Applied Computing and Intelligence

https://www.aimspress.com/journal/aci

Applied Computing and Intelligence
3 (2): 116–144
DOI: 10.3934/aci.2023007
Received: 23 May 2023
Revised: 21 September 2023
Accepted: 26 September 2023
Published: 07 October 2023

Research article

The Mountain Gazelle Optimizer for truss structures optimization

Nima Khodadadi[1,*], El-Sayed M. El-Kenawy[2], Francisco De Caso[1], Amal H. Alharbi[3], Doaa Sami Khafaga[3] and Antonio Nanni[1]

[1] Department of Civil and Architectural Engineering, University of Miami, Coral Gables, FL, USA; nima.khodadadi@miami.edu, fdecaso@miami.edu, nanni@miami.edu

[2] Department of Communications and Electronics, Delta Higher Institute of Engineering and Technology, Mansoura 35111, Egypt; skenawy@ieee.org

[3] Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia; ahalharbi@pnu.edu.sa, dskhafga@pnu.edu.sa

* **Correspondence:** Email: nima.khodadadi@miami.edu; Tel: +1-786-822-9664.

Academic Editor: Jidong Yang

Abstract: Computational tools have been used in structural engineering design for numerous objectives, typically focusing on optimizing a design process. We first provide a detailed literature review for optimizing truss structures with metaheuristic algorithms. Then, we evaluate an effective solution for designing truss structures used in structural engineering through a method called the mountain gazelle optimizer, which is a nature-inspired meta-heuristic algorithm derived from the social behavior of wild mountain gazelles. We use benchmark problems for truss optimization and a penalty method for handling constraints. The performance of the proposed optimization algorithm will be evaluated by solving complex and challenging problems, which are common in structural engineering design. The problems include a high number of locally optimal solutions and a non-convex search space function, as these are considered suitable to evaluate the capabilities of optimization algorithms. This work is the first of its kind, as it examines the performance of the mountain gazelle optimizer applied to the structural engineering design field while assessing its ability to handle such design problems effectively. The results are compared to other optimization algorithms, showing that the mountain gazelle optimizer can provide optimal and efficient design solutions with the lowest possible weight.

Keywords: mountain gazelle optimizer (MGO) algorithms; optimal design; truss structures

1. Introduction

Optimization involves determining the best values for a system's design parameters to minimize or maximize the fitness function while satisfying all constraints. Challenges in optimization are encountered across various industries, academic fields and research areas. There are different strategies for optimization, including exact algorithms and heuristic and metaheuristic algorithms. The former category, exact algorithms, requires fewer complex computations, making it quicker to execute but potentially less useful and practical. Conversely, the second category, metaheuristic algorithms, displays random or stochastic characteristics and makes an informed search decision in some intelligent areas [1].

A meta-heuristic algorithm is a high-level strategy used to guide the search for a solution to an optimization problem. These algorithms are often used when the complexity of the problem cannot be processed by traditional computational optimization methods, for which a known solution does not exist. There are many examples of meta-heuristic algorithms that are used for structure problems, such as genetic algorithms (GAs) [2], an artificial bee colony (ABC) [3], the chaotic coyote algorithm [4], ant colony optimization (ACO) [5], the artificial gorilla troops optimizer (AGTO) [6], big bang–big crunch (BB-BC) [7] and the stochastic paint optimizer (SPO) [8].

The meta-heuristic algorithms vary in the different types, such as hybrid, multi-objective and improved versions. Some of the existing hybrid versions include the invasive weed optimization shuffled with frog-leaping algorithms (SFLA-IWO) [9], the particle swarm optimization with genetic algorithms (PSO-GA) [10], the optimality criterion with genetic algorithms (OC-GA) [11], the cuckoo search with stochastic paint optimizers (CSSPO) [12] and moth-flame optimization with simulated annealing (MFO-SA) [13].

On the other hand, some of the well-known multi-objective versions, which can solve problems with multiple aims or goals, include thermal exchange optimization (MOTEO) [14], the stochastic paint optimizer (MOSPO) [15], chaos game optimization (MOCGO) [16], teaching-learning based optimization (MOTLBO) [17], moth-flame optimization (MMFO) [18], the chimp optimizer (MOCO) [19], the arithmetic optimization algorithm (MAOA) [20], atomic orbital search (MOAOS) [21], the search group algorithm (MOSGA) [22] and the artificial hummingbird algorithm (MOAHA) [23].

In addition, enhanced versions of meta-heuristic algorithms include the chaotic stochastic paint optimizer (CSPO) [24], improved chicken swarm optimization (ICSO) [25] and the advanced neural network algorithm (ANNA) [26]. These algorithms have also been developed for use in structural engineering design and other fields [27–30]. These algorithms improve a candidate solution iteratively through a series of random or probabilistic moves guided by some form of heuristic or rule of thumb.

The overarching goal of the algorithms is to converge on a near-optimal or optimal solution within a reasonable amount of time. Two significant components of any meta-heuristic algorithm are exploration and exploitation. Exploration refers to generating diverse solutions to explore the search space on a global scale. Furthermore, exploitation means focusing on the search in a local region by exploiting the information that a current good solution is found in this region. Meta-heuristic algorithms must maintain an adequate balance between the exploration and exploitation tendencies to be competitive in terms of robustness and performance [31].

While the simplicity and flexibility of meta-heuristic algorithms make algorithms most appealing, it is impossible to guarantee that in any given problem, a globally optimal solution can be found in any given problem. This is because most algorithms are presented as stochastic optimizations, meaning the final solution depends heavily on various kinds of generated random variable fields [32].

Structural optimization has ushered in advanced methodologies, transcending the confines of traditional analysis, which often relied solely on a designer's experience and intuition. The evolution of digital computing capabilities has significantly empowered and enabled the application of contemporary optimization algorithms. Consequently, designers can now ascertain optimal solutions in a time-efficient manner, solutions that might have remained elusive when only conventional methods were employed [33].

Trusses in structural engineering are a structural form of members subjected to pure tension or compression. Members are connected by means of pin joints to create rigid structures. The joints are subjected to external forces. Trusses are deemed efficient because they ensure a consistent stress level across the entire cross-section of each member, depending on the loading conditions.

Deterministic and random methods are the two major categories of optimization problem-solving techniques. Optimization issues that are linear, continuous, differentiable and convex are well-handled by deterministic methods. The drawback of these methods is that they are unable to solve problems that are nonlinear, nonconvex, nondifferentiable, high dimensional, NP-hard (non-deterministic polynomial-time hardness) and involve discrete search spaces.

These elements are among the characteristics of real-world optimization problems and have caused deterministic approaches to fail. Stochastic algorithms—particularly metaheuristic algorithms—have been developed to meet this challenge. By utilizing random search in the problem-solving space and relying on random operators, metaheuristic algorithms can offer suitable solutions to optimization problems.

It is important to note that there is no assurance that a solution found using metaheuristic algorithms will be the best or globally optimal. Researchers have created numerous metaheuristic algorithms because of this fact and produce better solutions. The No Free Lunch (NFL) theorem [34] states that an algorithm can only be expected to offer optimal solutions for certain problems while it may deliver average results for others. This is why the ongoing search for more algorithms for truss optimization is necessary.

Many researchers have found that metaheuristic algorithms are useful for designing and analyzing truss structures with numerous members [35]. However, the efficiency of optimization algorithms is problem-dependent, meaning that not all algorithms perform well in every scenario. Hence, it is crucial to evaluate and contrast the efficiency of the recently developed algorithm mountain gazelle optimizer (MGO) [36], which has not been utilized in structural optimization, by implementing it across various practical engineering problems.

The motivation for this work is that mountain gazelles have demonstrated encouraging results in a different optimization area, making them a valuable candidate for consideration in the optimization of truss structures.

Recently, Abdollahzadeh et al. [36] introduced MGO, a nature-inspired meta-heuristic algorithm inspired by wild mountain gazelles' social behavior and hierarchical structure. The algorithm utilizes mathematical formulations to emulate gazelle populations' hierarchical and social dynamics. This approach is intended to improve the search for optimal solutions to optimization problems. To this end, the application of MGO for structural optimization is proposed herein as a technique applied to optimize truss structures.

In summary, this paper makes several key contributions:

• It compares four different metaheuristic algorithms on three design optimization problems.
• It uses three space truss structures to evaluate the algorithms mentioned.
• It includes comparing different methods using statistical results and convergence curves.
• It found that the proposed MGO algorithm outperformed the other algorithms mentioned.

The next section of this paper provides a detailed literature review of the optimization of truss structures with metaheuristic algorithms. Then, the application of MGO is presented and defined, followed by the problem definition, and results from the analysis of three truss structures are reported. Lastly, the conclusions are provided, introducing optimum design methodologies for designing truss structures using MGO compared to other optimization algorithms. The comparison is made with the arithmetic optimization algorithm (AOA) [37], material generation algorithm (MGA) [38] and crystal structure algorithm (CRY) [39].

2. Truss structures related works

In recent years, metaheuristic algorithms have been utilized as an optimization technique for the optimum design of various structural systems. Optimal design of structures refers to finding the best design plan under an array of different constraints; such constraints may include the lightest weight, lowest cost, project construction time and/or maximum rigidity. Typically, the structure's behavior and the project's cost-effectiveness are critical, and the approach presented herein considers both in the optimization. Additionally, it ensures the structure's safety, reduces construction costs and minimizes the use of natural resources and materials.

Degertekin [40] suggested two new harmony search (HS) algorithms, the efficient harmony search algorithm (EHS) and the self-adapting harmony search algorithm (SAHS), for the sizing optimization of truss structures. It is known that the HS algorithm is extremely sensitive to the tuning parameters even though the efficiency of the HS algorithm has been demonstrated in numerous engineering optimization applications. Four classic truss construction weight minimization issues were described to show the suggested algorithm efficiency. The outcomes of the current algorithms were contrasted with those of the HS algorithm and other meta-heuristic algorithms.

Degertekin and Hayalioglu [41] optimized truss structures using a novel meta-heuristic search technique called teaching-learning based optimization (TLBO). The approach utilized the similarity between how students learn and how solutions to optimization issues are found. The teacher phase and learner phase are the two phases of the TLBO. The four design examples showed the method's applicability. Results for the design examples showed that while the TLBO occasionally produced slightly heavier designs than other meta-heuristic methods, for the most part, it achieved results that were at least as good compared to other meta-heuristic optimization methods in terms of convergence capability and optimum solutions.

For the best truss structure design, a novel fusion of swarm intelligence and chaos theory was provided by Kaveh et al. [42]. The technique is known as the chaotic swarming of particles (CSP) and was inspired by the tendency to form swarms found in various animals and chaos theory. The CSP algorithm was used to optimize truss structures, and the outcomes were compared to those of the previous meta-heuristic algorithms to demonstrate the effectiveness of the new approach.

The weight of truss constructions was reduced using the created flower pollination algorithm

(FPA) by Bekdas et al. [43] which also took into account design variables for sizing. Because flowering plants self-pollinate and cross-pollinate, the new algorithm may effectively combine local and global searches. It also employed an iterative constraint management strategy where trial designs were approved or disapproved depending on the permitted level of a constraint violation, which progressively decreased as the search process got closer to the optimum. More important, this approach searched to produce optimum designs that were always realizable. Three traditional 2D and 3D truss structure sizing optimization tasks were used to test the novel approach. According to optimization findings, the suggested method was competitive with other cutting-edge metaheuristic algorithms.

Recently, Jawad et al. [44] studied the combined optimization of truss structures using the artificial bee colony algorithm (ABC), a swarm intelligence-based optimization technique. The goal was to maximize truss structure architecture and member size under displacement, stress and buckling constraints. The foundation of the ABC is the simulation of honeybees' clever foraging activity. The design factors for shape and size optimization for the truss structure system were the cross-sectional areas of the members and the nodal coordinates of the joints. The problem constraints were regarded as the allowable stress, the Euler buckling stress and the displacement. Four benchmark structural optimization tasks put the ABC to the test. The outcomes showed that ABC is better than other algorithms regarding optimal weight, standard deviation and quantity of structural analyses. With a 100% success rate, ABC has had a strong performance.

Structural optimization has also been completed using the recent political optimizer (PO) method proposed by Awad [45]. The algorithm, modeled after the multi-phased political process in parliamentary democracies, effectively balances exploration and exploitation by logically dividing the search agent population into political parties competing for dominance of constituencies. While preserving algorithmic speed and efficiency, this competitive-based population partitioning technique ensured that sufficient search space was adequately examined for the global optima. Three planar trusses—10 members, 18 members and 200 members—and four space trusses—22 members, 25 members, 72 members and 942 members—with various loading circumstances and design restrictions had been taken into consideration in order to evaluate the algorithm's effectiveness statistically. The results showed that the PO algorithm outperforms all previously proposed state-of-the-art optimization methodologies for small to medium-sized structural systems in every aspect, including final optimized weight, algorithmic stability and convergence speeds.

The search group algorithm (SGA), a metaheuristic optimization technique, was presented by Gonçalves et al. [46] to address truss structure optimization specifically. A selection of benchmark issues from the literature was used to illustrate the usefulness of the SGA. Due to truss complexity, problems were given special consideration, including topology optimization, discrete design variables and/or natural frequency limitations. The key finding of these numerical tests was that the SGA was capable of producing the lightest structures yet discovered for five of the six samples examined. In most instances, it also enhanced the statistics of the algorithm's independent runs. These findings highlight the SGA's strengths in this area and promote its continued improvement and use in solving applied engineering challenges.

Most recently, in order to optimize space trusses with continuous design variables, Goodarzimehr et al. [47] seeks to offer a new hybrid technique that combines particle swarm optimization and genetic algorithm (PSOGA), resulting in an effective hybridization algorithm for solving optimization issues. These algorithms have demonstrated exceptional performance when it comes to resolving optimization issues involving continuous variables. The PSO essentially simulates how individual birds communicate with one another to provide information about their position, velocity and fitness. The outcomes show that both exploration and exploitation were enhanced using the hybrid PSO algorithm. Hence, optimizing the weight of structural issues under stress and displacement constraints applies the PSO method.

A dynamic variant of the water strider algorithm (DWSA) has been suggested by Kaveh et al. [48]. The dynamic water strider algorithm (DWSA) and WSA were both used to reduce the weight of various skeletal components. The water strider insect's territorial behavior, intelligent ripple communication, mating style, feeding mechanics and succession are all mimicked by the nature-inspired metaheuristic known as WSA. These algorithms' effectiveness was examined by optimizing various truss and frame constructions that were subject to various loading circumstances and limitations. When DWSA findings were compared to those of other approaches, it was clear that DWSA is an effective methodology for maximizing structural design and reducing structural weight while meeting all limitations.

The most effective hybrid optimization approach for truss design has been provided by Kaveh et al. [49]. Based on the invasive weed optimization algorithm and the shuffled frog-leaping algorithm (ISO-SFLA), the suggested technique uses the shuffled frog-leaping algorithm to find optimal solution regions quickly. The invasive weed optimization takes advantage of global solutions. The novel hybrid method was used to improve several benchmark truss structures. In comparison to specific other approaches, this algorithm converged to better or at least equivalent solutions while using less structural analysis. The results were compared to those acquired previously utilizing other recently established metaheuristic optimization techniques.

An enhanced chicken swarm optimization (CSO) technique for truss structure optimization was presented by Li et al. [25] in order to increase the effectiveness of the structural optimization design in truss computation. The idea of mixing chaotic strategy with reverse learning strategy was introduced in the initialization of the fundamental CSO algorithm to guarantee the capability of a global search. In order to more effectively mix global and local search, the inertia weighting element and the learning component were added to the chicken position updating mechanism. The differential evolution method was then used to maximize the algorithm's total individual position. This study offered a novel technique for optimizing truss structures.

3. Problem definition

Three truss structure optimization designs are solved using the MGO algorithm to evaluate the performance of the MGO algorithm compared to alternative metaheuristic algorithms. The alternative algorithms include the arithmetic optimization algorithm (AOA) [37], material generation algorithm (MGA) [38] and crystal structure algorithm (CRY) [39].

Population size and the maximum number of function evaluations were consistent with other algorithms for a fair comparison. Using a termination criterion based on the maximum number of function evaluations ensures a fair comparison. Each problem is tackled separately, with 30 runs conducted for each one. The parameters for each algorithm are modified based on a review of the existing literature. The algorithm was developed using MATLAB 2023a, and the trusses were solved using the direct stiffness method with SAP2000 v14.1 and API. The computer used for testing had an Apple M2 Max and 96 GB of RAM on a Macintosh Ventura operating system. Three specific truss structure case studies are conducted: spatial structure with 25 members, spatial structure with 72 members and dome structure with 120 members. Results are compared to other methods.

The mathematical formulas for the size optimization of truss structures solved herein are presented in Eq. (1). The optimization aims to find the optimal cross-section (A_i) values that minimize the structural weight of the W while satisfying constraints on the design variable sizes and structural responses. The problem is formulated as follows:

$$\textit{Minimize:} \quad W(\{x\}) = \sum_{i=1}^{nm} \gamma_i . A_i . L_i(x) \tag{1}$$

$$\begin{aligned} subject\ to: \quad & \delta_{min} \leq \delta_i \leq \delta_{max} & i = 1,2,\ldots,m \\ & \sigma_{min} \leq \sigma_i \leq \sigma_{max} & i = 1,2,\ldots,n \\ & \sigma_i^b \leq \sigma_i^i \leq 0 & i = 1,2,\ldots,ns \\ & A_{min} \leq A_i \leq A_{max} & i = 1,2,\ldots,ng \end{aligned}$$

The optimization problem is formulated as follows: $W(\{x\})$ is the weight of the structure, n is the number of members in the structure, m is the number of nodes, ns is the number of compression elements and ng is the number of design variables or member groups. Here, γ_i is the material density of member i, L_i is the length of member i, $and\ W(\{x\})$ is the cross-sectional area of member i chosen between $A_{\min}$ and $A_{\max}$, where min and max indicate the lower and upper bounds, respectively. σ_i and δ_i are the stress and nodal deflection, respectively. The penalty function is as follows:

$$f_{penalty}(X) = (1 + \varepsilon_1 . \nu)^{\varepsilon_2} \qquad \nu = \sum_{i=1}^{n} \max[0, \nu_i] \tag{2}$$

where ν is the total amount of constraints violated, and constants ε_1 and ε_2 are chosen based on the exploration and exploitation rate of the search space. In this case, $\varepsilon_1$is set to 1, and ε_2 is chosen to minimize penalties and reduce cross-sections. At the start of the search process, ε_2 is set to 1.5 and is then increased to 3.

4. The mountain gazelle optimizer

The algorithm, known as the mountain gazelle optimizer (MGO) [36], uses a mathematical model based on mountain gazelles' social behaviors and life patterns. It optimizes using four key aspects of gazelle life: bachelor male herds, maternity herds, solitary, territorial males and migration to find food. The MGO algorithm assigns each gazelle (X_i) to a herd of maternity herds, bachelor male herds or solitary, territorial males during the optimization process. A new gazelle can be born from any of these herds.

The best global solution in the MGO is represented by adult male gazelles in their herd territories. One-third of the search population is estimated to have the lowest cost, as the gazelles in the male bachelor herds are young and not yet mature enough to reproduce or lead the female gazelles. Gazelles represent other potential solutions in maternity herds, where strong gazelles with good solutions engage in both exploitation and exploration. This means that a solution can both move towards the best solution and perform exploration, as determined by the four mechanisms in the MGO model.

In addition, other solutions offered to the general population are compared to gazelles in a herd. The strong gazelles with effective solutions are kept at the end of each cycle. Other solutions introduced to the general population with a low cost are considered weak and eliminated from the entire population. The MGO's mathematical formulae for optimization are described as follows.

4.1. Territorial solitary males (TSM)

When male mountain gazelles mature and attain strength, they establish individual territories and display territorial behavior, with substantial spaces between territories. The conflict between adult male gazelles occurs over territory control or mating rights to the females. Young males attempt to take over the territory or female, while adult males strive to defend their territory. Eqs. (3) to (5) have been utilized to simulate the adult male's territory.

$$TSM = male_{gazelle} - \left|\left(ri_1 \times BH - ri_2 \times X(t)\right) \times F\right| \times Cof_r \quad (3)$$

$$BH = X_{ra} \times \lfloor r_1 \rfloor + M_{pr} \times \lceil r_2 \rceil, \qquad ra = \left\{\left\lceil \frac{N}{3} \right\rceil \dots N\right\} \quad (4)$$

$$F = N_1(D) \times exp\left(2 - Iter \times \left(\frac{2}{MaxIter}\right)\right) \quad (5)$$

In Eq. (3), the $male_{gazelle}$ represents the position vector of the optimal global solution (adult male). The variables ri_1 and ri_2 are random integers, either 1 or 2. BH is the coefficient vector for the group of young males, calculated with Eq. (4). F is calculated using Eq. (5). Cof_r is a randomly chosen coefficient vector that is updated in each iteration to enhance search capability, computed using Eq. (6).

In Eq. (4), X_{ra} refers to a random solution (young male) within the range of ra. M_{pr} represents the average number of randomly selected search agents. N stands for the total number of gazelles, and r_1 and r_2 are random values between 0 and 1. In Eq. (5), N_1 is a random number derived from a standard distribution in the dimensions of the problem. The exponential function is denoted as exp. $MaxIter$ signifies the total number of iterations, and $Iter$ represents the current iteration number.

$$Cof_i = \begin{cases} (a+1) + r_3, \\ a \times N_2(D), \\ r_4(D), \\ N_3(D) \times N_4(D)^2 \times \cos\left((r_4 \times 2) \times N_3(D)\right), \end{cases} \quad (6)$$

$$a = -1 + Iter \times \left(\frac{-1}{MaxIter}\right) \quad (7)$$

In Eq. (6), a is calculated using Eq. (7), while r_3 and r_4, are random numbers within the range of 0 and 1. N_2, N_3 and N_4 are random numbers in the standard range and dimensions of the problem. In the problem dimensions, r_4 is also a random number between 0 and 1. Finally, cos represents the Cosine function.

4.2. Maternity herds (MH)

Maternity herds are crucial in the life cycle of mountain gazelles, as they give birth to strong male gazelles. Male gazelles can also be involved in the delivery of gazelles and young males trying to claim females, which is modeled using Eq. (8).

$$MH = \left(BH + Cof_{1,r}\right) + \left(ri_3 \times male_{gazelle} - ri_4 \times X_{rand}\right) \times Cof_{1,r} \quad (8)$$

In Eq. (8), BH represents the impact factor vector of young males, calculated using Eq. (4). $male_{gazelle}$ stands for the best global solution (adult male) in the current iteration. Finally, X_{rand} is the vector position of a randomly selected gazelle from the entire population.

4.3. Bachelor male herds (BMH)

As male gazelles grow stronger, they establish their own territory and compete to control female gazelles. Young males fight against adult males for the territory and possession of the females, sometimes resulting in violent confrontations.

$$BMH = (X(t) - D) + \left(ri_5 \times male_{gazelle} - ri_6 \times BH\right) \times Cof_r \quad (9)$$

$$D = \left(|X(t)| + |male_{gazelle}|\right) \times (2 \times r_6 - 1) \quad (10)$$

This behavior is represented mathematically in Eq. (9) using variables such as $X(t)$ (the position of the gazelle in the current iteration), D (calculated using Eq. (10)), ri_5 and ri_6 (random integers 1 or 2), BH (the impact factor of the young male herd, calculated using Eq. (4)), $male_{gazelle}$ (the position of the male gazelle, the best solution) and r_6 (a random number between 0 and 1). Eq. (10) uses $X(t)$ and $male_{gazelle}$ (the positions of the gazelle vectors in the current iteration and the best solution, respectively).

4.4. Migration to Search for Food (MSF)

Mountain gazelles roam to find food and migrate due to their high speed and jumping ability. The behavior is modeled in Eq. (11) as follows:

$$MSF = (ub - lb) \times r_7 + lb \quad (11)$$

where ub and lb represent the upper and lower limits of the problem, and r_7 is a random integer between 0 and 1.

The TSM, MH, BMH and MSF mechanisms are applied to all gazelles to generate new generations. After each era, the gazelles are ranked, and the best ones, with high-quality and low-cost solutions, are kept while the old and weak ones are removed from the population. The best gazelle is the adult male who owns the territory. The MGO general flowchart and pseudo-code are depicted in Figures 1 and 2. MGO can make optimization more accessible to engineers and designers who may not have specialized knowledge in optimization techniques by being easier to use or implement than existing methods.

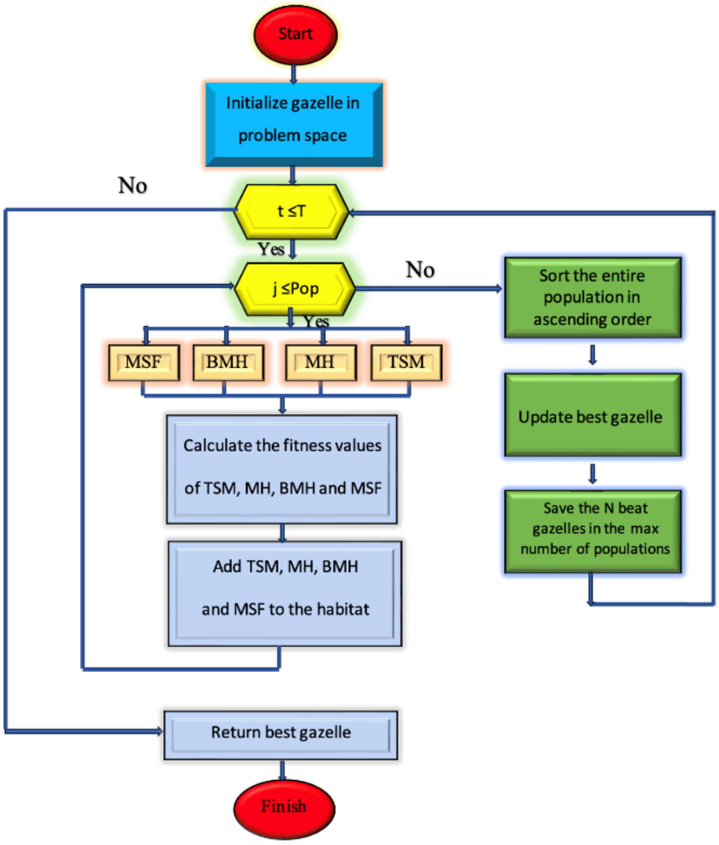

Figure 1. MGO flowchart.

Algorithm
Pseudo-code of MGO

Inputs: The population size N and the maximum number of iterations T

Outputs: Gazelle's location and fitness potential

Create a random population using $X_i(i = 1,1,\ldots,N)$

Calculate Gazelle's fitness levels

While the stopping condition is not met

For each Gazelle (X_i)

Calculate TSM using Eq. (3)

Calculate MH using Eq. (8)

Calculate BMH using Eq. (9)

Calculate MSF using Eq. (11)

Calculate the fitness values of TSM, MH, BMH and MSF, then add them to the habitat

End For

Sort the entire population in ascending order

Update $best_{Gazelle}$

Save the $best_{Gazelle}$ in the maximum number of populations

End While

Return $X_{BestGazelle}$

Figure 2. Pseudo-Code of MGO [36].

5. Truss structures

For each truss structure described below, the MGO solution results are compared with the existing metaheuristic methods available in the literature. To this end, the statistical approaches required several independent optimization runs to be conducted in this work, where a total of 30 runs were performed. The truss structural details were complementary to these optimization runs.

5.1. Spatial structure with 25 members

The first truss structure in this work is a spatial truss structure with 25-member elements. Structural members are made of steel with yield stresses of $\pm 40\ ksi$ and a modulus of elasticity of $104\ ksi$. The density of steel material is $0.1\ lb/in^3$. Table 1 presents a classification of the members by number, indicating each member's beginning and end node. Figure 3 illustrates a schematic view of the structure with dimensions, as well as truss member and node numbering. Each node is constrained by a maximum displacement limit of ± 0.35 inches in each direction, where Table 2 shows the corresponding axial load restrictions for each group. The cross-sectional areas for the members ranged from 0.01 to 3.4 square inches. Table 3 shows the allowable stress for each element.

The convergence history of the MGO for the best run and the average runs against the other algorithms (AOA, MGA and CRY) is provided in Figure 4. The figures show the superior capability of MGO for solving this truss design optimization, clearly outperforming the other methods as the number of iterations quickly reduces.

Table 1. Element group number.

G1	G2	G3	G4	G5	G6	G7	G8
1:(	2:(1,4)	6:(2,4)	10:(6,3)	12:(3,4)	14:(3,10)	18:(4,7)	22:(10,6)
	3:(2,3)	7:(2,5)	11:(5,4)	13:(6,5)	15:(6,7)	19:(3,8)	23:(3,7)
	4:(1,5)	8:(1,3)			16:(4,9)	20:(5,10)	24:(4,8)
	5:(2,6)	9:(1,6)			17:(5,8)	21:(6,9)	25:(5,9)

Table 2. Load Condition for 25-member truss.

Node. No	$Load\ Case\ 1\ (Kips)$			$Load\ Case\ 2\ (Kips)$		
	P_x	P_y	P_z	P_x	P_y	P_z
1	0	20	-5	1	10	-5
2	0	-20	-5	0	10	-5
3	0	0	0	0.5	0	0
6	0	0	0	0.5	0	0

Table 3. Compressive and tensile limitations of 25-member truss.

Element Group	Compressive stress Limitations Ksi (MPa)	Tensile stress Limitations Ksi (MPa)
1: A_1	35.092 (241.96)	40.0 (275.80)
2: A_2 - A_5	11.590 (79.0913)	40.0 (275.80)
3: A_6 - A_9	17.305 (119.31)	40.0 (275.80)
4: A_{10} - A_{11}	35.092 (241.96)	40.0 (275.80)
5: A_{12} - A_{13}	35.092 (241.96)	40.0 (275.80)
6: A_{14} - A_{17}	6.759 (46.603)	40.0 (275.80)
7: A_{18} - A_{21}	6.959 (47.982)	40.0 (275.80)
8: A_{22} - A_{25}	11.082 (76.410)	40.0 (275.80)

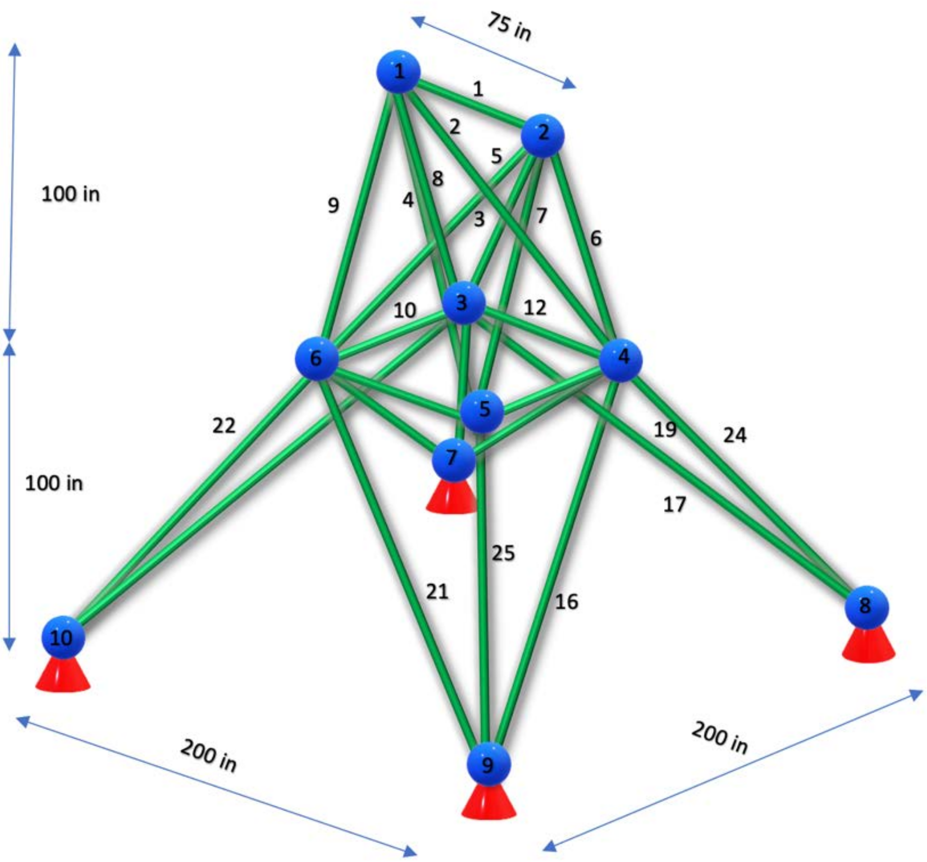

Figure 3. Schematic view of spatial structure with 25 members.

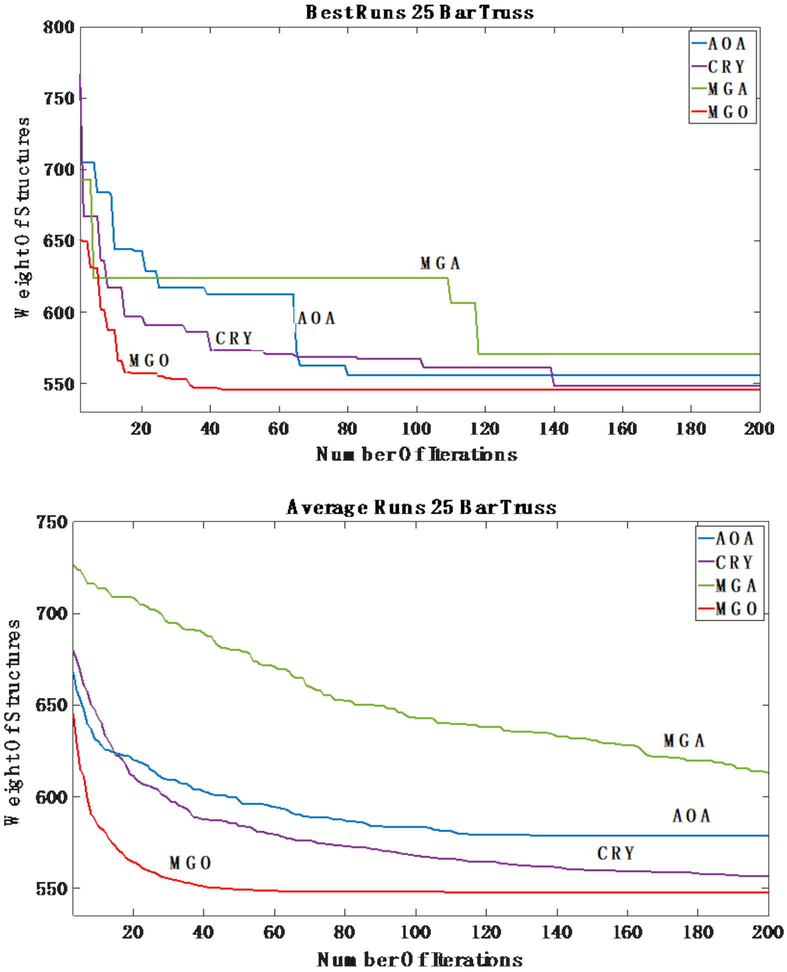

Figure 4. Best run and average run convergence curve's spatial structure with 25 members with different methods

Table 4. The MGO results of spatial structure with 25 members.

Group Member	AOA	CRY	MGA	MGO
1 (A_1)	0.210	0.010	1.441	0.010
2 (A_2 - A_5)	1.915	1.895	2.165	1.879
3 (A_6 - A_9)	3.400	3.324	2.613	3.106
4 (A_{10} - A_{11})	0.010	0.010	0.167	0.010
5 (A_{12} - A_{13})	0.324	0.010	0.298	0.010
6 (A_{14} - A_{17})	0.597	0.606	0.860	0.653
7 (A_{18} - A_{21})	1.576	1.630	1.731	1.728
8 (A_{22} - A_{25})	2.661	2.690	2.636	2.661
Optimum weight (lb.)	555.88	548.37	571.10	545.72
Average weight (lb.)	578.68	556.72	612.96	548.01
Standard deviation	17.46	5.51	21.00	2.12
Number of analyses	1000	7000	1000	2300

Table 4 Summarizes the optimum outcome of the several optimization runs using MGO and other alternative methods to solve the spatial structure with 25 members problem. The minimum acceptable weight for the truss is determined to be the MGA's target weight. To understand the MGA's strength in predicting the optimal weight of the structure, several alternative metaheuristics are obtained to have a better perspective on the capabilities of the MGO.

As reported in Table 4, we can see that the MGA is able to predict a weight of $545.72\ lb$. In this design example, the maximum number of function evaluations was limited to 10,000, and MGO obtained the best results with just 2300 function evaluation numbers. According to the statistical results, close similarities are observed between the mean of multiple runs by MGO and the best run. Also, 30 independent runs are shown in Figure 5. The MGO resulted in the lightest estimate for the mean weight of the spatial structure with 25 members ($548.01\ lb$), followed by AOA. In the case of the 25-member truss structure, the MGO algorithm provides the lowest standard deviation (SD), equal to 2.12, whereas the CRY method can deliver only an SD of 5.51. Figure 6 presents the design constraints, including stresses in each truss member and displacements that occurred in each member, for the best optimization run performed by the MGO. This figure also puts the capacity of the MGO method in constraint handling technique into context.

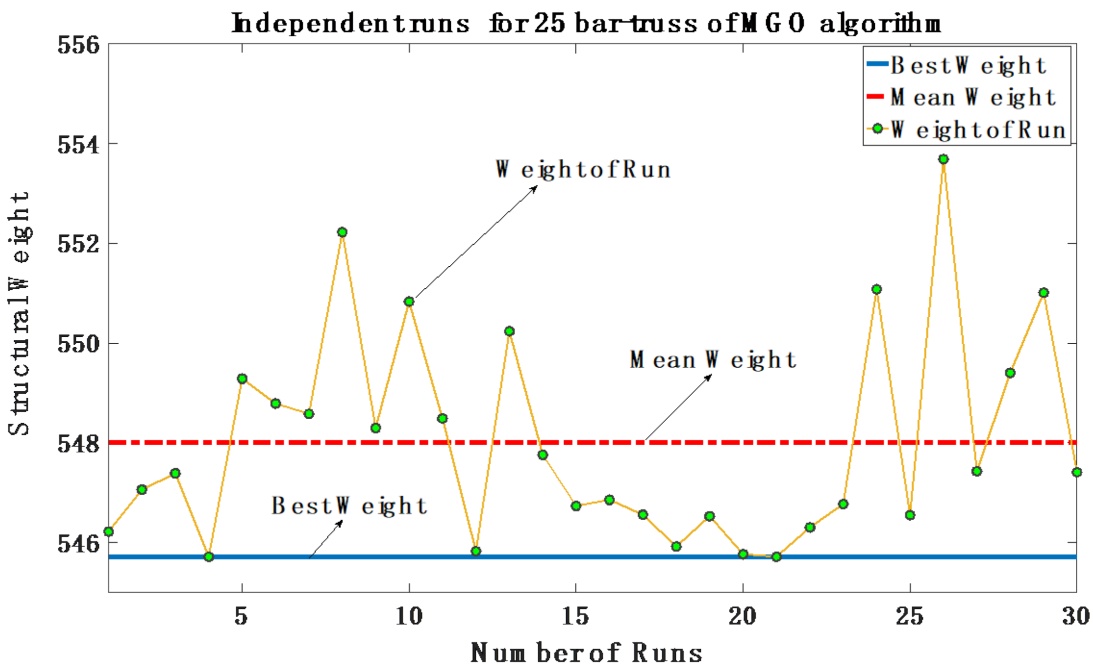

Figure 5. The 30 Independent runs for spatial structure with 25 members with MGO.

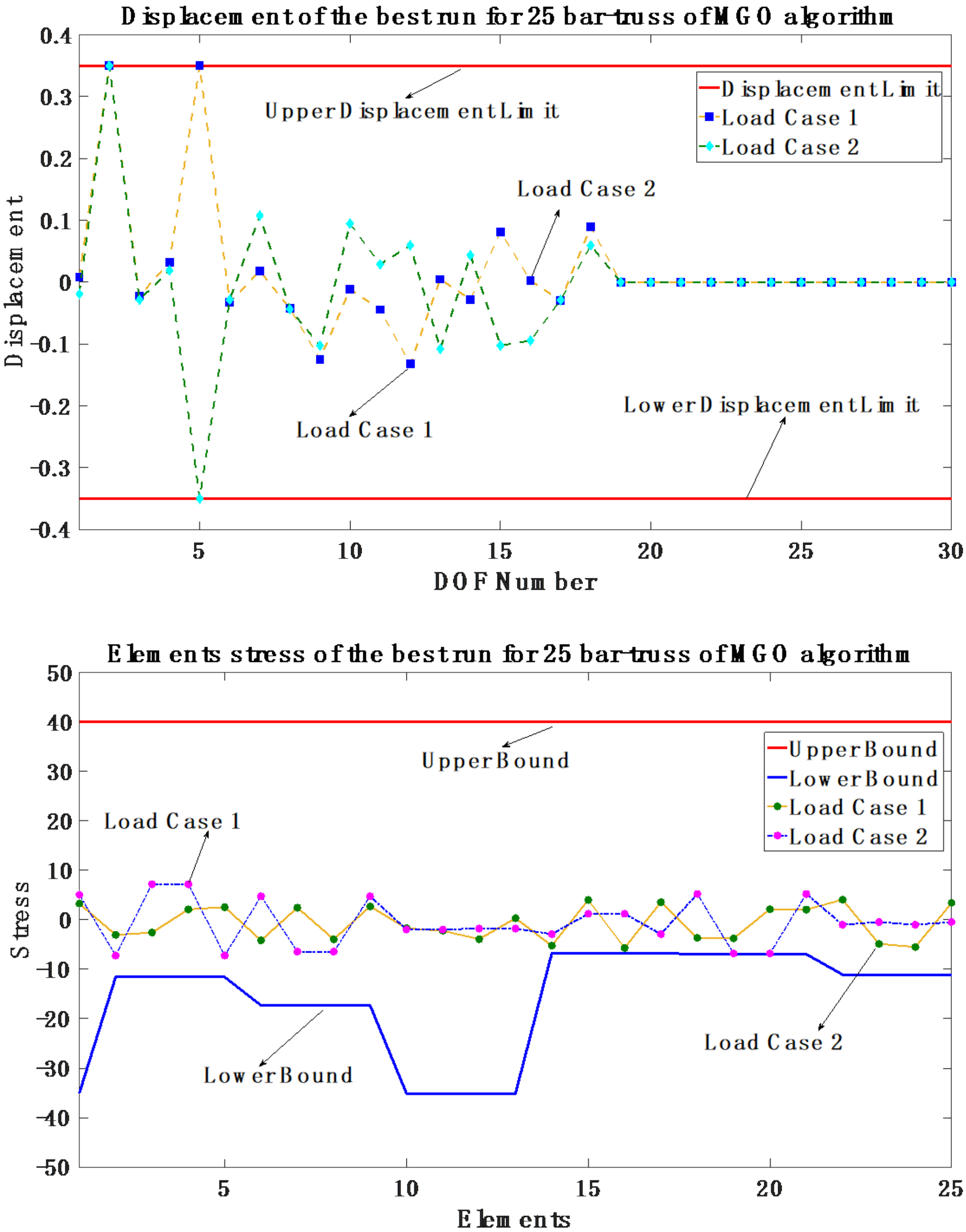

Figure 6. Stress and Displacement ratio for spatial structure with 25 members with MGO.

5.2. Spatial structure with 72 members

This design's truss structure comprises 72 truss members pinned to the ground at four nodes. Structural members are made of steel with yield stresses of $\pm 25\ ksi$ and a modulus of elasticity of $10^4\ ksi$. The density of steel material is 0.1 lb/in^3. Each member's minimum and maximum permissible cross-section area is $0.10\ in^2$ and $4.00\ in^2$. Figure 7 presents a schematic view of the structure with truss members, node numbering and overall dimensions. The convergence history of the MGO for the best run and average runs against all other methods is shown in Figure 8, indicating that the MGO method ranks first based on the number of iterations completed to reach an optimum solution.

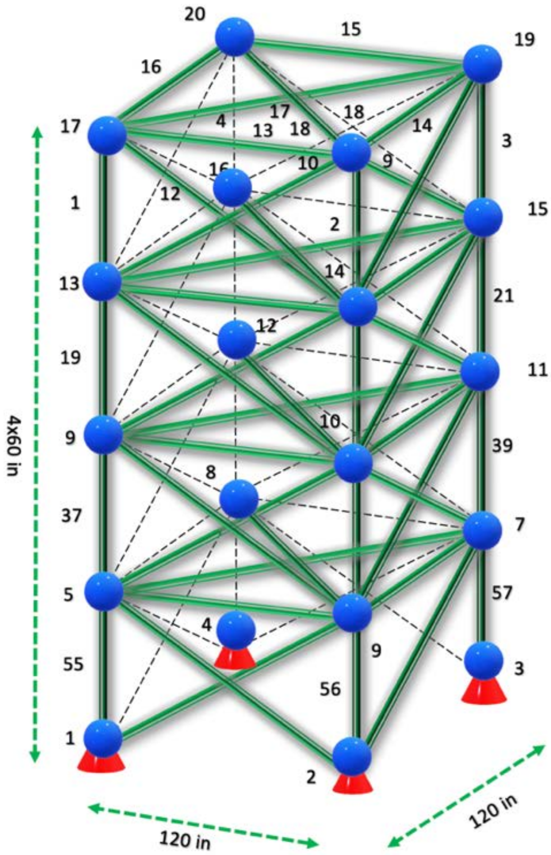

Figure 7. Schematic view of spatial structure with 72 members.

Table 5 presents the best solution achieved via 30 optimization runs performed by the MGO, in addition to the statistical findings. The MGO's results are compared to the other optimization techniques, and it is determined that 379.95 lb is the minimum weight this example can support. In addition, MGO has more reasonable statistical findings, with a mean of $382.54\,lb$ and SD of $2.04\,lb$. However, while taking into account the standard deviation, the MGO algorithm provided the lowest SD, the CRY method achieved the second lowest SD value of equal to 33.74, while the MGA algorithm arrived at the highest SD value of 85.17 for the 72-member truss structure.

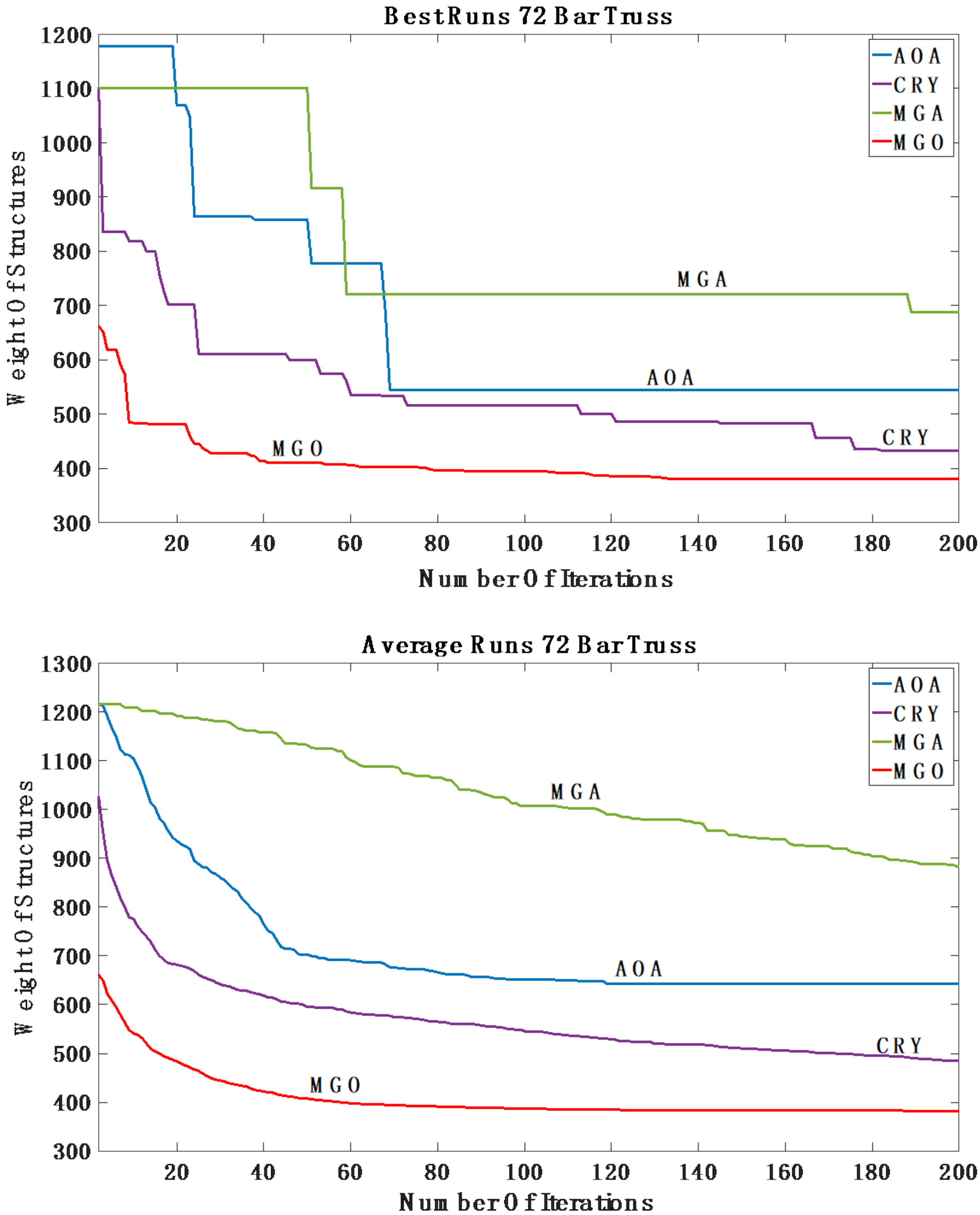

Figure 8. Best run and average run convergence curve spatial structure with 72 members with different methods.

Table 5. The MGO results of spatial structure with 72 members.

Group Member	AO	CRY	MGA	MGO
1 (A_1 - A_4)	2.1	2.221	0.931	1.824
2 (A_5 - A_{12})	0.7	0.404	0.434	0.515
3 (A_{13} - A_{16})	0.1	0.100	0.507	0.100
4 (A_{17} - A_{18})	0.1	0.100	0.100	0.100
5 (A_{19} - A_{22})	1.5	1.410	1.745	1.245
6 (A_{23} - A_{30})	0.4	0.447	0.714	0.536
7 (A_{31} - A_{34})	0.1	0.105	0.783	0.100
8 (A_{35} - A_{36})	1.0	0.114	0.882	0.100
9 (A_{37} - A_{40})	0.7	0.332	1.579	0.554
10 (A_{41} - A_{48})	0.3	0.754	0.629	0.520
11 (A_{49} - A_{52})	1.3	0.100	0.683	0.100
12 (A_{53} - A_{54})	0.1	0.175	0.100	0.100
13 (A_{55} - A_{58})	0.1	0.471	2.398	0.156
14 (A_{59} - A_{66})	1.1	0.534	0.903	0.538
15 (A_{67} - A_{70})	0.5	0.765	0.104	0.380
16 (A_{71} - A_{72})	0.1	0.953	0.521	0.592
Optimum weight (lb.)	545	432.97	687.77	379.95
Average weight (lb.)	643	485.69	881.59	382.54
Standard deviation	64.	33.74	85.17	2.04
Number of analyses	100	9150	10000	6800

Figure 9 presents the design constraints, including stresses in each truss member and displacements that occurred in each member, for the best optimization run performed by the MGO. This figure also puts the capacity of the MGO method in constraint handling technique into context. The 30 independent runs for MGO are shown in Figure 10. According to this figure, the best eight and mean weights of all runs are depicted, as well as all other obtained weights.

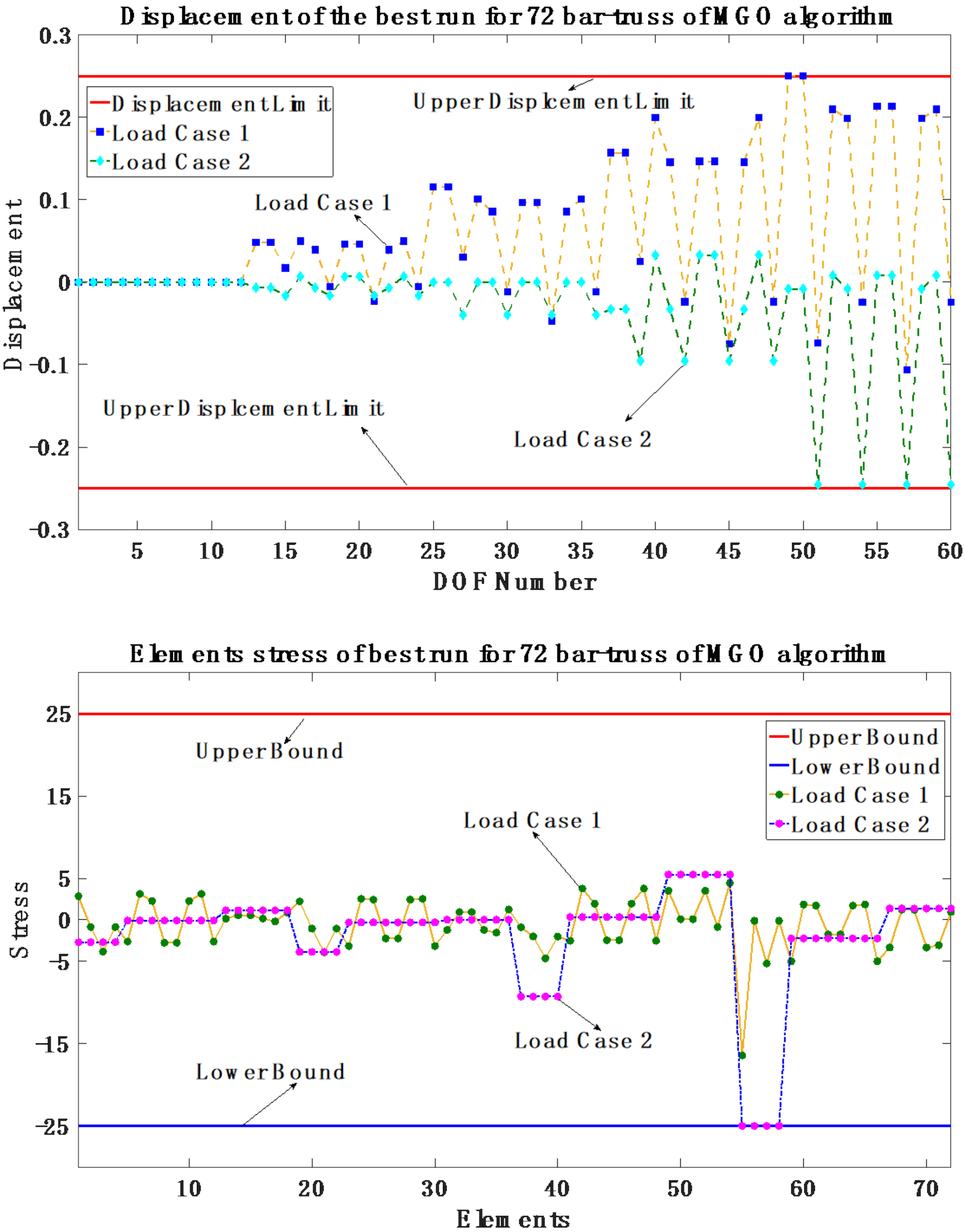

Figure 9. Stress and Displacement ratio for spatial structure with 72 members with MGO.

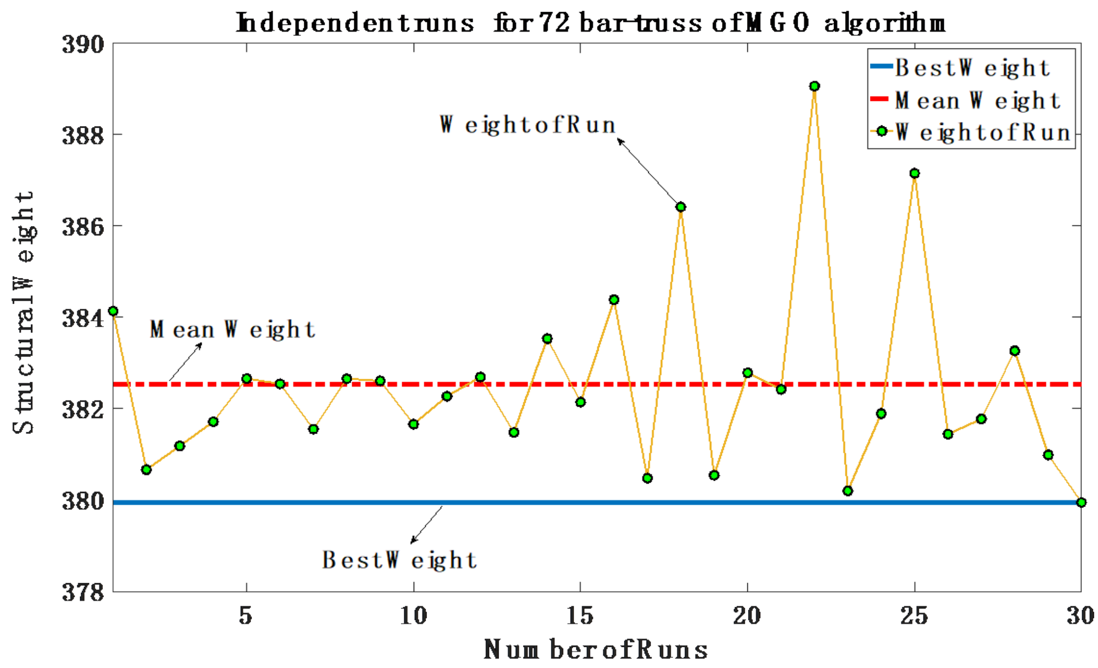

Figure 10. The 30 Independent runs for spatial structure with 72 members with MGO.

5.3. Dome structure with 120 members

The third truss design addresses minimizing the dome structure's weight with 120 members, as shown in Figure 11. Soh and Yang [50] studied this design case as a configuration optimization problem. The tensile and compressive stresses were established based on the AISC [51] code. The modulus of elasticity is $30450\, ksi\ (210000\, MPa)$, the density of the material is $0.288\, lb/in^3\ (7971.810\ \ lb/in^3)$ and the yield stress of steel used is $58.0\, ksi\ (400\, MPa)$. The members of the dome are divided into seven groups.

The dome is subjected to vertical loads on all unsupported joints, with a load of -13.49 kips (60 kN) at node 1, $-6.744\, kips\ (30\, kN)$ at nodes 2-14 and $-2.248\, kips\ (10\, kN)$ at remaining nodes. The minimum cross-sectional area of elements is $0.775\, in^2\ (5\, cm^2)$ and the constraints considered are stress based on the AISC [51] code and displacement constraints of $\pm 0.1969\, in\ (5\, mm)$.

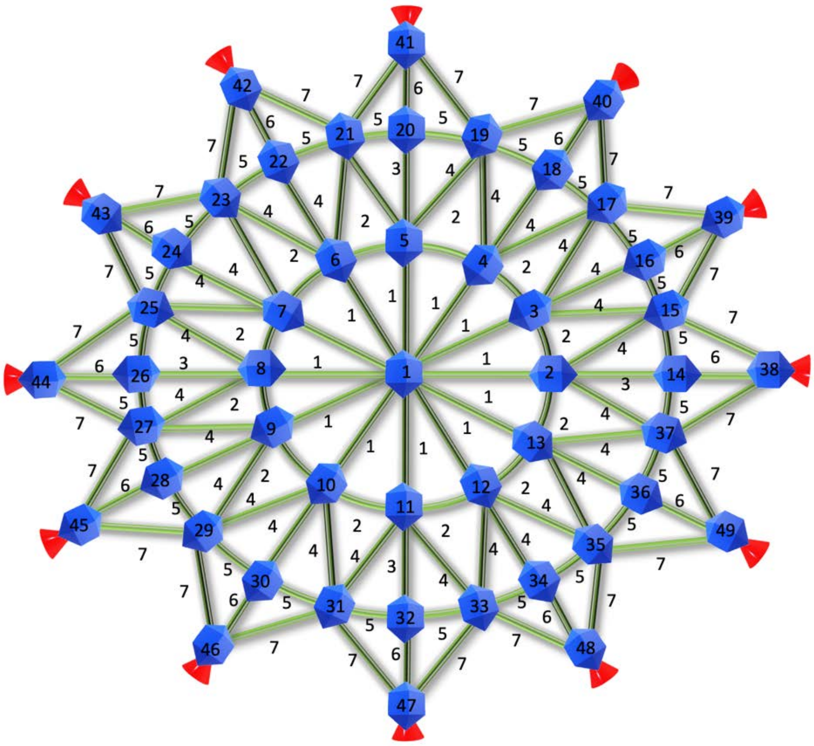

Figure 11. Schematic view of dome structure with 120 members.

Figure 12 shows the 30 independent runs carried out to solve the dome structure with 120 members design using MGO. Results for the truss-optimized weight of MGO compared with AOA, MGA and CRY are presented in Table 6, showing that MGO obtains the optimal design. Similar to 25-member and 72-member truss problems, the MGO method is able to predict the lowest weight for the structure. The authors calculate the results of the other three metaheuristics algorithms in Table 6 to compare algorithm strengths against the MGO. As reported, the MGO is able to calculate an overall weight of $33281.68\ lb$ for the truss, which is again the lowest compared to other methods, such as CRY, which is able to predict an overall weight of $34052.17\ lb$.

In terms of the statistical findings, the MGO can provide a mean of $33418.16\ lb$ and an SD of 121.10; these values have been computed most accurately up to this point for this particular truss design. Hence, the MGO algorithm can provide reasonable and economical results, as represented in Figure 13, showing that the MGO can calculate the most optimum weight for the structure. The results of achieved stresses and displacement within the members using MGO optimization runs for load cases are presented in Figure 14. The figures define stress and displacement limits, and the capability of the constraint-handling approach is in perspective.

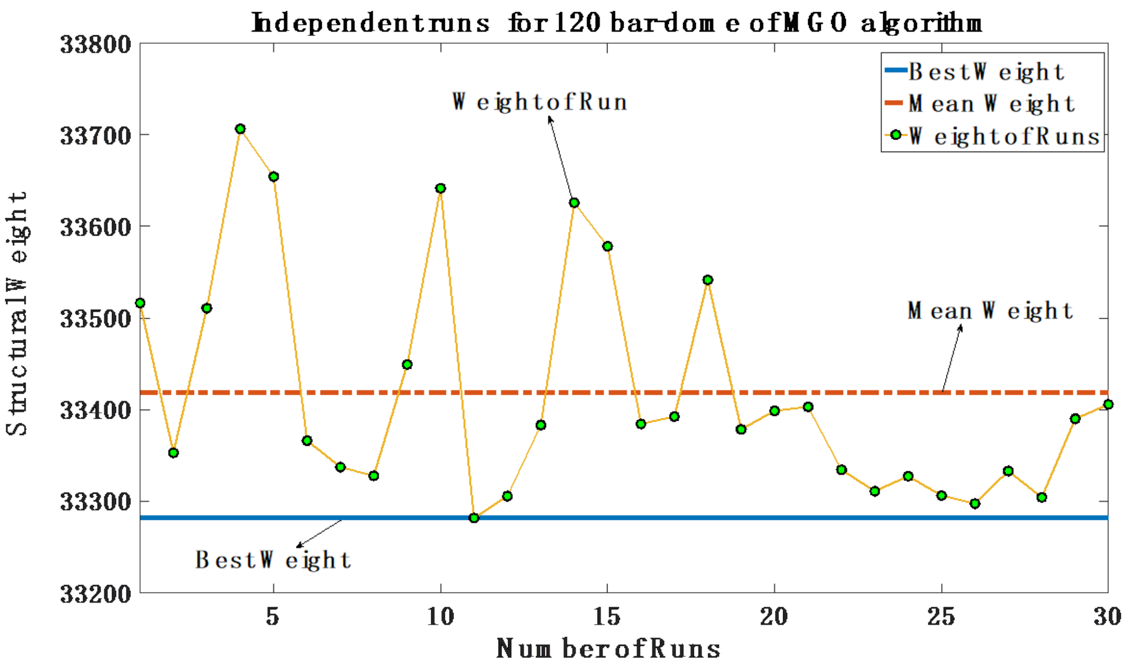

Figure 12. Best run and average runs convergence curves dome structure with 120 members with different methods.

Table 6. The MGO results in dome structure with 120 members.

Group Member	AO	CRY	MGA	MGO
1	3.99	3.134	3.589	3.026
2	20.0	12.330	8.594	15.051
3	4.96	5.276	6.170	4.982
4	3.70	3.269	3.935	3.116
5	7.41	9.413	11.411	8.280
6	3.43	3.687	5.764	3.590
7	2.57	2.765	4.206	2.496
Optimum weight (lb.)	3692	34052.17	41028.10	33281.68
Average weight (lb.)	4188	36872.80	44354.35	33418.16
Standard deviation	3266	1148.37	2526.17	121.10
Number of analyses	1000	10000	10000	2650

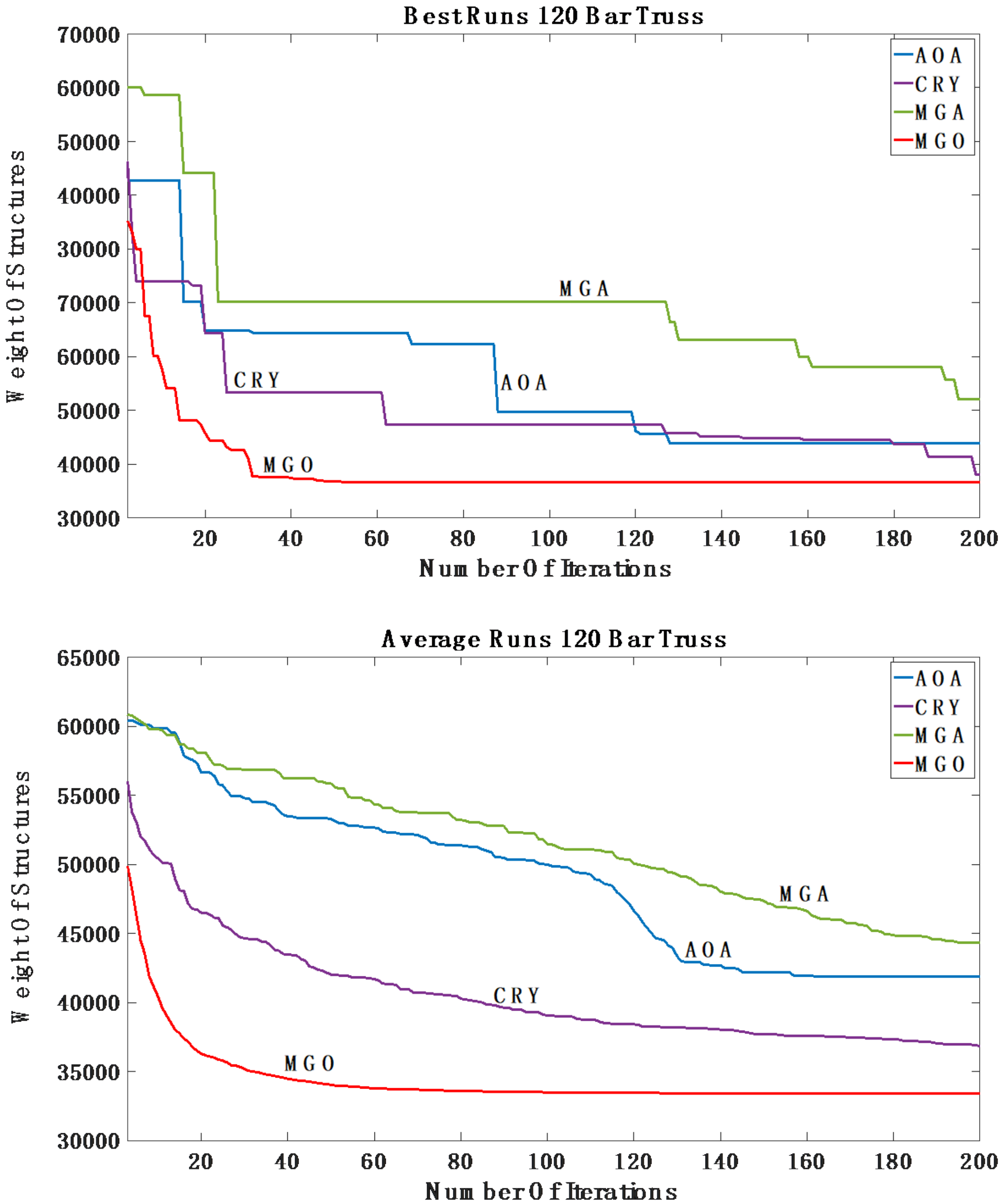

Figure 13. Best run and average runs convergence curves dome structure with 120 members with different methods.

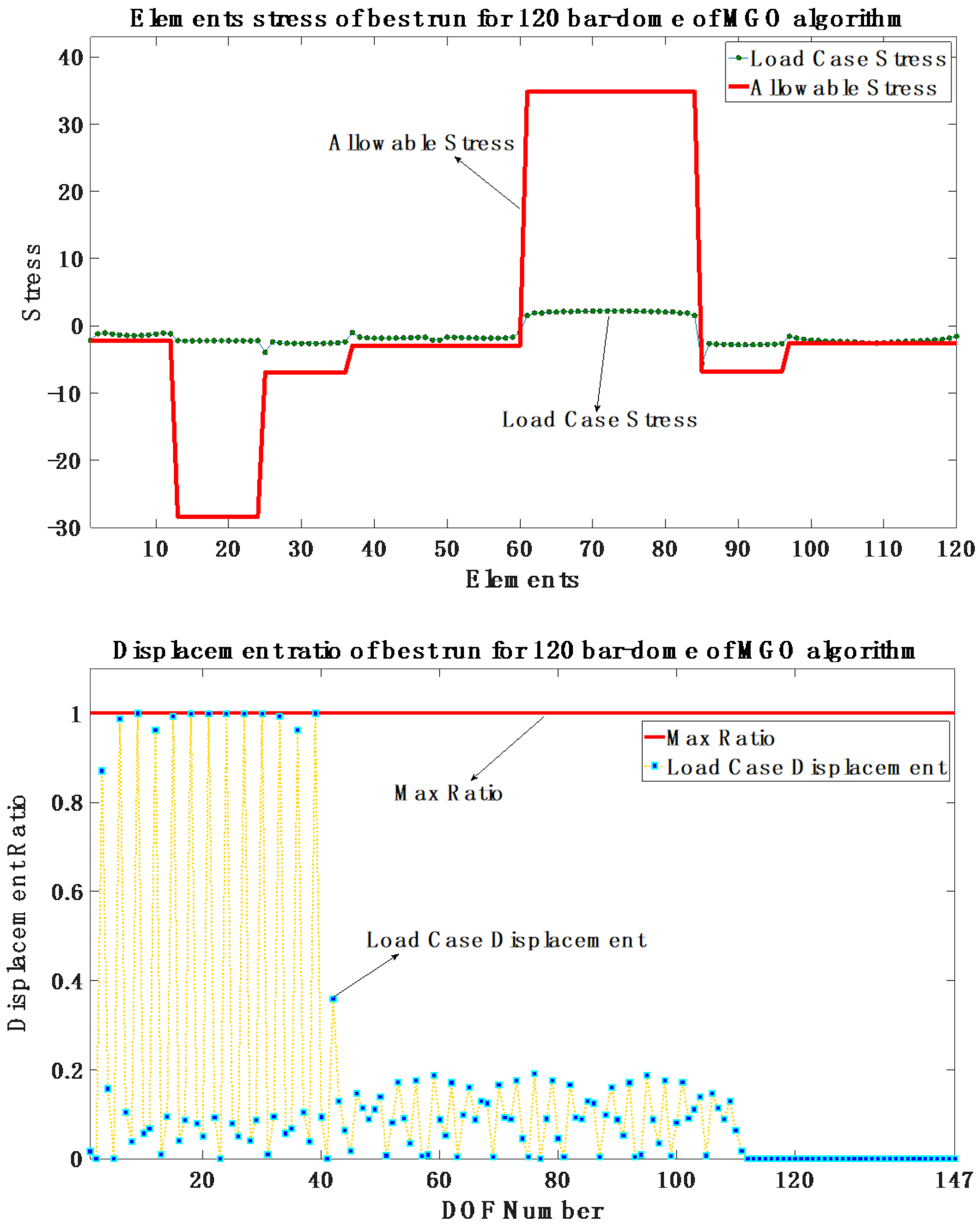

Figure 14. Stress and Displacement ratio for dome structure with 120 members with MGO.

6. Conclusions

This work introduces the Mountain Gazelle Optimizer (MGO), which is a nature-inspired meta-heuristic algorithm, as a truss optimization method. After a detailed literature review of the optimization of truss structures with metaheuristic algorithms, MGO is used to solve three different benchmark truss design examples with different orientations and geometries under predefined loading conditions, namely trusses with 25-, 72- and 120-member elements. MGO results are compared with the AOA, MGA and CRY. A penalty method was complementary to incorporate constraint handling. Several runs using optimization algorithms were carried out to get accurate statistical data to compare the results between the selected metaheuristic algorithms. In all three truss designs, the MGO algorithm provided reasonable results with the most optimum structural weight compared to other metaheuristic algorithms and the lowest number of function evaluations, which is considered the minimum value for problems with higher complexity. MGO can reduce the time required to design and analyze truss structures by finding solutions more quickly than existing methods based on the convergence rate. It can lead to more economical structures by finding better solutions with low standard deviation that are more cost-effective and lighter. This work is the first evidence showing how MGO effectively solves and optimizes structural engineering problems, such as trusses. Moreover, this work can be expanded to other structural engineering optimization problems, including binary and multi-objective versions of MGO.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgment

Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R120), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Conflict of interest

All authors declare no conflict of interest regarding the publication of this paper.

References

1. A. A. Abdelhamid, S. K. Towfek, N. Khodadadi, A. A. Alhussan, D. S. Khafaga, M. M. Eid, et al., Waterwheel Plant Algorithm: A Novel Metaheuristic Optimization Method, *Processes*, **11** (2003), 1502. https://doi.org/10.3390/pr11051502
2. W. Lingyun, Z. Mei, W. Guangming, M. Guang, Truss optimization on shape and sizing with frequency constraints based on genetic algorithm, *Comput. Mech.*, **35** (2005), 361–368. https://doi.org/10.1016/j.jsv.2014.07.027
3. M. Sonmez, Artificial Bee Colony algorithm for optimization of truss structures, *Appl. Soft Comput.*, **11** (2011), 2406–2418. https://doi.org/10.1016/j.asoc.2010.09.003
4. J. Pierezan, L. dos Santos Coelho, V. C. Mariani, E. H. de Vasconcelos Segundo, D. Prayogo, Chaotic coyote algorithm applied to truss optimization problems, *Comput. Struct.,* **242** (2021),

106353. https://doi.org/10.1016/j.compstruc.2020.106353
5. A. Kaveh, B. F. Azar, S. Talatahari, Ant colony optimization for design of space trusses, *Int. J. Sp. Struct.*, **23** (2008), 167–181. https://doi.org/10.1260/026635108786260956
6. N. Khodadadi, F. S. Gharehchopogh, B. Abdollahzadeh, S. Mirjalili, Space truss structures' optimization using metaheuristic optimization algorithms, *Comprehensive Metaheuristics*, (2023), 163–179. https://doi.org/10.1016/B978-0-323-91781-0.00009-0
7. C. V. Camp, Design of space trusses using Big Bang–Big Crunch optimization, *J. Struct. Eng.*, **133** (2007), 999–1008. https://doi.org/10.1061/(ASCE)0733-9445(2007)133:7(999)
8. A. Kaveh, S. Talatahari, N. Khodadadi, Stochastic paint optimizer: theory and application in civil engineering, *Eng. Comput.*, (2020), 1–32. https://doi.org/10.1007/s00366-020-01179-5
9. A. Kaveh, S. Talatahari, N. Khodadadi, The Hybrid Invasive Weed Optimization-Shuffled Frog-leaping Algorithm Applied to Optimal Design of Frame Structures, *Period. Polytech. Civ. Eng.*, **63** (2019), 882–897. https://doi.org/10.1007/s40996-019-00280-0
10. H. Garg, A hybrid PSO-GA algorithm for constrained optimization problems, *Appl. Math. Comput.,* **274** (2016), 292–305. https://doi.org/10.1016/j.amc.2015.11.001
11. W. Zuo, J. Bai, B. Li, A hybrid OC–GA approach for fast and global truss optimization with frequency constraints, *Appl. Soft Comput.*, **14** (2014), 528–535. https://doi.org/10.1016/j.asoc.2013.09.002
12. N. Khodadadi, E. Harati, F. De Caso, A. Nanni, Optimizing Truss Structures Using Composite Materials under Natural Frequency Constraints with a New Hybrid Algorithm Based on Cuckoo Search and Stochastic Paint Optimizer (CSSPO), *Buildings,* **13** (2023), 1551. https://doi.org/10.3390/buildings13061551
13. G. I. Sayed, A. E. Hassanien, A hybrid SA-MFO algorithm for function optimization and engineering design problems, *Complex Intell. Syst.*, **4** (2018), 195–212. https://doi.org/10.1007/s40747-018-0066-z
14. N. Khodadadi, S. Talatahari, A. Dadras Eslamlou, MOTEO: a novel multi-objective thermal exchange optimization algorithm for engineering problems, *Soft Comput.*, (2022), 1–26. https://doi.org/10.1007/s00500-022-07050-7
15. N. Khodadadi, L. Abualigah, S. Mirjalili, Multi-objective Stochastic Paint Optimizer (MOSPO), *Neural Comput. Appl.*, **34** (2022), 18035–18058. https://doi.org/10.1007/s00521-022-07405-z
16. N. Khodadadi, L. Abualigah, Q. Al-Tashi, S. Mirjalili, Multi-objective chaos game optimization, *Neural Comput. Appl.*, **35** (2023), 14973–15004. https://doi.org/10.1007/s00521-023-08432-0
17. W. Lin, D. Y. Yu, C. Zhang, X. Liu, S. Zhang, Y. Tian, et al., A multi-objective teaching−learning-based optimization algorithm to scheduling in turning processes for minimizing makespan and carbon footprint, *J. Clean. Prod.*, **101** (2015), 337–347. https://doi.org/10.1016/j.jclepro.2015.03.099
18. N. Khodadadi, S. M. Mirjalili, S. Mirjalili, Multi-objective Moth-Flame Optimization Algorithm for Engineering Problems, in *Handbook of Moth-Flame Optimization Algorithm*, CRC Press, 79–96.
19. M. Khishe, N. Orouji, M. R. Mosavi, Multi-Objective chimp Optimizer: An innovative algorithm for Multi-Objective problems, *Expert Syst. Appl.*, **211** (2023), 118734. https://doi.org/10.1016/j.eswa.2022.118734

20. N. Khodadadi, L. Abualigh, E.-S. M. EI-Kenawy, V. Snasel, S. Mirjalili, An Archive-based Multi-Objective Arithmetic Optimization Algorithm for Solving Industrial Engineering Problems, *IEEE Access*, **10** (2022), 106673–106698. https://doi.org/10.1109/ACCESS.2022.3212081
21. M. Azizi, S. Talatahari, N. Khodadadi, P. Sareh, Multi-Objective Atomic Orbital Search (MOAOS) for Global and Engineering Design Optimization, *IEEE Access*, **10** (2022), 67727–67746. https://doi.org/10.1109/ACCESS.2022.3186696
22. T. H. B. Huy, P. Nallagownden, K. H. Truong, R. Kannan, D. N. Vo, N. Ho, Multi-objective search group algorithm for engineering design problems, *Appl. Soft Comput.*, **126** (2022), 109287. https://doi.org/10.1016/j.asoc.2022.109287.
23. N. Khodadadi, S. M. Mirjalili, W. Zhao, Z. Zhang, L. Wang, S. Mirjalili, Multi-Objective Artificial Hummingbird Algorithm, *Advances in Swarm Intelligence: Variations and Adaptations for Optimization Problems*, Springer, (2022), 407–419. https://doi.org/10.1007/978-3-031-09835-2_22
24. N. Khodadadi, S. M. Mirjalili, S. Z. Mirjalili, S. Mirjalili, Chaotic Stochastic Paint Optimizer (CSPO), Proceedings of 7th International Conference on Harmony Search, *Soft Computing and Applications*, (2022), 195–205. https://doi.org/10.1007/978-981-19-2948-9_19
25. Y. Li, S. Wang, M. Han, Truss structure optimization based on improved chicken swarm optimization algorithm, *Adv. Civ. Eng.*, (2019), 1–16. https://doi.org/10.1155/2019/6902428
26. N. Khodadadi, S. Talatahari, A. H. Gandomi, ANNA: advanced neural network algorithm for optimisation of structures, *Proc. Inst. Civ. Eng. - Struct. Build.,* (2023), 1–23. https://doi.org/10.1680/jstbu.22.00083
27. E.-S. M. El-Kenawy, N. Khodadadi, S. Mirjalili, T. Makarovskikh, M. Abotaleb, F. K. Karim, et al., Metaheuristic optimization for improving weed detection in wheat images captured by drones, *Mathematics*, **10** (2022), 4421. https://doi.org/10.3390/math10234421
28. S. Chakraborty, S. Sharma, A. K. Saha, A. Saha, A novel improved whale optimization algorithm to solve numerical optimization and real-world applications, *Artif. Intell. Rev.*, (2022), 1–112. https://doi.org/10.1007/s10462-021-10114-z
29. Q. Al-Tashi, S. Mirjalili, J. Wu, S. J. Abdulkadir, T. M. Shami, N. Khodadadi, et al., Moth-Flame Optimization Algorithm for Feature Selection: A Review and Future Trends, *Handb. Moth-Flame Optim. Algorithm*, (2022), 11–34.
30. A. A. Alhussan, A. A. Abdelhamid, S. K. Towfek, A. Ibrahim, L. Abualigah, N. Khodadadi, et al., Classification of Breast Cancer Using Transfer Learning and Advanced Al-Biruni Earth Radius Optimization, *Biomimetics*, **8** (2023), 270. https://doi.org/10.3390/biomimetics8030270
31. N. Khodadadi, S. Vaclav, S. Mirjalili, Dynamic Arithmetic Optimization Algorithm for Truss Optimization Under Natural Frequency Constraints, *IEEE Access*, **10** (2022), 16188–16208. https://doi.org/10.1109/ACCESS.2022.3146374
32. E.-G. Talbi, *Metaheuristics: from design to implementation*, vol. 74. John Wiley & Sons, 2009.
33. N. Khodadadi, S. Mirjalili, Truss optimization with natural frequency constraints using generalized normal distribution optimization, *Appl. Intell.,* (2022), 1–14. https://doi.org/10.1007/s10489-021-03051-5
34. D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol.*

Comput., **1** (1997), 67–82. https://doi.org/10.1109/4235.585893

35. N. Khodadadi, A. Ö. Çiftçioğlu, S. Mirjalili, A. Nanni, A comparison performance analysis of eight meta-heuristic algorithms for optimal design of truss structures with static constraints, *Decis. Anal. J.*, (2023), 100266. https://doi.org/10.1016/j.dajour.2023.100266
36. B. Abdollahzadeh, F. S. Gharehchopogh, N. Khodadadi, S. Mirjalili, Mountain Gazelle Optimizer: A new Nature-inspired Metaheuristic Algorithm for Global Optimization Problems, *Adv. Eng. Softw.*, **174** (2022), 103282. https://doi.org/10.1016/j.advengsoft.2022.103282
37. L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A. H. Gandomi, The arithmetic optimization algorithm, *Comput. Methods Appl. Mech. Eng.*, **376** (2021), 113609. https://doi.org/10.1016/j.cma.2020.113609
38. S. Talatahari, M. Azizi, A. H. Gandomi, Material Generation Algorithm: A Novel Metaheuristic Algorithm for Optimization of Engineering Problems, *Processes,* **9** (2021), 859. https://doi.org/10.3390/pr9050859
39. S. Talatahari, M. Azizi, M. Tolouei, B. Talatahari, P. Sareh, Crystal Structure Algorithm (CryStAl): A Metaheuristic Optimization Method, *IEEE Access*, **9** (2021), 71244–71261. https://doi.org/10.1109/ACCESS.2021.3079161
40. S. O. Degertekin, Improved harmony search algorithms for sizing optimization of truss structures, *Comput. Struct.*, **92** (2012), 229–241. https://doi.org/10.1016/j.compstruc.2011.10.022
41. S. O. Degertekin, M. S. Hayalioglu, Sizing truss structures using teaching-learning-based optimization, *Comput. Struct.*, **119** (2013), 177–188. https://doi.org/10.1016/j.compstruc.2012.12.011
42. A. Kaveh, R. Sheikholeslami, S. Talatahari, M. Keshvari-Ilkhichi, Chaotic swarming of particles: a new method for size optimization of truss structures, *Adv. Eng. Softw.,* **67** (2014), 136–147. https://doi.org/10.1016/j.advengsoft.2013.09.006
43. G. Bekdaş, S. M. Nigdeli, X.-S. Yang, Sizing optimization of truss structures using flower pollination algorithm, *Appl. Soft Comput.*, **37** (2015), 322–331. https://doi.org/10.1016/j.asoc.2015.08.037
44. F. K. J. Jawad, C. Ozturk, W. Dansheng, M. Mahmood, O. Al-Azzawi, A. Al-Jemely, Sizing and layout optimization of truss structures with artificial bee colony algorithm, *Structures*, (2021), 546–559. https://doi.org/10.1016/j.istruc.2021.01.016
45. R. Awad, Sizing optimization of truss structures using the political optimizer (PO) algorithm, *Structures*, (2021), 4871–4894. https://doi.org/10.1016/j.istruc.2021.07.027
46. M. S. Gonçalves, R. H. Lopez, L. F. F. Miguel, Search group algorithm: a new metaheuristic method for the optimization of truss structures, *Comput. Struct.,* **153** (2015), 165–184. https://doi.org/10.1016/j.compstruc.2015.03.003
47. V. Goodarzimehr, F. Omidinasab, N. Taghizadieh, Optimum design of space structures using hybrid particle swarm optimization and genetic algorithm, *World J. Eng.*, 2022, in press. https://doi.org/10.1108/WJE-05-2021-0279
48. A. Kaveh, A. D. Eslamlou, N. Khodadadi, Dynamic Water Strider Algorithm for Optimal Design of Skeletal Structures, *Period. Polytech. Civ. Eng.,* **64** (2020), 904–916. https://doi.org/10.3311/PPci.16401
49. A. Kaveh, S. Talatahari, N. Khodadadi, Hybrid invasive weed optimization-shuffled

frog-leaping algorithm for optimal design of truss structures, *Iran. J. Sci. Technol. Trans. Civ. Eng.,* **44** (2019), 405–420. https://doi.org/10.1007/s40996-019-00280-0
50. C. K. Soh, J. Yang, Fuzzy controlled genetic algorithm search for shape optimization, *J. Comput. Civ. Eng.*, **10** (1996), 143–150. https://doi.org/10.1061/(ASCE)0887-3801(1996)10:2(143)
51. A. Construction, Manual of steel construction: allowable stress design, A*m. Inst. Steel Constr.* Chicago, IL, USA, 1989.