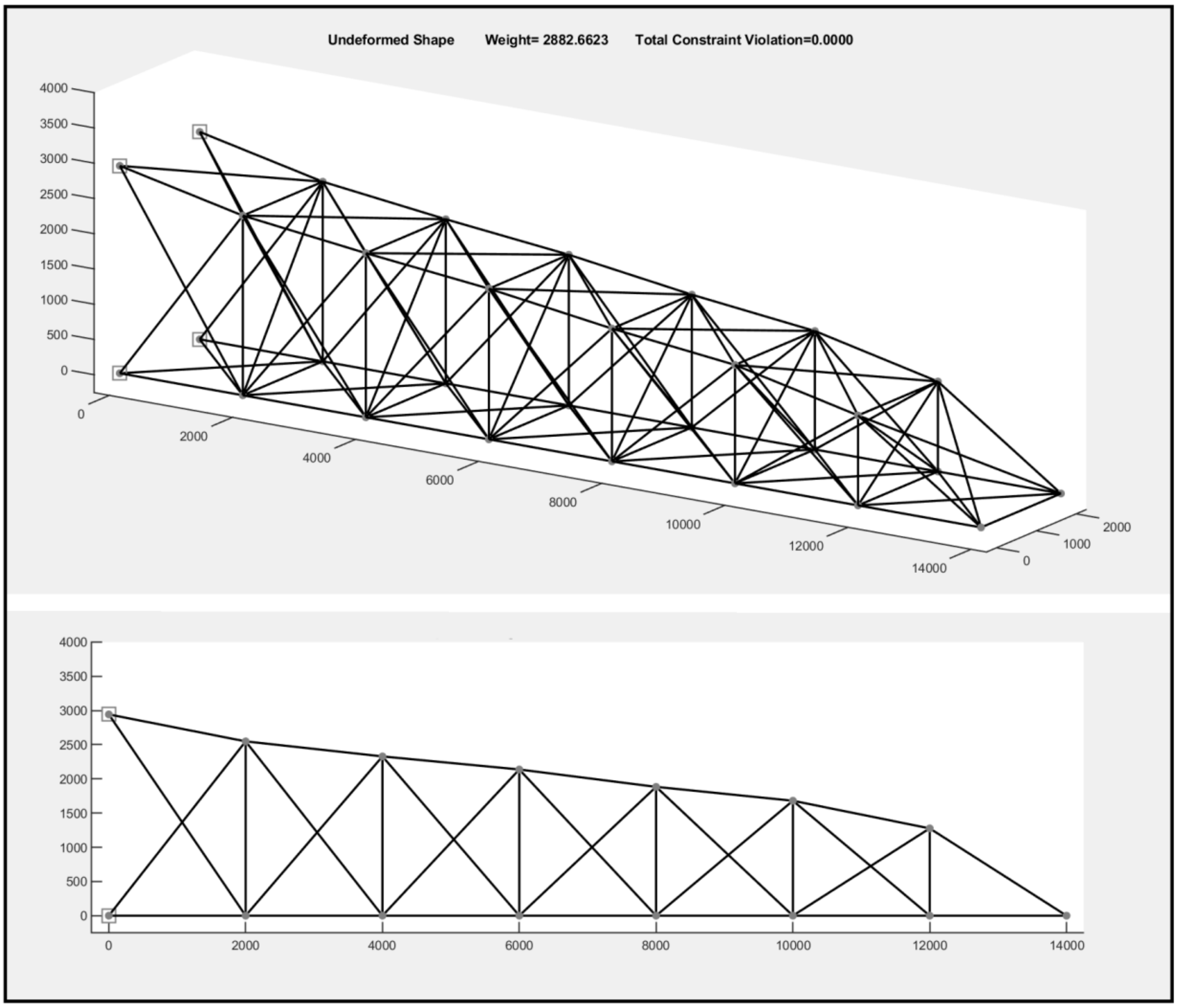


Figure 3: Optimized geometry obtained using EBB-BC(25) for ISCSO 2016 test problem

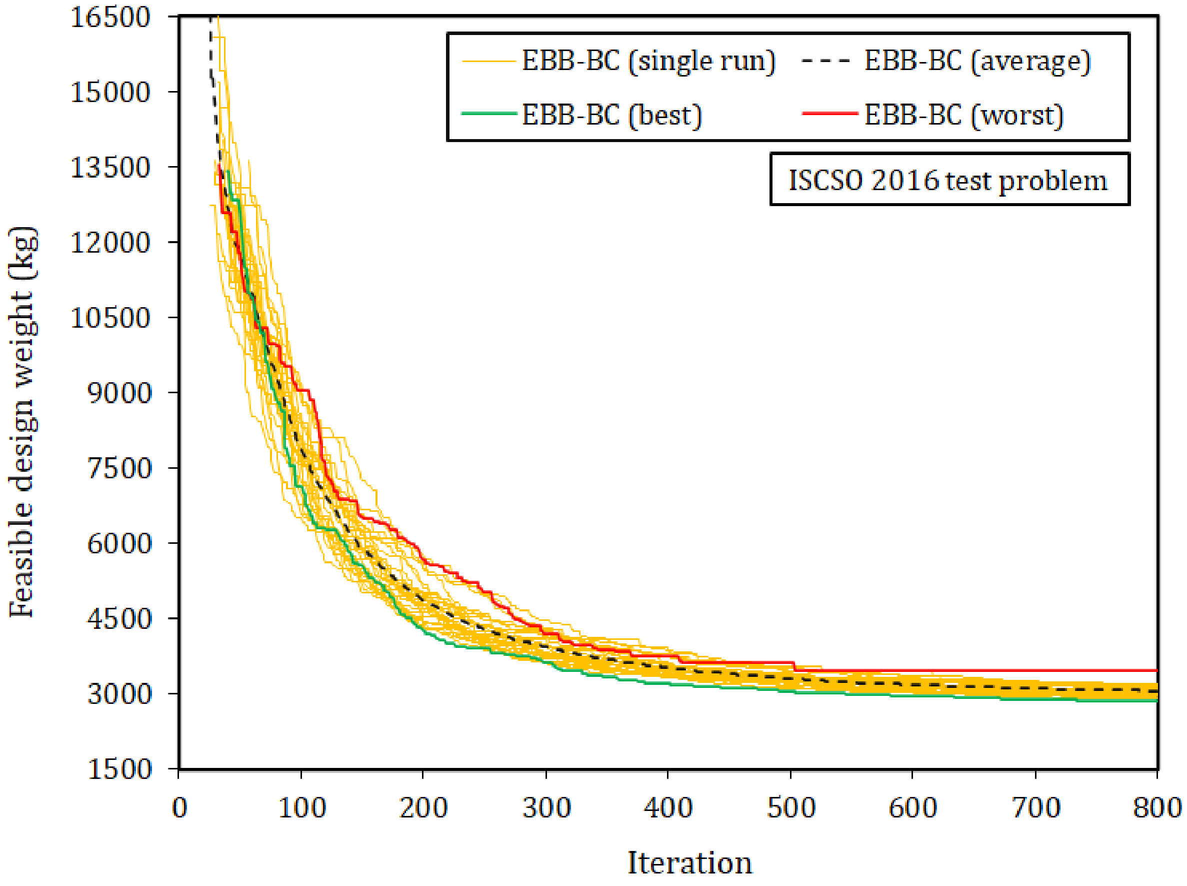


Figure 4: Convergence curves of 30 runs using EBB-BC(25) for ISCSO 2016 test problem

GCN

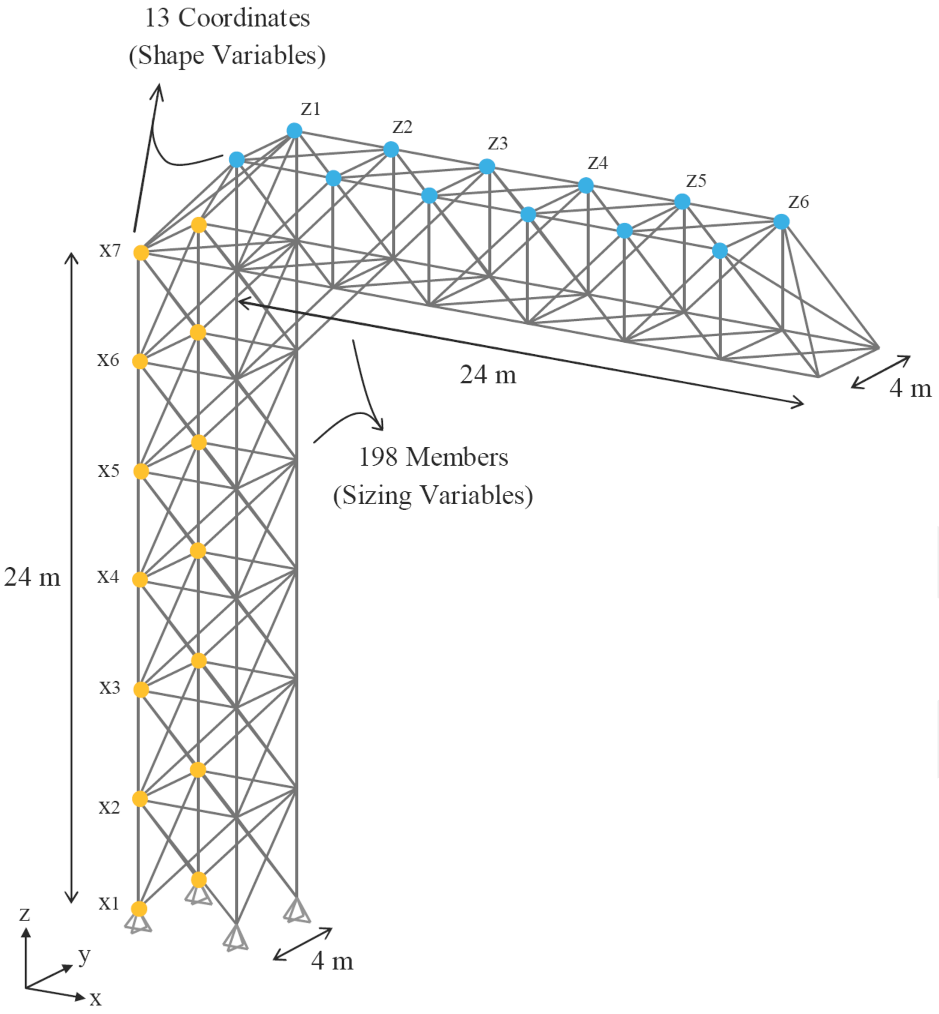


Figure 5: 198-member truss structure (ISCSO 2017)

Figure 5 shows the 198-member truss structure selected from the ISCSO 2017. This optimization problem includes 198 sizing variables (which represent the cross-sections of the truss members) and 13 shape variables (which are the z-coordinates of the top blue nodes as well as the x-coordinates of the side orange nodes of the structure) resulting in a total of 211 design variables. It should be emphasized that each pair of blue nodes with the same x-coordinate will have the same height. Similarly, each pair of orange nodes with the same height will have the same x-coordinate. For this test example, the maximum number of objective function evaluations performed by the algorithms is limited to 150000. The best feasible design weight which is reported by the winner of the ISCSO

2017 for this optimization problem is 44090.5356 kg [23].

Design optimization of the 198-member truss structure is performed using EBB-BC$_{(25)}$, EBB-BC$_{(50)}$, and EBB-BC$_{(100)}$ and the obtained solutions through independent optimization runs are given in Table 3. As presented in the table, for the ISCSO 2017 test example EBB-BC$_{(100)}$ algorithm obtains a design weight of 47461.7 kg which is slightly better than the solutions achieved by EBB-BC$_{(25)}$, and EBB-BC$_{(50)}$ i.e., 47746.4 and 47466.7 kg, respectively. Regarding the existing best result for this test example from the ISCSO 2017, it can be observed from the optimization results that all the employed EBB-BC configurations could achieve promising solutions for ISCSO 2017 problem instance with normalized solution qualities ranging between 0.92 and 0.93. Figure 6 illustrates the final optimized shape of the structure obtained using EBB-BC$_{(100)}$ for the ISCSO 2017 test problem.

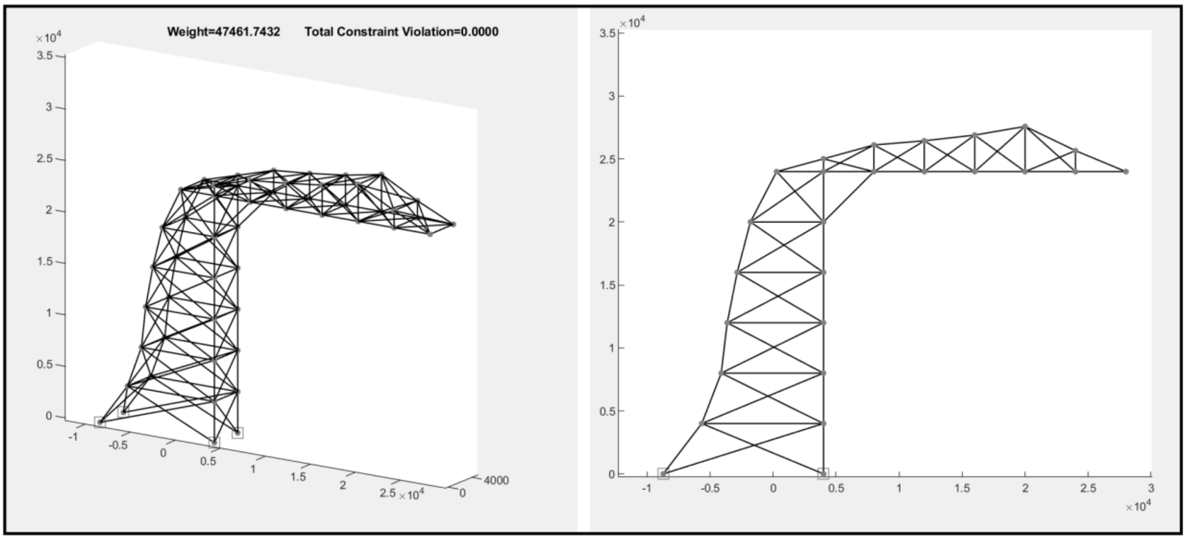


Figure 6: Optimized geometry obtained using EBB-BC$_{(100)}$ for ISCSO 2017 test problem

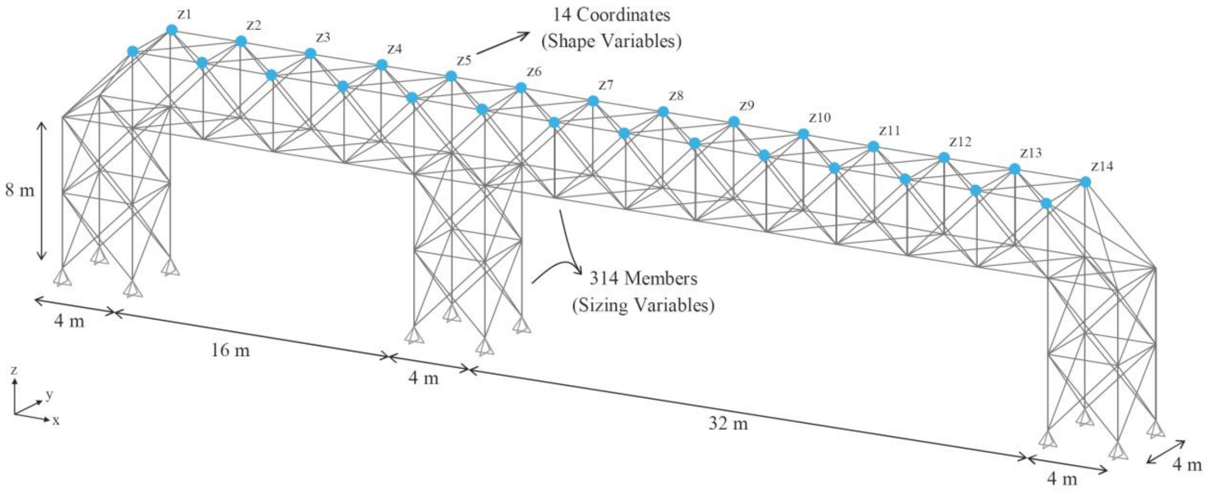


Figure 7: 314-member truss structure (ISCSO 2018)

The 314-member truss structure depicted in Figure 7 is chosen from the ISCSO 2018. This design optimization problem includes 314 sizing variables (which represent the cross-sections of the truss members) as well as 14 shape variables (which are the z-coordinates of the top blue nodes of the structure) resulting in a total of 328 design variables. It should be emphasized that each pair of blue nodes with the same x-coordinate will have the same height. For this test example, the maximum number of objective function evaluations carried out by the optimization algorithms is limited to 200000. The best feasible design weight which is reported by the winner of the ISCSO 2018 for this optimization problem is 14425.0973 kg [23].

Structural optimization of the 314-member truss structure is performed using EBB-BC$_{(25)}$, EBB-BC$_{(50)}$, and EBB-BC$_{(100)}$ and the obtained solutions over independent optimization runs are tabulated in Table 4. As can be seen from Table 4, for the ISCSO 2018 test example EBB-BC$_{(50)}$ algorithm obtains a design weight of 17934.3 kg which is lighter than the design weights achieved by EBB-BC$_{(25)}$, and EBB-BC$_{(100)}$ namely, 18906.3 and 18704.7 kg, respectively. Considering a reference design weight of 14425.0973 kg [23] from the ISCSO 2018 as the best solution known for this test example, it can be

GCRIS

observed from the optimization results that EBB-BC$_{(25)}$, EBB-BC$_{(50)}$, and EBB-BC$_{(100)}$ achieve normalized solution qualities of 0.76, 0.80, and 0.77, respectively. Figure 8 shows the optimized geometry obtained using EBB-BC$_{(50)}$ for the ISCSO 2018 optimization test problem. The convergence curves of independent runs using EBB-BC$_{(50)}$ are plotted for this test example in Figure 9.

Table 4: Feasible design weights obtained over independent optimization runs (ISCSO 2018-2019)

Run no.	**ISCSO (2018)**			**ISCSO (2019)**		
	EBB-BC$_{(25)}$	EBB-BC$_{(50)}$	EBB-BC$_{(100)}$	EBB-BC$_{(25)}$	EBB-BC$_{(50)}$	EBB-BC$_{(100)}$
1	21916.0	19101.9	28816.6	21732.7	17157.3	30429.5
2	21783.2	26245.1	21258.3	17878.0	20435.7	18510.7
3	22161.8	18700.3	22676.0	25800.7	28548.4	18375.8
4	22342.8	22041.6	20486.7	20760.4	18021.4	16682.2
5	23957.8	19567.5	22394.2	18678.5	16730.6	23843.4
6	23723.6	19886.0	21044.1	19452.9	17643.9	18685.5
7	19671.2	17934.3	19121.6	21412.4	17995.1	16961.5
8	20467.3	18960.8	20647.9	26010.2	18276.6	19496.9
9	26649.8	18423.2	19080.7	24310.5	21915.4	19372.8
10	22685.8	18528.9	21363.8	19320.6	17117.4	21893.3
11	26355.4	22833.0	21786.7	19251.1	18846.7	21249.6
12	19413.7	23237.2	24848.1	20245.2	20896.5	17402.7
13	24492.4	21361.3	19963.0	20314.7	23129.6	18782.6
14	25383.6	23122.6	19548.0	20147.3	17337.9	20810.9
15	18906.3	19229.4	19366.8	25474.7	19450.5	18257.4
16	19750.6	25896.2	30630.3	19808.5	17157.9	26344.0
17	24646.2	18567.4	18704.7	21914.5	18576.4	16531.5
18	20942.3	24130.7	22321.3	19007.5	21441.4	16810.8
19	28408.1	21491.4	22829.3	19232.4	17723.1	17722.5
20	23564.5	17989.5	22166.0	24021.7	17767.6	18500.3
21	21257.8	27459.5	19391.2	20713.0	22139.9	16999.3
22	27707.3	21225.1	20452.5	28993.2	17266.5	16233.5
23	20052.8	18824.9	22703.6	17926.2	17953.7	41927.3
24	27349.4	19533.5	24191.8	24916.7	22434.5	16964.3
25	23153.9	21187.9	20186.9	21272.6	18934.9	16843.4
26	22018.0	20246.9	19970.6	18111.9	17491.0	18260.5
27	21379.6	25238.6	20031.3	20606.8	19969.4	17483.3
28	24073.4	18830.2	20669.3	21150.9	15810.4	24708.6
29	23033.9	20933.7	19267.8	25008.4	23587.2	22540.5
30	28202.8	19692.0	21655.7	24407.6	29500.8	37815.1
Best weight (kg)	18906.3	17934.3	18704.7	17878.0	15810.4	16233.5
Worst weight	28408.1	27459.5	30630.3	28993.2	29500.8	41927.3
Mean weight	23181.7	21014.0	21585.8	21596.1	19708.6	20881.3

Standard deviation	2726.9	2664.3	2699.1	2864.3	3268.8	6139.8
Coefficient of variation (%)	11.8	12.7	12.5	13.3	16.6	29.4
No. analyses	200000	200000	200000	200000	200000	200000
*ISCSO winner solution (kg)	14425.0973	14425.0973	14425.0973	12329.1302	12329.1302	12329.1302
Normalized solution quality	**0.76**	**0.80**	**0.77**	**0.69**	**0.78**	**0.76**

*The design weight reported by the winner of the ISCSO [23].

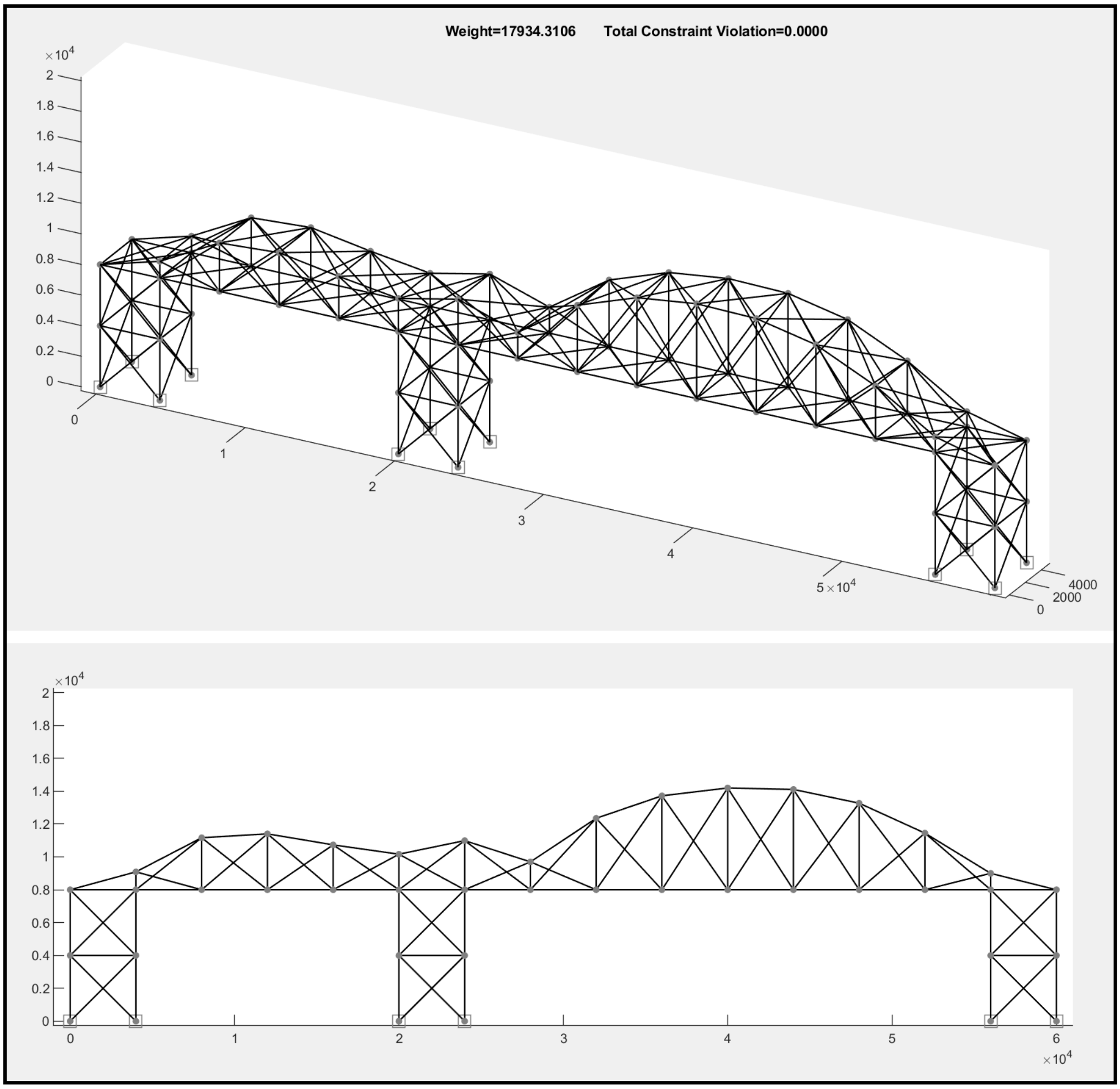


Figure 8: Optimized geometry obtained using EBB-BC$_{(50)}$ for ISCSO 2018 test problem

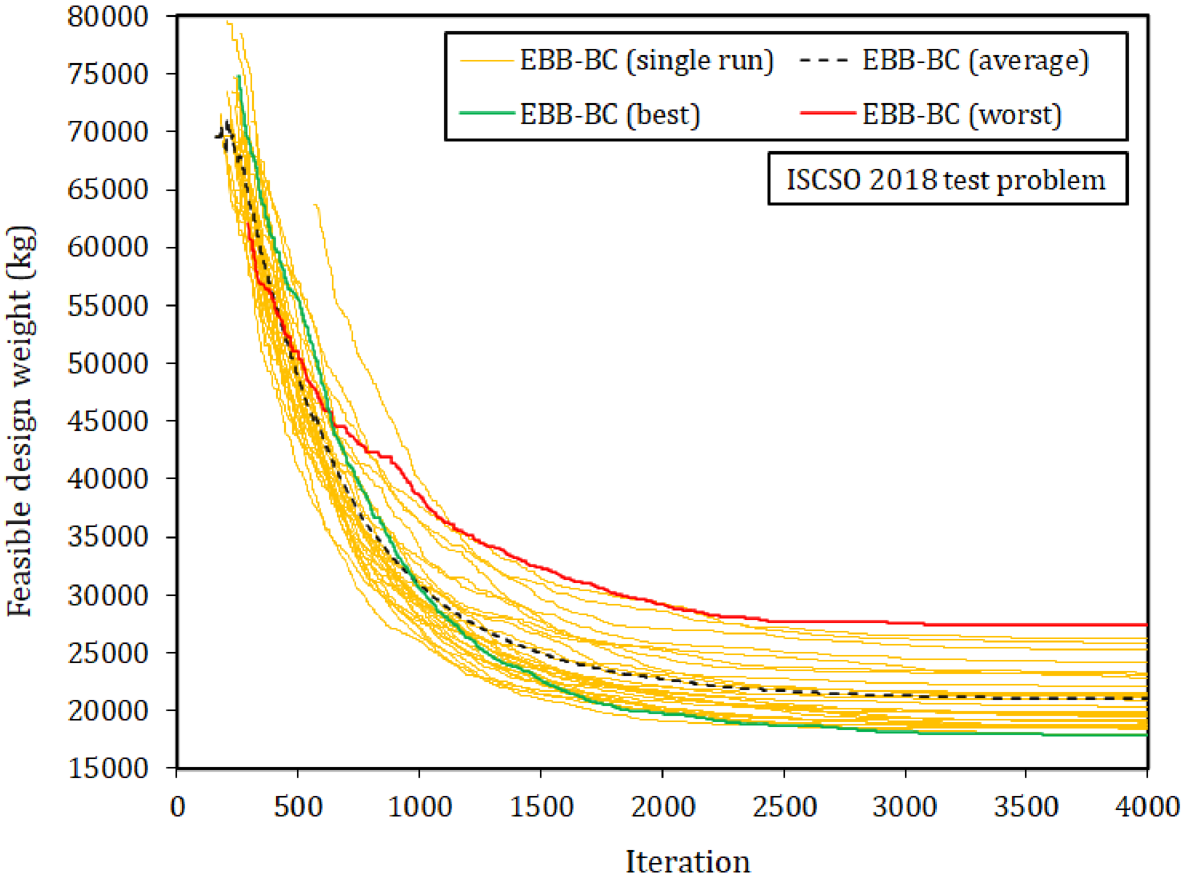


Figure 9: Convergence curves of 30 runs using EBB-BC$_{(50)}$ for ISCSO 2018 test problem

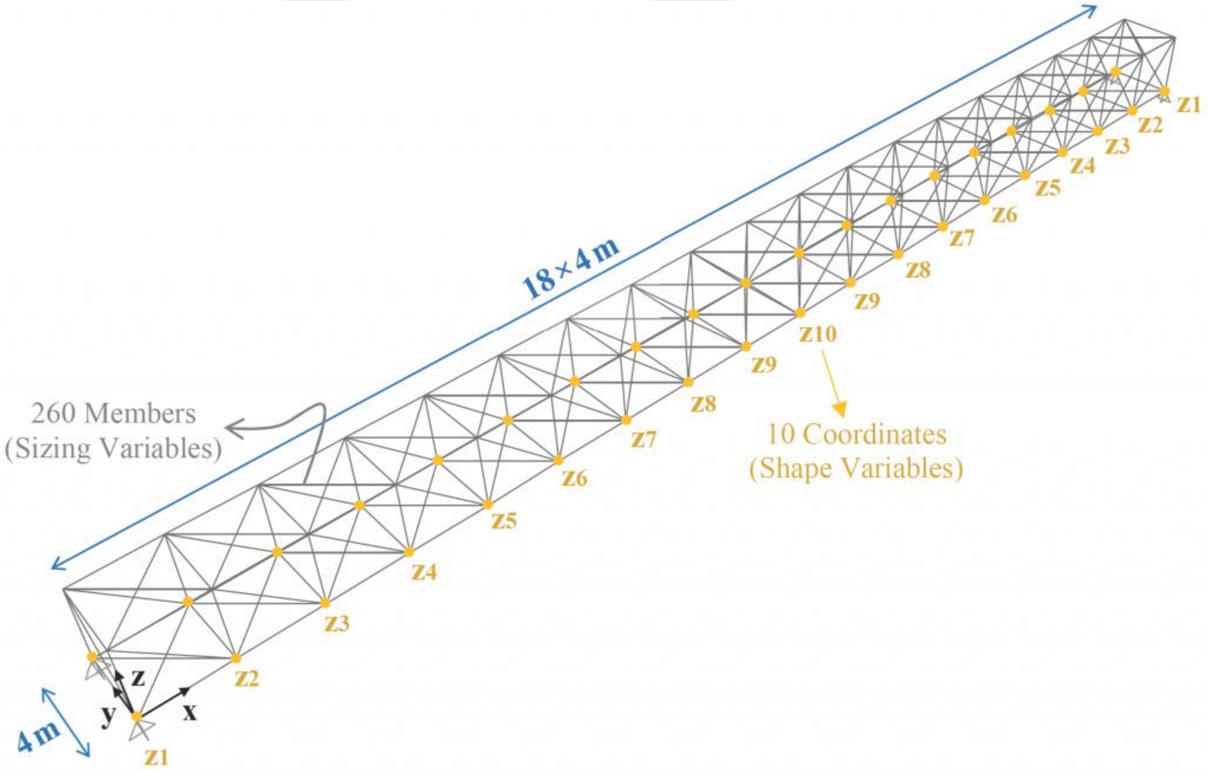


Figure 10: 260-member truss structure (ISCSO 2019)

Figure 10 shows the 260-member truss structure which is chosen from ISCSO 2019. This optimization instance contains 260 sizing variables (which represent the cross-sections of the truss members) and 10 shape variables (which are the z-coordinates of the bottom orange nodes of the structure, including the support nodes, which are grouped symmetrically about the mid-span). Hence, in total, the problem includes 270 distinct design variables. It should be noted that each pair of orange nodes with the same x-coordinate will have the same height. For this design optimization problem, the maximum number of objective function evaluations performed by the algorithms is limited to 200000. The best feasible design weight which is reported by the winner of the ISCSO 2019 for this optimization problem is 12329.1302 kg [23].

Optimum design of the 260-member truss structure is carried out using EBB-BC$_{(25)}$, EBB-BC$_{(50)}$, and EBB-BC$_{(100)}$ and the obtained feasible design weights are presented in Table 4. It is apparent from Table 4 that for the ISCSO 2019 test example EBB-BC$_{(50)}$ algorithm obtains a design weight of 15810.4 kg which is lighter than the design weights achieved by EBB-BC$_{(25)}$, and EBB-BC$_{(100)}$ namely, 17878.0 and 16233.5 kg, respectively. Considering a reference design weight of 12329.1302 kg [23] from the ISCSO 2019 as the best solution known for this test example, it can be seen from the optimization results that EBB-BC$_{(25)}$, EBB-BC$_{(50)}$, and EBB-BC$_{(100)}$ achieve normalized solution qualities of 0.69, 0.78, and 0.76, respectively. The convergence curves of all the independent optimization runs using EBB-BC$_{(50)}$ are plotted for this test example in Figure 11.

GCRIS

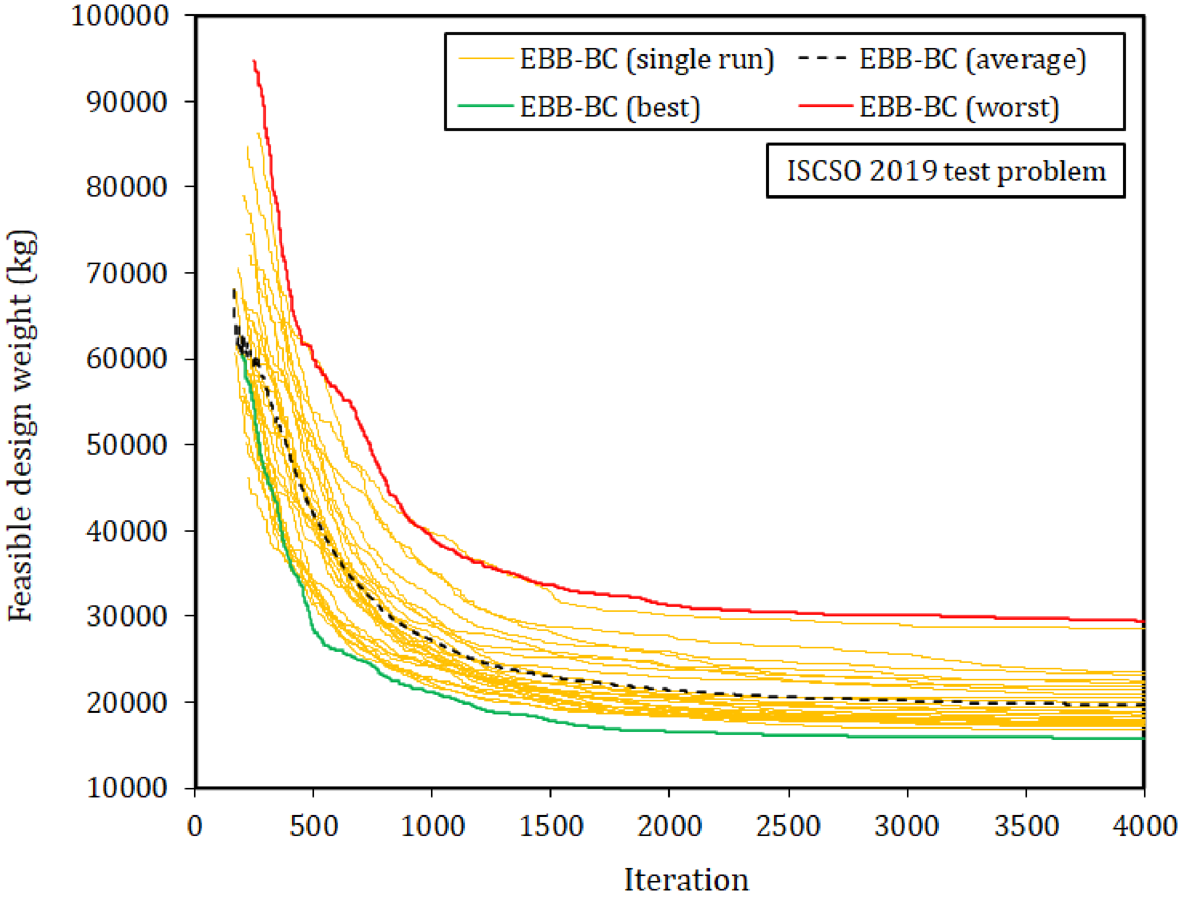


Figure 11: Convergence curves of 30 runs using EBB-BC$_{(50)}$ for ISCSO 2019 test problem

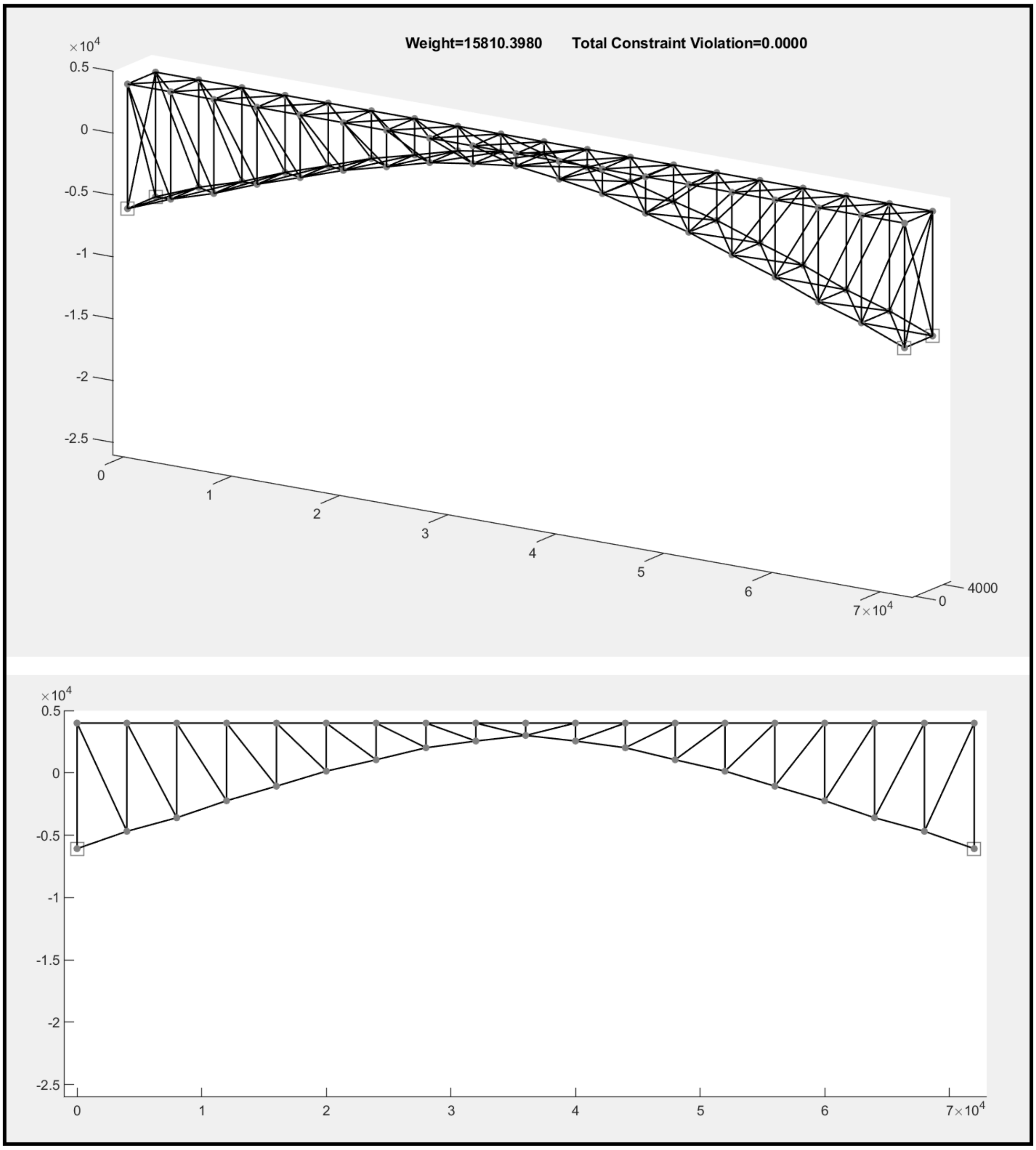


Figure 12: Optimized geometry obtained using EBB-BC(50) for ISCSO 2019 test problem

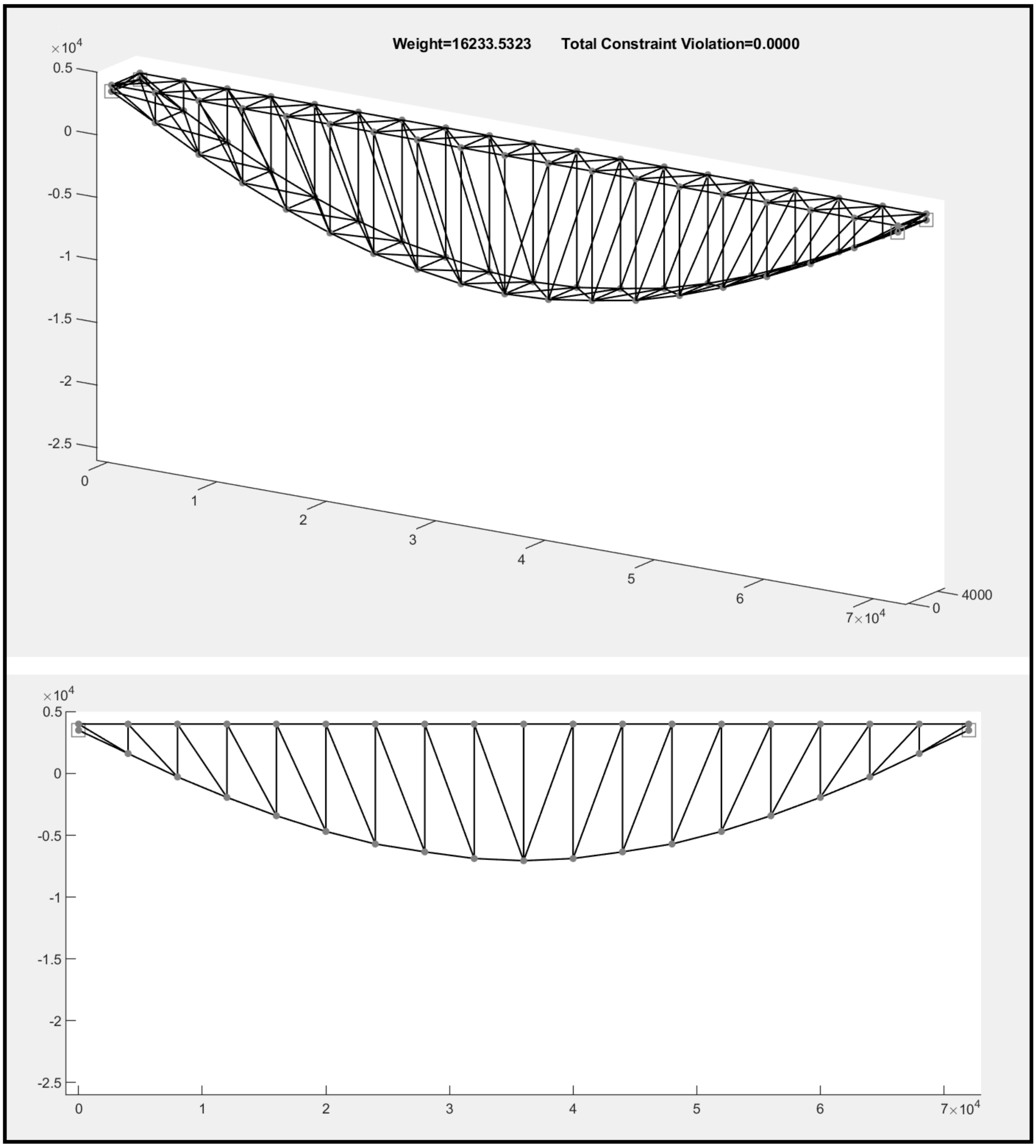
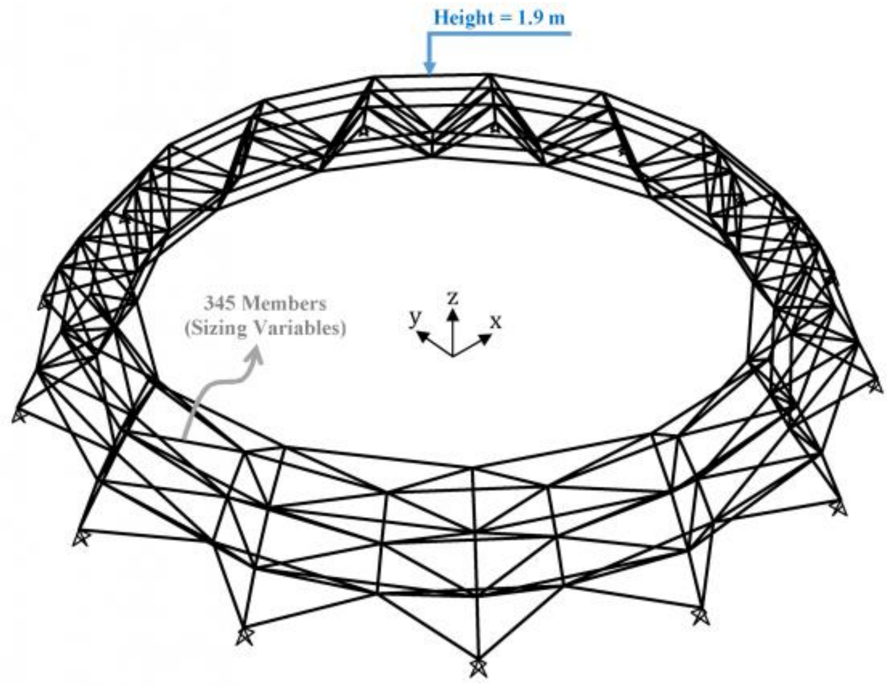


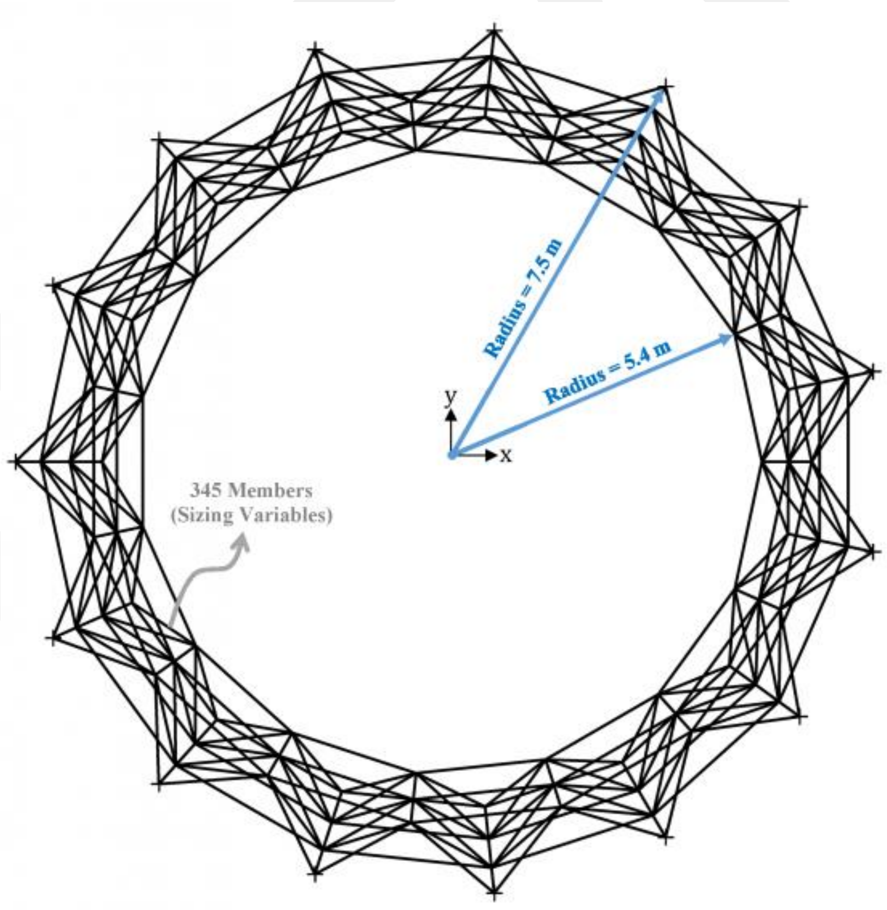
Figure 13: Optimized geometry obtained using EBB-BC$_{(100)}$ for ISCSO 2019 test problem

Figures 12 and 13 show the final optimized geometries obtained using EBB-BC$_{(50)}$ and EBB-BC$_{(100)}$ for the ISCSO 2019 problem instance, respectively. The optimized shapes shown in these figures illustrate the capability of metaheuristics in providing different alternative configurations as candidate final solutions for a specific structural system. Meanwhile, it can be observed from Figures 12 and 13 that the considerably different optimized geometries, achieved using the EBB-BC$_{(50)}$ and EBB-BC$_{(100)}$, correspond to relatively close design weights of 15810.4 and 16233.5 kg, respectively. Accordingly, it can be deduced that this test example involves strong local optima representing candidate solutions with relatively similar design weights yet significantly different configurations. Due to this property of the ISCSO 2019 test example, it could be difficult for optimization algorithms to avoid trapping in such local optima while exploring the search space. This may occur when an evolutionary algorithm converges to a specific geometrical configuration (e.g., Figure 12) at early stages of optimization and cannot alter the evolved geometry to an alternative yet substantially different configuration (e.g., Figure 13) in the succeeding iterations. This is usually due to the decrease in step size of the optimization algorithm for local search in later iterations that makes abrupt changes in the candidate designs less probable. For handling such cases and to avoid premature convergence, it could be a remedy to develop specific operators for evolutionary algorithms to preserve the diversification over iterations.

GCRIS



(a) perspective view



(b) plan view

Figure 14: 345-member truss structure (ISCSO 2021); (a) perspective view, and (b)

The 345-member truss structure shown in Figure 14 is selected from ISCSO 2021. This test problem includes 345 sizing variables which represent the cross-sections of the truss members. For this optimization example, the maximum number of objective function evaluations performed by the optimization algorithms is limited to 200000. The best feasible design weight which is reported by the winner of the ISCSO 2021 for this optimization problem is 3977.2609 kg [23].

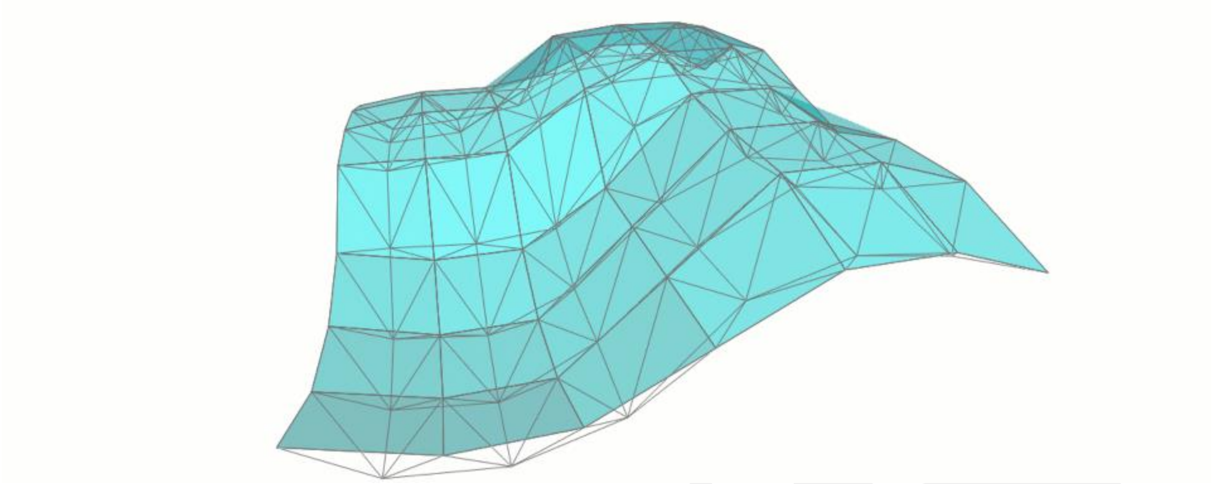
Design optimization of the 345-member truss structure is carried out using EBB-BC$_{(25)}$, EBB-BC$_{(50)}$, and EBB-BC$_{(100)}$ and the obtained solutions through independent optimization runs are given in Table 5. As given in the table, for the ISCSO 2021 test example EBB-BC$_{(100)}$ algorithm obtains a design weight of 4399.0 kg which is slightly better than the solutions achieved by EBB-BC$_{(25)}$, and EBB-BC$_{(50)}$ i.e., 4520.0 and 4415.2 kg, respectively. Considering a reference design weight of 3977.2609 kg [23] from the ISCSO 2021 as the best solution known for this problem, it can be seen from the optimization results that EBB-BC$_{(25)}$, EBB-BC$_{(50)}$, and EBB-BC$_{(100)}$ yield normalized solution qualities ranging between 0.88 and 0.90.

GCRIS

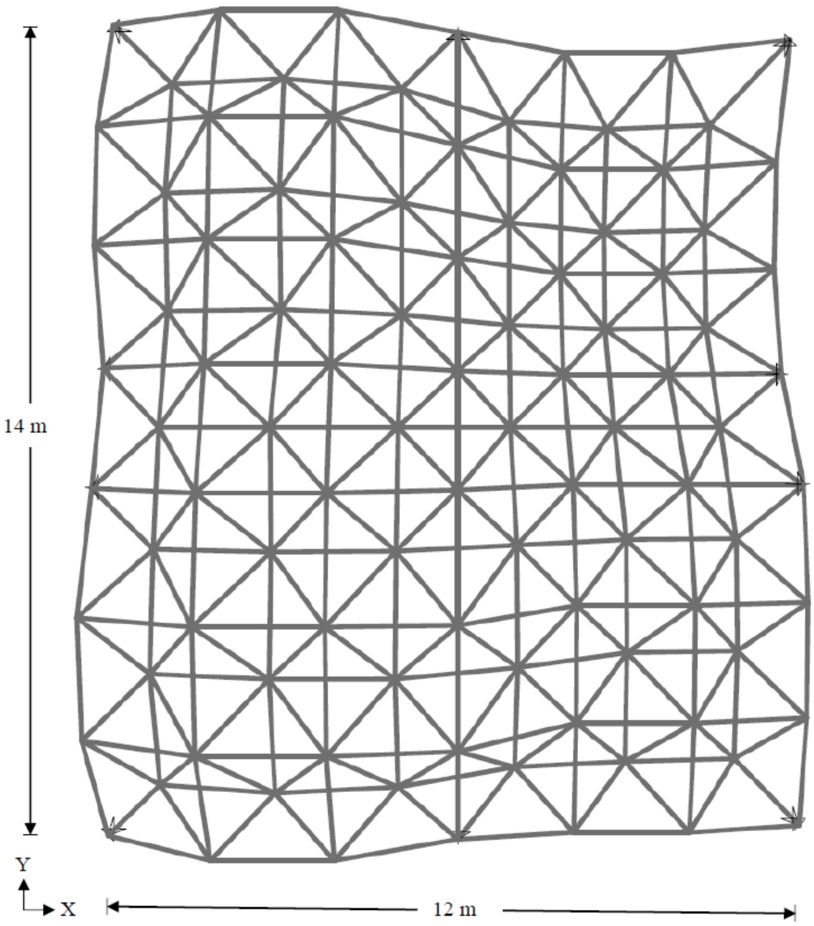
Table 5: Feasible design weights obtained over independent optimization runs (ISCSO 2021-2022)

Run no.	**ISCSO (2021)**			**ISCSO (2022)**		
	EBB-BC$_{(25)}$	EBB-BC$_{(50)}$	EBB-BC$_{(100)}$	EBB-BC$_{(25)}$	EBB-BC$_{(50)}$	EBB-BC$_{(100)}$
1	4772.5	4716.9	4540.1	6748.9	6280.8	6277.7
2	4898.0	4607.6	4537.0	6589.2	6439.7	6255.3
3	4945.0	4536.7	4542.6	6654.6	6381.2	6147.4
4	4773.8	4649.8	4399.0	6547.9	6317.6	6253.8
5	4808.5	4753.6	4483.0	6551.7	6241.4	6307.3
6	4849.2	4544.8	4472.6	6957.6	6216.4	6436.5
7	4755.5	4795.6	4611.9	6493.0	6292.0	6302.2
8	4599.0	4556.4	4542.4	6539.0	6465.4	6268.1
9	4520.0	4690.2	4783.9	6389.4	6311.2	6224.2
10	5051.3	4521.9	4578.9	6756.0	6495.6	6410.5
11	4676.4	4714.2	4593.8	6451.7	6271.5	6352.2
12	4627.6	4503.0	4570.6	6402.6	6409.2	6300.4
13	4603.9	4632.0	4636.0	6600.1	6579.2	6218.8
14	4845.9	4736.4	4595.0	6501.6	6321.5	6301.0
15	4626.1	4545.1	4704.4	6828.3	6317.2	6281.0
16	4832.1	4553.0	4658.8	6917.8	6440.0	6271.5
17	4696.0	4758.0	4420.9	6539.5	6511.2	6173.9
18	5630.8	4587.7	4565.4	6530.9	6238.7	6322.0
19	4871.4	4800.0	4481.4	6376.2	6281.6	6330.0
20	4979.9	4608.9	4434.2	6597.3	6385.3	6411.8
21	4796.5	4656.3	4519.5	6436.8	6685.1	6532.5
22	4721.2	4810.9	4852.6	6331.3	6559.3	6253.4
23	5067.7	4482.0	4641.5	6515.3	6501.1	6425.5
24	4829.6	4415.2	4491.4	6394.4	6403.9	6278.7
25	4910.0	4753.0	4693.2	6597.8	6353.4	6623.3
26	4661.9	4634.2	4615.7	6633.0	6321.8	6469.6
27	4806.1	4457.5	4405.6	6405.0	6372.4	6517.2
28	4890.3	4629.3	4564.7	6590.5	6394.6	6329.7
29	4688.7	4452.0	4502.6	6284.6	6417.8	6366.2
30	4823.1	4519.2	4738.1	6440.4	6318.3	6307.9
Best weight (kg)	4520.0	4415.2	4399.0	6284.6	6216.4	6147.4
Worst weight	5630.8	4810.9	4852.6	6957.6	6685.1	6623.3
Mean weight	4818.6	4620.7	4572.6	6553.4	6384.1	6331.6
Standard deviation	203.0	113.1	109.6	163.1	110.8	107.3
Coefficient of variation (%)	4.2	2.4	2.4	2.5	1.7	1.7
No. analyses	200000	200000	200000	200000	200000	200000
*ISCSO winner solution (kg)	3977.2609	3977.2609	3977.2609	5976.5057	5976.5057	5976.5057
Normalized solution quality	**0.88**	**0.90**	**0.90**	**0.95**	**0.96**	**0.97**

*The design weight reported by the winner of the ISCSO [23].



(a) perspective view



(b) plan view

Figure 15: 336-member free-form truss structure (ISCSO 2022); (a) perspective view, and (b) plan view.

Figure 15 shows the 336-member free-form truss structure adopted from ISCSO 2022. This optimization problem includes 336 sizing variables which represent the cross-sections of the truss members. For this test example, the maximum number of objective function evaluations carried out by the optimization algorithms is limited to 200000. The best feasible design weight which is reported by the winner of the ISCSO 2022 for this optimization problem is 5976.5057 kg [23].

Optimum design of the 336-member free-form truss structure is accomplished using the three configurations of EBB-BC algorithm and the obtained feasible design weights are tabulated in Table 5. As can be seen from the table, for the ISCSO 2022 test example EBB-BC$_{(100)}$ algorithm obtains a design weight of 6147.4 kg which is lighter than those of EBB-BC$_{(25)}$, and EBB-BC$_{(50)}$ namely, 6284.6 and 6216.4 kg, respectively. Regarding a reference design weight of 5976.5057 kg [23] from the ISCSO 2022 as the existing best result for this problem, it can be observed from the obtained results that EBB-BC$_{(25)}$, EBB-BC$_{(50)}$, and EBB-BC$_{(100)}$ achieve promising normalized solution qualities of 0.95, 0.96, and 0.97, respectively.

To summarize the general performance of all the three configurations of the EBB-BC algorithm in the ISCSO 2016-2022 test examples, corresponding normalized solution qualities are visualized for each instance in Figure 16. In general, considering the challenging nature of the test examples, it can be inferred from the results that the employed three optimizers exhibit relatively acceptable performances especially for population sizes of 50 and 100. Thus, with respect to the obtained results, and in line with the previous works [34, 37], a population size of 50 can be suggested as a good starting value for parameter tuning. On the other hand, the numerical experiments reveal that, among the investigated structural optimization examples, the ISCSO 2018 and 2019 test

GCRIS

problems are more challenging instances for the employed three optimizers where a deterioration in the performances can be observed (see Table 4). In contrast, as shown in Figure 16, the most successful performances of the employed configurations of the EBB-BC algorithm have been observed in the ISCSO 2022 test instance. The presented numerical results together with readily replicable challenging ISCSO test problems can be utilized for benchmarking current and new structural optimization algorithms. This could alleviate some of the main problems in benchmarking and comparison methodologies of evolutionary algorithms [39] especially when adopted for structural optimization applications.

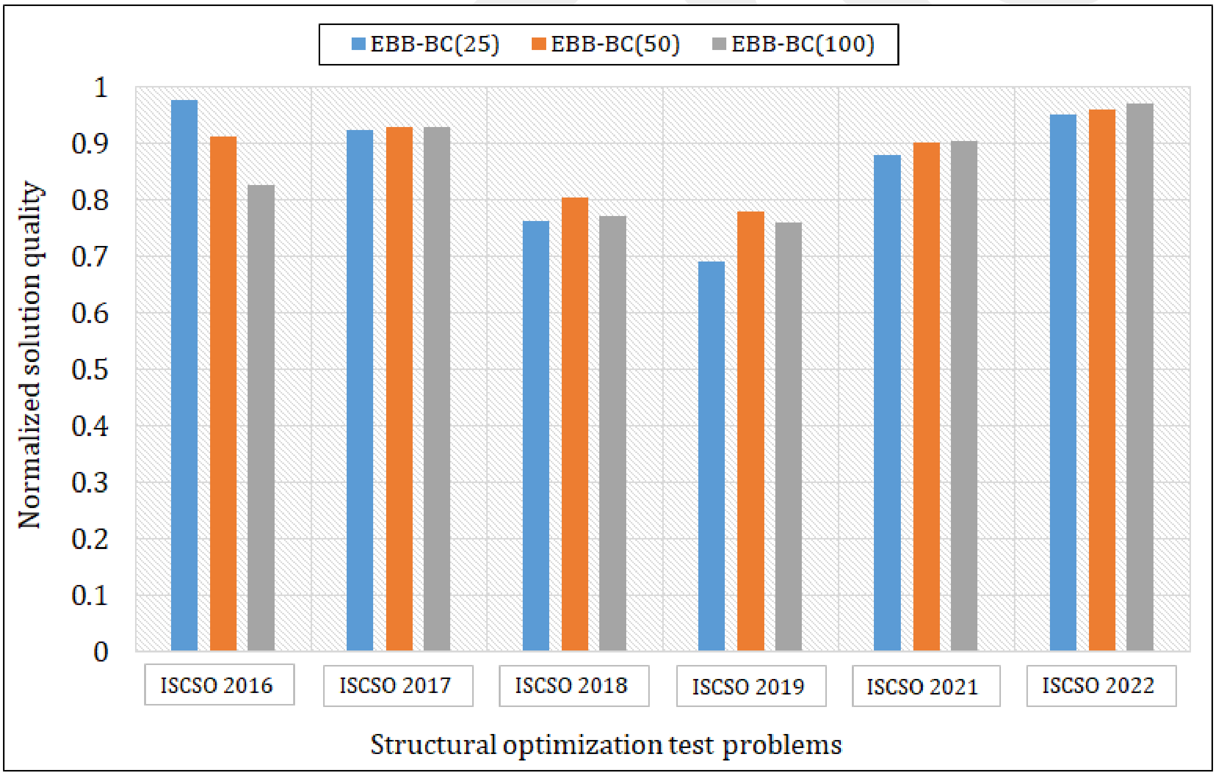


Figure 16: Normalized solution qualities obtained for the ISCSO 2016-2022 test problems

6. Concluding remarks

In the past decades, there has been a drastic increase in the number of metaheuristic or evolutionary optimization techniques developed for handling structural optimization problems. Despite considerable works and efforts devoted to developing sophisticated truss optimization techniques, the literature has not witnessed a concurrent progress in developing new standard benchmark suites for better assessment of the proposed algorithms. Consequently, often evaluation of the algorithms has been performed using a selection of classic test instances which became unchallenging for contemporary, sophisticated techniques. In this work, with respect to the common difficulties encountered in developing and replicating new test suites, it is aimed to provide a new and challenging baseline for the benchmarking of truss optimization algorithms. For this purpose, the most recent six optimization test instances of the ISCSO are compiled to create the "ISCSO 2016-2022 benchmarking suite" – a new, challenging and easily replicable test set for the evaluation of truss optimization algorithms. Furthermore, to provide statistical insight, the proposed test suite is tackled using a contemporary metaheuristic technique namely, EBB-BC algorithm, and the obtained solutions are compared to that reported by the winner of the ISCSO in each year.

The numerical results indicate that while the solutions found by the three different configurations of the optimization algorithm show an acceptable level of consistency with the best solutions known for the problem instances, none of the employed configurations was able to outperform the corresponding best solution. In general, considering the challenging nature of the ISCSO problem instances, it can be inferred from the obtained results that among the employed three optimizers, EBB-BC$_{(50)}$, and EBB-BC$_{(100)}$ exhibit more reliable performances. Therefore, with respect to the numerical experiments, and in line with the previous works, a population size of 50 can be suggested as a good starting

GCRIS

point for parameter tuning.

The provided statistical results together with the easy-to-implement ISCSO structural analysis, design, and visualization functions, accessible from ISCSO website [23] (also provided as supplementary material to this paper), can pave the way for more rigorous assessment of structural optimization algorithms. It is worthwhile to highlight that research in metaheuristic structural optimization could be better promoted through widespread use of standard benchmark suites and performance evaluation rules. This can help change the current counterproductive debates on algorithms, such as those on the source of inspiration of metaheuristics, to more constructive discussions and collaborations for establishing realistic and tangible performance assessment metrics using standard benchmark suites. Consequently, structural optimization algorithms with a better performance on standard benchmark suites, would receive more attention in the literature, regardless of the name of the algorithm or its source of inspiration. Indeed, domain-specific optimization competitions will play an important role in the development of standard benchmark suites in the future as they did in the past.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Supplementary data to this article can be found online at:

https://doi.org/10.1016/j.istruc.2023.105409

GCRIS

References

[1] Lagaros, N.D. (2018) The environmental and economic impact of structural optimization, Struct Multidisc Optim, 58, 1751–1768.

[2] Stolpe, M. (2016) Truss optimization with discrete design variables: a critical review, Struct Multidisc Optim 53: 349–374.

[3] Kicinger, R., Arciszewski, T., De Jong K. (2005) Evolutionary computation and structural design: A survey of the state-of-the-art, Comput Struct, 83 (23–24): 1943-1978.

[4] Lamberti, L., Pappalettere, C. (2011)"Metaheuristic Design Optimization of Skeletal Structures: A Review", Computational Technology Reviews, vol. 4, pp. 1-32. doi:10.4203/ctr.4.1

[5] Kashani, A.R., Camp, C.V., Rostamian, M., Azizi, K., Gandomi, A.H. (2022) Population-based optimization in structural engineering: a review, Artif Intell Rev 55: 345–452.

[6] Lee, K.S., Geem, Z.W. (2004) A new structural optimization method based on the harmony search algorithm, Computers & Structures, 82(9–10): 781-798.

[7] Camp, CV. (2007) Design of space trusses using Big Bang-Big Crunch optimization, J. Struct. Eng., ASCE, 133: 999–1008.

[8] Perez, R.E., Behdinan, K. (2007) Particle swarm approach for structural design optimization, Computers & Structures, 85(19–20): 1579-1588.

[9] Li, L.J., Huang, Z.B., Liu, F. (2009) A heuristic particle swarm optimization method for truss structures with discrete variables, Comput Struct, 87: 435–443.

[10] Lamberti, L., Pappalettere, C. (2011) "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

GCRIS

Computing Technology in Civil, Structural and Environmental Engineering", Civil-Comp Press, Stirlingshire, UK, Paper 11, doi:10.4203/ccp.97.11

[11] Kazemzadeh Azad, S., Hasançebi, O. (2014) An elitist self-adaptive step-size search for structural design optimization, Applied Soft Computing, 19: 226-235.

[12] Jafari, M., Salajegheh, E., Salajegheh, J. (2021) Optimal design of truss structures using a hybrid method based on particle swarm optimizer and cultural algorithm, Structures, 32: 391-405.

[13] Tang, H., Lee, J. (2023) Chaotic enhanced teaching-based differential evolution algorithm applied to discrete truss optimization, Structures, 49: 730-747.

[14] Cohn, M.Z. (1993). Theory and Practice of Structural Optimization. In: Rozvany, G.I.N. (eds) Optimization of Large Structural Systems. NATO ASI Series, vol 231. Springer, Dordrecht.

[15] Cohn, M.Z. and Dinovitzer, A.S. (1994) Application of structural optimization. ASCE Journal of Structural Engineering, 120(2): 617–650.

[16] Alimoradi, A., Foley, C.M., Pezeshk, S. (2010) Benchmark Problems in Structural Design and Performance Optimization: Past, Present, and Future-Part I., Structures Congress, ASCE, 455–466.

[17] Molina, D., LaTorre, A., Herrera, F. (2018) An Insight into Bio-inspired and Evolutionary Algorithms for Global Optimization: Review, Analysis, and Lessons Learnt over a Decade of Competitions, Cognitive Computation, 10: 517–544.

[18] Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A., Tiwari, S. (2005) Problem definitions and evaluation criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Tech. Report, Nanyang Technological University, Kangal Report Number 2005005.

[19] García, S., Molina, D., Lozano, M., Herrera, F. (2009) A study on the use of non-

GCRIS

parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization, Journal of Heuristics, 15, 617–644.

[20] Omidvar, M.N., , Li, X., Tang, K., (2015) Designing benchmark problems for large-scale continuous optimization, Information Sciences, 316: 419-436.

[21] Maučec, M.S., Brest, J. (2019) A review of the recent use of differential evolution for large-scale global optimization: An analysis of selected algorithms on the CEC 2013 LSGO benchmark suite, Swarm and Evolutionary Computation, 50, 100428.

[22] Hasançebi, O., Çarbas, S., Dogan, E., Erdal, F., Saka, M.P. (2009) Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures, Comput. Struct., 87: 284–302.

[23] International Student Competition in Structural Optimization (ISCSO) https://www.brightoptimizer.com/

[24] Kazemzadeh Azad, S., Kazemzadeh Azad, S., Hasançebi, O.,(2016) Structural optimization problems of the ISCSO 2011-2015: A Test set, International Journal of Optimization in Civil Engineering, 6 (4): 629-638.

[25] Kale, I.R., Kulkarni, A.J. (2021) Cohort intelligence with self-adaptive penalty function approach hybridized with colliding bodies optimization algorithm for discrete and mixed variable constrained problems, Complex Intell. Syst. 7, 1565–1596.

[26] Albert, B.A., Zhang, A.Q. (2022) SpartaPlex: A deterministic algorithm with linear scalability for massively parallel global optimization of very large-scale problems, Advances in Engineering Software, 166, 103090.

[27] Ghosh, A., Deb, K., Averill, R., Goodman, E. (2021) Combining User Knowledge and Online Innovization for Faster Solution to Multi-objective Design Optimization

GCRIS

Problems. In: , et al. Evolutionary Multi-Criterion Optimization. EMO 2021. Lecture Notes in Computer Science, vol 12654. Springer, Cham.

[28] Etaati, B., Dehkordi, A.A., Sadollah, A., El-Abd, M., Neshat, M. (2022) A Comparative State-of-the-Art Constrained Metaheuristics Framework for Truss Optimisation on Shape and Sizing, Mathematical Problems in Engineering, vol. 2022, Article ID 6078986, 13 pages.

[29] Dehkordi, A.A., Etaati, B., Neshat, M., Mirjalili, S. (2023) "Adaptive Chaotic Marine Predators Hill Climbing Algorithm for Large-Scale Design Optimizations," in IEEE Access, 11: 39269–39294.

[30] Kaveh, A., Biabani Hamedani, K. (2022) Improved arithmetic optimization algorithm and its application to discrete structural optimization, Structures, 35: 748–764.

[31] Abualigah ,L., Diabat, A., Mirjalili, S., Abd Elaziz, M., Gandomi, A.H. (2021) The arithmetic optimization algorithm, Comput Methods Appl Mech Eng, 376: 113609.

[32] Lagaros, N.D., Plevris, V., Kallioras, N.A. (2022) The Mosaic of Metaheuristic Algorithms in Structural Optimization, Archives of Computational Methods in Engineering, 29: 5457–5492.

[33] MATLAB, 2019, version 9.7 (R2019b) The MathWorks Inc., Natick, Massachusetts.

[34] Hasançebi, O., Kazemzadeh Azad, S. (2012) An exponential big bang-big crunch algorithm for discrete design optimization of steel frames, Comput. Struct, 110–111: 167–179.

[35] American Institute of Steel Construction (AISC). Manual of Steel Construction, Load & Resistance Factor Design. 2nd ed. Chicago, 1994.

[36] Erol, O.K., Eksin, I. (2006) A new optimization method: big bang-big crunch, Adv. Eng. Software, 37: 106–111.

GCRIS

[37] Kazemzadeh Azad, S., Kazemzadeh Azad, S., Hasançebi, O., (2016) Structural optimization using big bang-big crunch algorithm: A review, Int. J. Optim. Civ. Eng., 6: 433–445.

[38] Kazemzadeh Azad, S. (2018) Seeding the initial population with feasible solutions in metaheuristic optimization of steel trusses, Eng. Optim. 50: 89–105.

[39] Del Ser, J., Osaba, E., Molina, D., Yang, X.-S., Salcedo-Sanz, S., Camacho, D., Das, S., Suganthan, P.N., Coello Coello, C.A., Herrera, F. (2019) Bio-inspired computation: Where we stand and what's next, Swarm and Evolutionary Computation, 48: 220–250.

GCRIS