

Article

Efficient Sizing and Layout Optimization of Truss Benchmark Structures Using ISRES Algorithm

Muhammed Serdar Avcı ¹, Demet Yavuz ², Emre Ercan ¹ and Ayhan Nuhoglu ^{1,*}¹ Department of Civil Engineering, Ege University, 35040 Izmir, Turkey;

muhammed.serdar.avci@ege.edu.tr (M.S.A.); emre.ercan@ege.edu.tr (E.E.)

² Department of Civil Engineering, Faculty of Engineering, University of Van Yüzüncü Yıl, 65090 Van, Turkey; demetyavuz@yyu.edu.tr

* Correspondence: ayhan.nuhoglu@ege.edu.tr

Abstract: This paper presents a comprehensive investigation into the application of the Improved Stochastic Ranking Evolution Strategy (ISRES) algorithm for the sizing and layout optimization of truss benchmark structures. Truss structures play a crucial role in engineering and architecture, and optimizing their designs can lead to more efficient and cost-effective solutions. The ISRES algorithm, known for its effectiveness in multi-objective optimization, is adapted for the single-objective optimization of truss designs with multiple design constraints. This study encompasses a wide range of truss benchmark structures, including 10-bar, 15-bar, 18-bar, 25-bar, and 72-bar configurations, each subjected to distinct loading conditions and stress constraints. The objective is to minimize the truss weight while ensuring stress and displacement limits are met. Through extensive experimentation, the ISRES algorithm demonstrates its ability to efficiently explore the solution space and converge to optimal solutions for each truss benchmark structure. The algorithm effectively handles the complexity of the problems, which involve numerous design variables, stress constraints, and nodal displacement limits. A comparative analysis is conducted to assess the performance of the ISRES algorithm against other state-of-the-art optimization methods reported in the literature. The comparison evaluates the quality of the solutions and the computational efficiency of each method. Furthermore, the optimized truss designs are subjected to finite element analysis to validate their structural integrity and stability. The verification process confirms that the designs adhere to the imposed constraints, ensuring the safety and reliability of the final truss configurations. The results of this study demonstrate the efficacy of the ISRES algorithm in providing practical and reliable solutions for the sizing and layout optimization of truss benchmark structures. The algorithm's competitive performance and robustness make it a valuable tool for structural engineers and designers, offering a versatile and powerful approach for complex engineering optimization tasks. Overall, the findings contribute to the advancement of optimization techniques in structural engineering, promoting the development of more efficient and cost-effective truss designs for a wide range of engineering and architectural applications. The study's insights empower practitioners to make informed decisions in selecting appropriate optimization strategies for complex truss-design scenarios, fostering advancements in structural engineering and sustainable design practices.

Keywords: truss optimization; ISRES; sizing optimization; layout optimization; structural engineering

Citation: Avcı, M.S.; Yavuz, D.; Ercan, E.; Nuhoglu, A. Efficient Sizing and Layout Optimization of Truss Benchmark Structures Using ISRES Algorithm. *Appl. Sci.* **2024**, *14*, 3324. <https://doi.org/10.3390/app14083324>

Academic Editor: Juan A. Gómez-Pulido

Received: 18 March 2024

Revised: 4 April 2024

Accepted: 10 April 2024

Published: 15 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Optimization is in our lives everywhere we turn. Athletes preserve their breath to run/swim longer/faster, people use the shortest path to their office in order to spend less time on the road, and engineers design more green and eco-friendly buildings to save energy while satisfying consumers' needs. In all these examples, the main goal is to use existing resources wisely and economically. Structural optimization basically relies on the same fundamentals. The aim of structural optimization is to perform better under

specified goals while meeting variable constraints. In previous studies, gradient-based algorithms were used considerably for structural optimization [1,2]. While these types of algorithms can sometimes help to reach a solution, without a decent starting point obtaining a solution is not guaranteed. Meta-heuristic algorithms, on the other hand, are free from this restriction. These types of algorithms need initial data, decision and state variables, constraints, etc. Initial states can be calculated, assigned randomly, or used as default [3].

Structural optimization is practiced under three different categories, namely: size, shape, and topology. The aim for optimization in truss elements is to reduce the structures' weight. While optimization could be carried out with the use of a single category of these above-mentioned options, it can be also reached with different combinations of any two or altogether. Size is the largely applied category for truss optimization, followed by shape and topology [3]. The research gap falls between topology and shape, with very little work available in the literature. In truss optimization, weight, geometry, stress, stiffness, nodal displacements, and buckling problems are the most largely studied types of functions [4]. While the aim is reducing the weight of the truss with optimization, different optimization categories offer different types of solutions. In sizing optimization, the aim is to find the optimum cross-sectional area of the truss. In shape optimization, in changing the coordinates of the nodals, finding an optimum position is possible, and finally, in topology optimization, removing the structural members from their original position allows for minimizing the volume of the truss [5].

Meta-heuristic algorithms have been applied to truss optimization for a long time. Meta-heuristic algorithms used by researchers frequently are as follows: genetic algorithm [6], Cuckoo Search [7], Big Bang-Big Crunch [8], particle swarm optimization [9], the Firefly algorithm [10], Harmony search [11], differential evolution [12], and the bat algorithm [13].

Miguel et al. [14] used the Firefly algorithm in truss structures to optimize simultaneously size, shape, and topology. Four benchmark problems such as 11-bar, 39-bar, 25-bar, and 15-bar trusses were studied. As a result, it was decided that this method is useable for mixed-optimization problems. Khodadadi et al., 2023, [15] examined the effectiveness of eight population-based meta-heuristic approaches: the African Vultures Optimization Algorithm (AVOA), the Flow Direction Algorithm (FDA), the Arithmetic Optimization Algorithm (AOA), Generalized Normal Distribution Optimization (GNDO), the Stochastic Paint Optimizer (SPO), the Chaos Game Optimizer (CGO), the Crystal Structure Algorithm (CRY), and the Material Generation Algorithm (MGO). The objective is to optimize the dimensions of three aluminum truss structures with the goal of minimizing the weight of truss members while satisfying specified displacement and stress constraints. The study evaluates the performance of these methods by applying them to three established truss-structure benchmarks under certain constraints. The findings indicate that the Stochastic Paint Optimizer (SPO) outperforms the other algorithms in terms of both accuracy and convergence rate. Assimi and Jamali [16] applied hybrid genetic programming (HGP) to truss structures with different static and dynamic constraints. Benchmark problems, namely 24-bar, 20-bar, and 72-bar, were investigated with different variables for sizing and topology problems. The results obtained from this study were compared against other relative studies, and HGP showed a better performance for most of the cases. Jafari et al. [17] proposed a new model for the optimization of truss structures with (PSO) and the cultural algorithm (CA). Together, 10-bar, 25-bar, 72-bar, 120-bar, and 160-bar truss structures were used for checking algorithm viability. They reported that the new proposed model was superior or competitive in comparison with other meta-heuristic algorithms. With the inspiration of the Jaya algorithm (JA), Degertekin et al. [18] developed a parameter-free jaya algorithm (PFJA) for sizing/layout problems of truss structures and compared the outputs with the standard jaya algorithm (JA), the modified jaya algorithm (MJJA), and other meta-heuristic algorithms in the literature. Optimization was conducted on 37-bar, 52-bar, 72-bar, 200-bar, 942-bar, 600-bar, 1180-bar, and 1410-bar truss structures. With eight

different truss weight optimization problems studied in this paper, it is stated that PFJA is a reliable algorithm in terms of minimizing truss weight and convergence speed. For size/shape optimization, Azizi et al. [19] applied chaos game optimization with frequency constraints. Five different truss structures such as 10-bar, 37-bar, 52-bar, 72-bar, and 120-bar structures were considered. They reported that with the mean of 30 independent runs, chaos game optimization (CGO) reached the minimum truss weight among the algorithms considered in this study. In another recent study, Jawad et al. [20] employed the artificial bee colony algorithm (ABC) for 15-bar, 18-bar, 25-bar, and 47-bar truss structures for sizing and layout optimization. Stress, displacement and buckling constraints were considered. The authors reported that the ABC algorithm outperformed other algorithms with regard to weight and iteration. Kumar et al., 2021, focuses on four objective functions: mass, compliance, first natural frequency, and buckling factor [21]. Due to the limited availability of optimization methods designed for addressing many-objective truss-optimization challenges, it becomes crucial to evaluate the performance of contemporary algorithms on these complex problems. This research contributes by examining the relative effectiveness of eighteen established algorithms across various dimensions. The assessment is based on four metrics, aiming to solve intricate truss problems with many objectives. Statistical analysis is conducted considering the best mean and standard deviation outcomes of the objective functions, along with Friedman's rank test. The overall comparison identifies MMIPDE as the top-performing algorithm, with SHAMODE utilizing a whale-optimization approach and the standard SHAMODE as the runners-up. Dehghani et al. [22] inspired by the imperialist competitive algorithm (ICA), cellular automata (CA), and the dolphin echolocation (DE) algorithm, introduced a new algorithm abbreviated as CA-ACEA. They dictated that the proposed algorithm outperforms other algorithms considered in this study for weight minimization. Pierezan et al. [23] applied the chaotic coyote algorithm (COA) for the 52-bar, 72-bar, 120-bar, and 200-bar for truss optimization to minimize the structure weight. They also proposed a modified model of the chaotic coyote algorithm (MCOA) and reported that compared to the original COA, the newly proposed algorithm showed competitive results. Pholdee and Bureerat [24] aimed to minimize the weight of the trusses with dynamic constraints and applied various meta-heuristic algorithms. Five different truss-optimization problems were considered to evaluate the performance of these algorithms. They reported that with different benchmark problems, the best algorithm to minimize the weight of the truss changed case to case. Serpik [25] proposed a meta-heuristic algorithm based on a job-search strategy. Five different numerical examples, namely, 10-bar, 25-bar, 200-bar, 18-bar, and 354-bar, were considered for efficiency analyses. The authors' goal was to rule out penalty functions. Performing selected benchmark problems showed that the proposed procedure displayed effective solutions for optimization.

Particle swarm optimization (PSO) is one of the frequently used meta-heuristic algorithms for truss optimization. Researchers used this algorithm for different benchmark problems, modified and improved over the years [26–30]. Cao et al. [26] proposed an enhanced version of PSO in truss structures to decrease the structural-analysis process. Size and shape optimization was considered. The results stated that enhanced PSO reduces the computational time required for optimization without sacrificing any search capability. In truss structures, Luh and Lin [27] investigated the optimization process under different constraints such as deflection, stress, and kinematic stability with two-stage particle swarm optimization. It is reported that this approach outperformed other methodologies reported in the literature. Li et al. [28] used heuristic particle swarm optimization (HPSO) based on standard particle swarm optimization (PSO) for truss structures. They noted that HPSO was able to expedite convergence speed and can be efficient for steel structures. Gomes [29] applied PSO and compared it with other algorithms and emphasized that the optimization results obtained with PSO are similar to other known algorithms, or, in some benchmark problems, even better. Some researchers such as Kim and Byun [30] proposed a modified and improved version of PSO for truss optimization.

Another algorithm considered in this study was differential evolution (DE). DE is a widely used algorithm for optimization problems [31–38]. Over the years, authors contributed to the efficiency of the algorithm with different approaches. Kao et al. [36] suggested a new plan to enhance DE performance by altering the mutation operator. Tang et al. [35] proposed teaching-based DE, while Ho-Huu et al. [37] used DE based on a roulette selection for trusses.

From the moment it was introduced to the literature, the genetic algorithm (GA) has been used for all kinds of optimization problems. In this section, a GA focused on truss optimization in recent years is presented [39–46]. Assimi et al. [39] used the GA to minimize the cross-section area of truss systems; Liu and Xia [40] proposed a hybrid intelligent genetic algorithm (HIGA) and reported that the new method is a convenient tool for truss optimization. Noii et al. [41] and Dong et al. [42] proposed a new approach to improve the GA, while Delyová et al. [43] used the GA for size and topology optimization, and Assimi et al. [46] introduced a new mutant operator for the GA.

In this study, our main objective is to perform a single-objective optimization of truss benchmark structures using the Improved Stochastic Ranking Evolution Strategy (ISRES) algorithm. Truss structures are widely used in engineering and architecture, and optimizing their designs for a specific objective can lead to more efficient and cost-effective solutions. The ISRES algorithm is a powerful optimization technique that belongs to the class of evolutionary algorithms. It efficiently explores the solution space to search for the optimal solution, focusing on a single objective in this case. The ISRES algorithm utilizes selection, mutation, and recombination operations to iteratively improve the candidate solutions until convergence to the optimal solution is achieved.

The optimization process will begin by formulating the truss-design problem as a single-objective optimization task. The key design variables, such as the cross-sectional areas and the lengths of truss members, will be considered as decision variables to be optimized. The objective function will be defined based on the specific criterion that needs to be maximized (e.g., structural stiffness, load-carrying capacity, or cost-effectiveness).

To evaluate the performance of each truss design, we will use structural analysis software or finite element methods to calculate the objective function value. The structural analysis will consider the given constraints to ensure the safety and stability of the truss.

The ISRES algorithm will then be employed to search for the optimal truss design that maximizes the defined objective function. During the optimization process, we will fine-tune the algorithm's parameters and control settings to ensure its effectiveness and efficiency in finding the optimal solution.

Multiple runs of the ISRES algorithm may be performed to account for the impact of randomness and obtain a more reliable optimal design solution.

After obtaining the optimal design solution, we will perform a detailed analysis of the truss to understand its performance characteristics and compare it with other benchmark solutions. This will provide valuable insights into the influence of different design variables on the objective function and guide engineers and designers in making informed decisions for real-world applications.

Overall, this study aims to demonstrate the effectiveness of the ISRES algorithm in the single-objective optimization of truss benchmark structures. The results obtained will provide valuable guidance for engineering and architectural applications, promoting the development of more efficient and reliable truss designs tailored to specific performance objectives.

2. Problem Formulations

The main purpose of this study is to minimize the weights of benchmark truss structures with different constraints and variables; the value of the objective function of the optimization procedure is the total weight of the structure, as given in Equation (1)

$$\min W(X_Z, X_s) = \rho \sum_{i=1}^M L_i A_i \quad (1)$$

This objective function is subjected to the following:

$$\begin{aligned}
 & \text{Displacement constraints,} & \delta_{min} \leq \delta_j \leq \delta_{max} & j = 1, 2, \dots, J \\
 & \text{Stress constraints,} & \sigma_{min} \leq \sigma_i \leq \sigma_{max} & i = 1, 2, \dots, M \\
 & \text{Cross – sectional area constraints,} & A_{min} \leq A_k \leq A_{max} & k = 1, 2, \dots, G \\
 & \text{Buckling constraints,} & \left| \sigma_i^{comp} \right| - \sigma_i^{cr} \leq 0 \text{ where } \sigma_i^{cr} = \frac{k_i A_i E_i}{L_i^2} & i = 1, 2, \dots, M
 \end{aligned}$$

In this context, the variables are defined as follows: W represents the weight of the truss; X_Z and X_S denote the size and geometry variables, respectively. The allowable displacements for the joints are denoted as δ_{max} and δ_{min} , while σ_{min} and σ_{max} represent the allowable stress for the bars, respectively. A_i (in) represents the cross-sectional area of each bar, constrained within upper and lower limits A_{max} and A_{min} . ρ represents the material density, and finally L_i (in) represents the length of each bar ‘ i ’, respectively. k_i represents the buckling coefficient. σ_i^{cr} denotes the critical buckling stress, and σ_i^{comp} represents the stress under compression. E is the Young’s modulus of elasticity. It is imperative that each design case adheres to its specific constraints, such as displacement (δ_j) for each joint ‘ j ’ and stresses (σ_i) for each member ‘ i ’. ‘ J ’ signifies the number of joints, ‘ M ’ indicates the number of bars, and ‘ G ’ refers to the number of element groups, which correlates with the number of sizing variables.

In the scope of this study, the truss benchmark problems exhibit different constraint scenarios. Some nodes have displacement constraints, while others do not. However, stress constraints are valid for all elements in the problems. The presence of these constraints can vary across different problems, with some having all constraints, and others having only a subset of them.

To effectively handle constraint violations during optimization, penalty functions are introduced. Penalty methods are a specific class of algorithms used for solving constrained optimization problems. These methods convert constrained optimization problems into unconstrained ones, allowing the solutions to converge to the optimal solution of the original problem. The unconstrained problems are formulated as follows:

$$W'(X_Z, X_S) = W(X_Z, X_S) + K(g_j(x)) \quad \text{for } j = 1, 2, \dots, N_i$$

where

K is the penalty coefficient and W' is the fitness function and g_j is the constraint violation. Additionally, it is worth noting that the value of K can vary depending on the problem at hand. In our study, we have chosen K to be one, which signifies that we have penalized the objective function by the amount of constraint violation. This decision was made to effectively manage constraint violations and steer the optimization process towards feasible solutions.

Constraint violation is the summation of how much a solution is violated, where the constraints such as a stress in an element are higher than the allowed amount. The penalties for the aforementioned constraints are given below:

$$\begin{aligned}
 CV &= C_d + C_s + C_b \\
 CV &= \sum_j^{N_j} CV_\delta^j + \sum_j^{N_k} CV_\sigma^j + \sum_j^{N_m} CV_b^j \\
 C_s &= \begin{cases} \left| \frac{\delta_j - \delta_{all}}{\delta_{all}} \right| & \text{if } \delta_j > |\delta_{all}| \\ 0 & \text{else } \delta_j \leq |\delta_{all}| \end{cases}
 \end{aligned}$$

$$C_d = \begin{cases} \left| \frac{\sigma_j - \sigma_{all}}{\sigma_{all}} \right| & \text{if } \sigma_j > |\sigma_{all}| \\ 0 & \text{else } \sigma_j \leq |\sigma_{all}| \end{cases}$$

$$C_b = \begin{cases} \left| \frac{\sigma_j - \sigma_{all}}{\sigma_{all}} \right| & \text{if } \sigma_j < 0 \text{ and } \sigma_j < -\frac{kEA_l}{L_l^2} \\ 0 & \text{else } \sigma_j > 0 \text{ or } \sigma_j > -\frac{kEA_l}{L_l^2} \end{cases}$$

where

CV represents the total constraint violation in the optimization problem, which is decomposed into three components: displacement constraint violation (C_d), stress constraint violation (C_s), and buckling constraint violation (C_b). For displacement constraint violation, C_d , it is calculated as the sum of the absolute differences between individual displacement values (δ_j) and the allowable displacement limit (δ_{all}) divided by the allowable displacement limit. If the individual displacement (δ_j) exceeds the allowable limit (δ_{all}), the constraint violation is penalized; otherwise, it is considered to be zero. C_s , the stress constraint violation, is computed similarly to C_d . It involves the sum of the absolute differences between individual stress values (σ_j) and the allowable stress limit (σ_{all}), divided by the allowable stress limit. If the individual stress (σ_j) surpasses the allowable limit (σ_{all}), the constraint violation is accounted for; otherwise, it remains zero. C_b , representing the buckling constraint violation, is determined based on the individual stress values (σ_j). If the stress is negative ($\sigma_j < 0$) and it falls below a critical buckling threshold $\frac{kEA_l}{L_l^2}$, then the constraint is violated and penalized accordingly. Otherwise, if the stress is either positive or above the critical threshold, the constraint violation is considered to be zero.

In summary, the total constraint violation (CV) is the sum of the constraint violations for displacement, stress, and buckling, each computed according to the specified equations and conditions.

3. Improved Stochastic Ranking Evolutionary Strategy (ISRES)

Using the penalty function method for certain issues can be successful; however, it is hard to determine what an optimal (or nearly optimal) value for it should be. If the value is too small, then infeasible solutions may not be penalized enough, allowing for them to be generated by the evolutionary algorithm. A “small” penalty function value implies a relatively low penalty applied to infeasible solutions. In such cases, the penalty may not be sufficient to discourage the generation of infeasible solutions by the evolutionary algorithm, potentially leading to suboptimal outcomes. Conversely, a value that is too high will cause feasible solutions to be found, but of poor quality. A “large” penalty function value indicates a higher penalty imposed on infeasible solutions. While this may effectively discourage the generation of infeasible solutions, excessively high penalties can lead to premature convergence to suboptimal feasible solutions. Moreover, a large penalty value can restrict the exploration of the search space, particularly in regions where feasible and infeasible solutions are intertwined or separated. To maximize the effectiveness of the exploration of infeasible regions, it is best to consider an appropriate value for the problem. Depending on the situation, different values may be necessary. Finding an optimal or near-optimal value for the penalty function is crucial for balancing the trade-off between penalizing infeasible solutions and promoting the exploration of the search space. The appropriate value depends on the specific characteristics of the optimization problem, such as the nature of the constraints and the distribution of feasible regions in the search space. A penalty value that is too small will suppress any exploration of unfeasible regions, even in the beginning of the evolution. This is especially ineffective for problems where feasible regions are disconnected from each other in the search space. Exploring infeasible regions can act as pathways linking two or more feasible areas. The key concern is how much exploration of infeasible regions (i.e., how large the value is) should be considered a reasonable amount. This is dependent on the problem. Even for the same issue, different stages of evolutionary search may need different values [47].

Different values can create different fitness functions; therefore, what is considered a fit individual under one fitness function may not be considered fit under another. Consequently, it is necessary to rank individuals in a population adaptively in order to find near-optimal solutions. To do this, Runarsson and Yao [47] introduced a novel rank-based selection method for ranking individuals, which does not involve the specification of a value. To assess the effectiveness and efficiency of this method, experimental studies have been conducted, and it can be seen as an exterior penalty approach [48].

Adaptively finding a near-optimal solution is the same as adaptively ranking individuals in a population. Therefore, the problem becomes how to sort individuals based on their objective and penalty scores. The SRES ranks the individuals without assigning a value.

The authors who developed the SRES created an improved version of it named the ISRES. The ISRES presents a modified version of the optimization algorithm that uses the exponential averaging of trial step sizes. The Improved Stochastic Ranking Evolutionary Strategy (ISRES) algorithm, a refinement of the Stochastic Ranking Evolutionary Strategy (SRES), introduces a suite of enhancements tailored to enhance efficiency and efficacy in optimization endeavors. Foremost among these enhancements is the implementation of the exponential averaging of trial step sizes, fostering a more nuanced and adaptive exploration of the search space compared to the fixed step sizes utilized by the SRES. This dynamic adjustment of step sizes based on historical data empowers the ISRES to adapt more effectively to the nuances of the optimization problem, thereby facilitating a more efficient exploration of promising regions. Furthermore, ISRES incorporates improved strategies for initializing parameters within predefined bounds, ensuring a more diverse and effective search from the outset. Its dynamic parameter-update mechanism continuously adjusts parameters, including step sizes, based on mutation outcomes, thereby enhancing convergence and exploration efficiency throughout the optimization process. ISRES also refines replication mechanisms to preserve diversity among candidate solutions, thereby staving off premature convergence and promoting the exploration of diverse regions within the search space. Moreover, the ISRES implements robust strategies for handling boundary constraints, such as allowing mutations that fall outside bounds to be retried multiple times before resorting to parent values. These mechanisms collectively ensure the comprehensive accessibility of the search space, driving superior performance, adaptability, and efficiency compared to the SRES. In sum, the ISRES represents a significant leap forward in optimization algorithms, particularly adept at tackling complex optimization tasks featuring high-dimensional search spaces and diverse constraints. The algorithm is outlined by the pseudocode in Figure 1. The exponential smoothing is performed on line 10. The parameters and step sizes are initially set to random values within the bounds (lines 1 and 2). Any mutation that falls outside of the bounds is retried up to 10 times before being set to its parent value. This approach ensures that the entire search space is initially accessible.

```

1  Initialize:  $\sigma'_k := (\bar{x}_k - \underline{x}_k) / \sqrt{n}$ ,  $x'_k = \underline{x}_k + (\bar{x}_k - \underline{x}_k) \cup_k (0, 1)$ 
2      while termination criteria not satisfied do
3          evaluate  $f(x'_k), g^+(x'_k), k = 1, \dots, \lambda$ 
4          rank the  $\lambda$  points and copy the best  $\mu$  in their ranked order
5           $(x_i, \sigma_i) \leftarrow (x'_{i,\lambda}, \sigma'_{i,\lambda}), i = 1, \dots, \mu$ 
6          for  $k:=1$  to  $\lambda$  do (replication)
7               $i \leftarrow \text{mod}(k - 1, \mu) + 1$  (cycle through the best  $\mu$  points)
8               $\sigma'_{k,j} \leftarrow \sigma'_{i,j} \exp(\tau N(0, 1) + \tau N_j(0, 1)), j = 1, \dots, n$ 
9               $x'_k \leftarrow x_i + \sigma'_k N(0, 1)$ 
10              $\sigma'_k \leftarrow \sigma_i + \alpha(\sigma'_k - \sigma_i)$ 
11         od
12     od

```

Figure 1. Pseudocode for the improved SRES algorithm (ISRES).

4. Benchmark Problems

The number of bars in each configuration directly influences the problem's dimension and complexity. In structural-optimization problems, the number of bars typically corresponds to the number of decision variables in the optimization problem, thus directly impacting the problem's dimensionality. As the number of bars increases, so does the dimensionality of the problem, making it more complex to solve. Additionally, a higher number of bars often implies a larger search space, leading to increased computational effort required for optimization.

In the scope of this paper, the ISRES algorithm was tested on five fairly small trusses with different loading conditions and design constraints. The direct stiffness method of analysis was used, and the results were compared to those found in the literature. Twenty independent runs were tabulated to account for the algorithm's stochastic nature, and the best and mean weights as well as the standard deviation were recorded.

4.1. 10-Bar Truss Structure

The 10-bar planar truss structure has been a common problem studied in the literature and has been used as a sizing benchmark by many researchers. Figure 2 displays the layout and loadings of the truss. Two loading cases were investigated: Case I with a load value of $P_1 = 100$ kips and $P_2 = 0$, and Case II with a load value of $P_1 = 150$ kips and $P_2 = 50$ kips. The material used to construct the truss had a mass density of 0.10 lb./in^3 and elastic modulus of $10,000 \text{ ksi}$. The size of each member was taken from a range between $A_{\min} = 0.1 \text{ in}^2$ and $A_{\max} = 35.0 \text{ in}^2$. The allowable member stresses and nodal displacement limits were set as $\pm 25 \text{ ksi}$ and $\pm 2 \text{ in}$ for all free nodes in the x and y direction, respectively. In total, the problem had 32 non-linear design constraints (10 tension, 10 compression, and 12 displacement constraints) and 14 discrete design variables. In order to handle the design constraints effectively during the optimization process, penalty functions were introduced. Penalty methods are a certain class of algorithms for solving constrained optimization problems. They convert constrained optimization problems into unconstrained ones by modifying the objective function to penalize constraint violations.

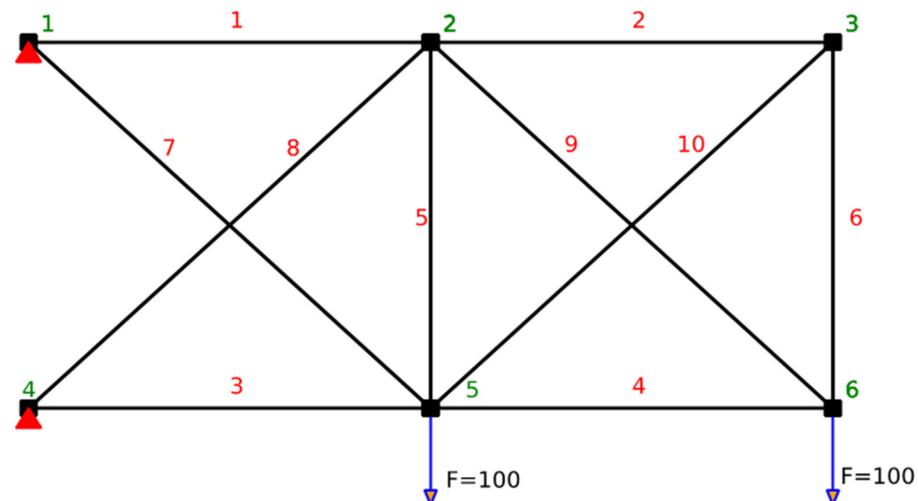


Figure 2. Schematic of the 10-bar structure.

The optimization process involved formulating the single-objective optimization problem for the truss structure with the weight as the objective to be minimized. The design variables (member sizes) were optimized by using the ISRES algorithm, which efficiently explores the solution space and searches for the optimal design.

For each loading case, the ISRES algorithm was executed multiple times with appropriate parameters and control settings to obtain diverse candidate solutions. The penalty

functions were utilized to ensure that the final solutions adhered to the specified stress and displacement constraints.

Through extensive experimentation and fine-tuning the algorithm, the ISRES approach effectively converged to optimal solutions that minimized the weight of the truss while satisfying all the design constraints. The optimization results revealed well-performing designs that met the structural requirements for both loading scenarios.

In addition, the results obtained from the ISRES algorithm were compared to those achieved through other optimization methods reported in the literature. This comparison allowed for an assessment of the ISRES algorithm's performance in terms of convergence speed, solution diversity, and solution quality. After applying the ISRES algorithm to both load case 1 and load case 2, the convergence graph for each case, as depicted in Figure 3, provided a dynamic snapshot of the optimization process. The graph vividly portrays the algorithm's iterative journey towards optimal solutions. The descending trendlines demonstrate the algorithm's persistent effort in refining the truss designs by minimizing their weight while satisfying the allowable stress limits. The consistent downward trajectory signifies the algorithm's ability to progressively enhance solution quality through each iteration, reflecting its effectiveness in converging to design configurations that strike a balance between structural performance and material efficiency.

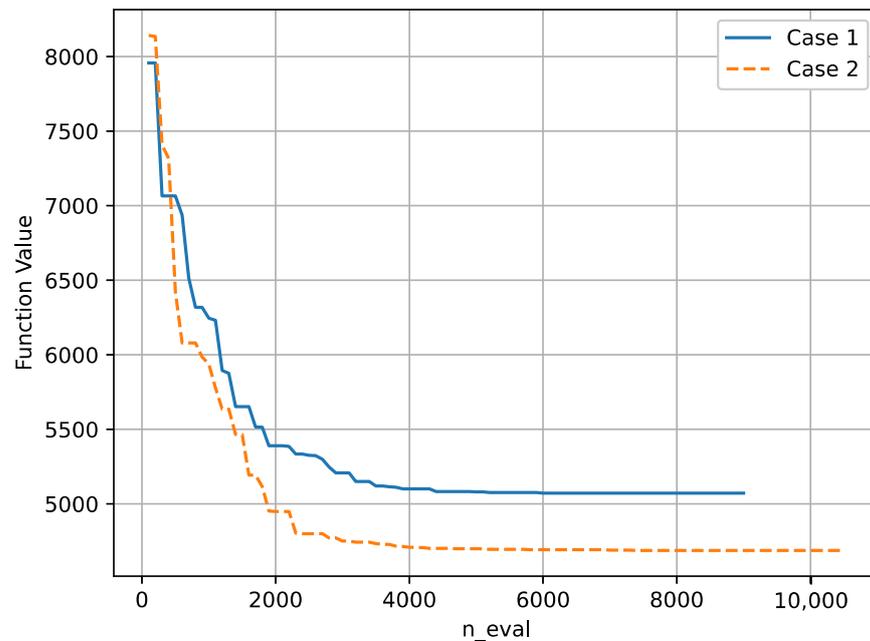


Figure 3. Convergence plot for 10-bar structure.

Figure 4 shows the constraints graph for load case 1 and load case 2 with allowable stress, and it offers a comprehensive visual representation of the algorithm's ability to manage multiple design constraints. The graph elegantly illustrates how the ISRES algorithm strategically guides the optimization trajectory within the permissible stress regions, assuring that the final truss designs align with the designated allowable stress thresholds. The intricate interplay between the optimization path and the constraint boundaries highlights the algorithm's prowess in navigating the intricate trade-offs between design efficiency and structural integrity for both loading scenarios.

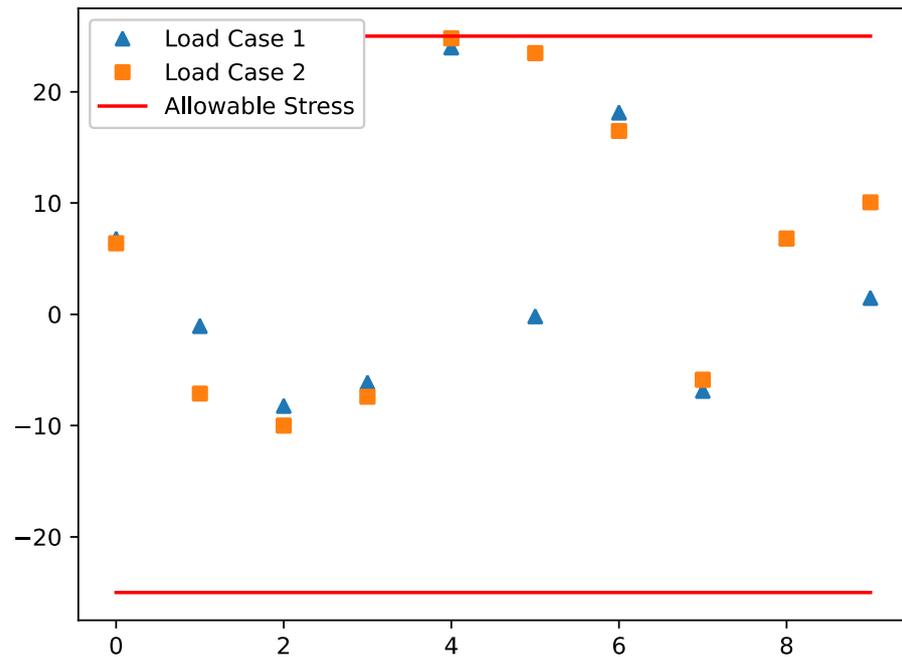


Figure 4. Stress constraint values with allowable limits for 10-bar structure.

4.2. 15-Bar Truss Structure

In this example, the sizing and layout optimization of a 15-bar planar truss structure is taken into consideration. The initial geometry of the structure is shown in Figure 5. A vertical load of 10 kips is applied at node 8. The stress limit is 25 ksi (172.369 MPa) in both tension and compression for all members. The material density is 0.1 lb./in³ (2767.99 kg/m³), and the modulus of elasticity is 10,000 ksi (68,947.6 MPa). All detailed information regarding this structure is given in Table 1. The ISRES algorithm is once again employed to optimize the truss structure for the minimum weight while satisfying the given constraints. The discrete design variables, representing the cross-sectional areas of each member, are optimized to attain the objective of minimizing the truss weight.

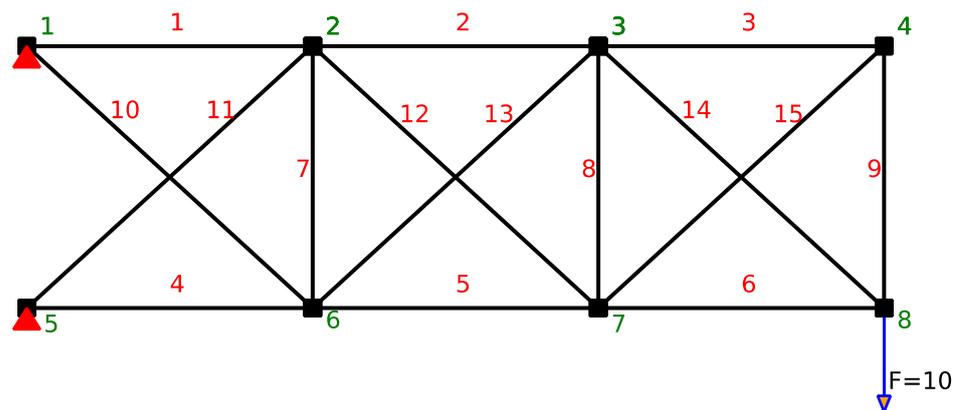


Figure 5. Schematic of 15-bar structure.

By running the ISRES algorithm multiple times with carefully selected parameters and control settings, the optimization process converges to a set of optimal solutions for the truss sizing and layout. These solutions are analyzed and compared to evaluate their performance and efficiency.

In the comparison phase, the results obtained from the ISRES algorithm are juxtaposed with those achieved using other optimization methods reported in the literature. The

comparison includes alternative evolutionary algorithms, gradient-based methods, and other metaheuristic approaches. The goal is to assess the ISRES algorithm’s superiority or competitiveness in terms of solution quality and computational efficiency.

Table 1. Element grouping for 15-bar structure.

Size Variables	Layout Variables
$A_i, i = 1, 2, \dots, 15$	$x_2 = x_6; x_3 = x_7; y_2; y_3; y_4; y_6; y_7; y_8$
Possible Sizes	
$A_i \in S = \{0.111, 0.141, 0.174, 0.220, 0.270, 0.287, 0.347, 0.440, 0.539, 0.954$ $1.081, 1.174, 1.333, 1.488, 1.764, 2.142, 2.697, 2.800, 3.131, 3.565, 3.813,$ $4.805, 5.952, 6.572, 7.192, 8.525, 9.300, 10.850, 13.330, 14.290, 17.170,$ $19.180\}(in^2)$	
Possible Coordinates	
$100 \text{ in.} \leq x_2 \leq 140 \text{ in.};$ $220 \text{ in.} \leq x_3 \leq 260 \text{ in.};$ $100 \text{ in.} \leq y_2 \leq 140 \text{ in.};$ $100 \text{ in.} \leq y_3 \leq 140 \text{ in.};$ $50 \text{ in.} \leq y_4 \leq 90 \text{ in.};$ $-20 \text{ in.} \leq y_6 \leq 20 \text{ in.};$ $-20 \text{ in.} \leq y_7 \leq 20 \text{ in.};$ $20 \text{ in.} \leq y_8 \leq 60 \text{ in.};$	
Young’s modulus $E = 104 \text{ (ksi)}$ Material density $\rho = 0.1 \text{ (lb./in}^3\text{)}$	

Furthermore, the study examines the practicality of the optimized truss designs by conducting a finite element analysis to verify their structural integrity and stability. The optimized truss configurations are subjected to the applied load, and their response is evaluated for stress distributions, displacements, and other relevant structural properties.

Through the comparison and verification processes, this study provides valuable insights into the capabilities and limitations of the ISRES algorithm in solving complex truss sizing and layout-optimization problems with multiple constraints. The findings offer engineers and designers valuable guidance in selecting appropriate optimization methods for similar structural design tasks in real-world engineering applications. The convergence graph and constraints graph for allowable stress, as presented in Figures 6 and 7, offer a detailed visual representation of the optimization outcomes for the 15-bar planar truss structure under consideration.

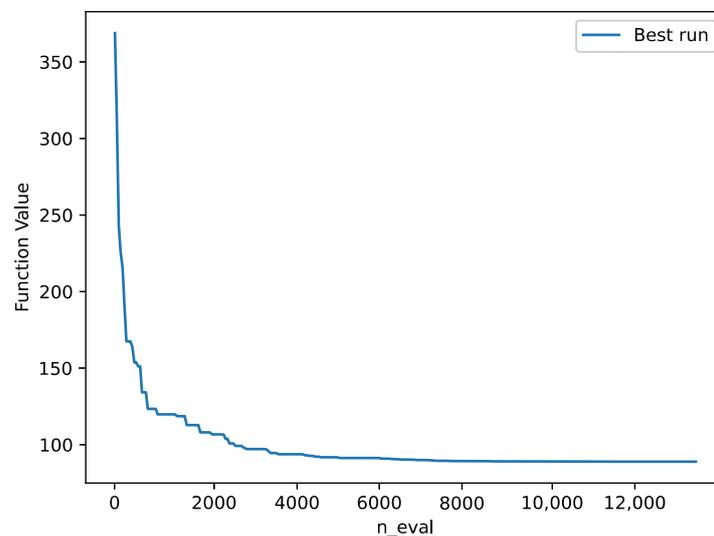


Figure 6. Convergence plot for 15-bar structure.

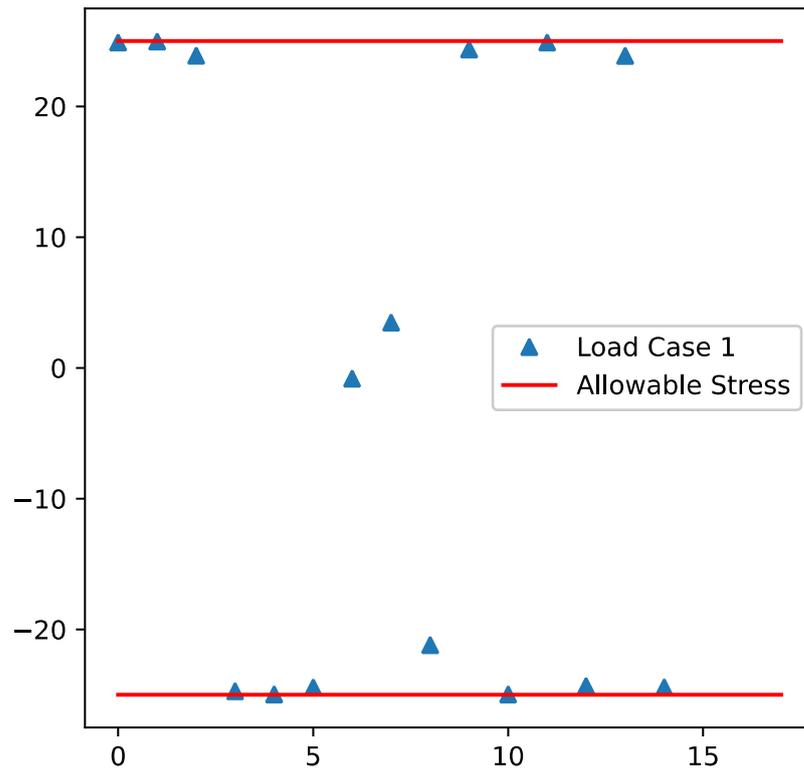


Figure 7. Stress constraint values with allowable limits for 15-bar structure.

4.3. 18-Bar Truss Structure

Figure 8 shows an 18-bar planar truss with five vertical concentrated loads which are directed downwards and applied to nodes 1, 2, 4, 6, and 8. The Young’s modulus of elasticity is 10,000 ksi, and the material density is 0.1 lb./in³. It is necessary to ensure that the maximum stress for all the members does not exceed ±25 ksi in both compression and tension. Additionally, the buckling strength of each member is calculated as $\frac{KEA_l}{L_l^2}$ for sizing optimization, and the bars are divided into four groups. All detailed information regarding this structure is given in Table 2. The ISRES algorithm is deployed once again to tackle the challenging sizing- and layout-optimization problems for this 18-bar truss structure. With appropriate parameter settings, the algorithm endeavors to minimize the truss weight while satisfying all the specified constraints.

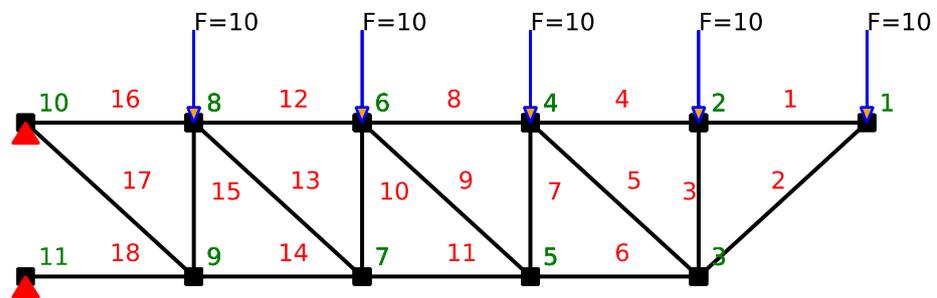


Figure 8. Schematic of the 18-bar structure.

As the ISRES algorithm generates multiple optimal solutions due to its stochastic nature, the best-performing solutions are selected based on the objective function value, which aims to minimize the truss weight. The selected solutions are then subject to further examination and comparison.

To assess the effectiveness of the ISRES algorithm, the obtained optimal truss designs are compared with solutions obtained from other well-established optimization methods. This comparative analysis includes various metaheuristic algorithms, gradient-based methods, and other state-of-the-art techniques. The comparison takes into account both the quality of the solutions and the computational efficiency of each method.

Table 2. Element grouping for 18-bar structure.

Size Variables
A1 = A4 = A8 = A12 = A16 A5 = A9 = A13 = A17 A2 = A6 = A10 = A14 = A18 A3 = A7 = A11 = A15
Possible Sizes
$A_i \in S = \{2, 2.25, 2.5, \dots, 21.75\}$
Possible Coordinates
775 in. $\leq x_3 \leq$ 1225 in.; 525 in. $\leq x_5 \leq$ 975 in.; 275 in. $\leq y_7 \leq$ 725 in.; 25 in. $\leq y_9 \leq$ 475 in.; −225 in. $\leq y_7, y_7, y_7 y_9 \leq$ 725 in
Young’s modulus $E = 100,000$ (ksi) Material density $\rho = 0.1$ (lb./in ³)

Additionally, the study conducts a thorough structural analysis on the optimized truss configurations to verify their compliance with the imposed constraints. Finite element analysis is performed to evaluate the stress distributions, member sizes, and overall stability of the optimized truss designs.

By comparing the results and verifying the solutions, this study provides valuable insights into the performance of the ISRES algorithm in handling complex truss optimization problems with multiple constraints. The findings contribute to a better understanding of the algorithm’s applicability and effectiveness in the field of structural engineering.

The convergence graph and constraints graph for allowable stress and buckling limit, depicted in Figures 9 and 10, offer a nuanced depiction of the ISRES algorithm’s prowess in addressing the intricate optimization challenges of the 18-bar planar truss structure.

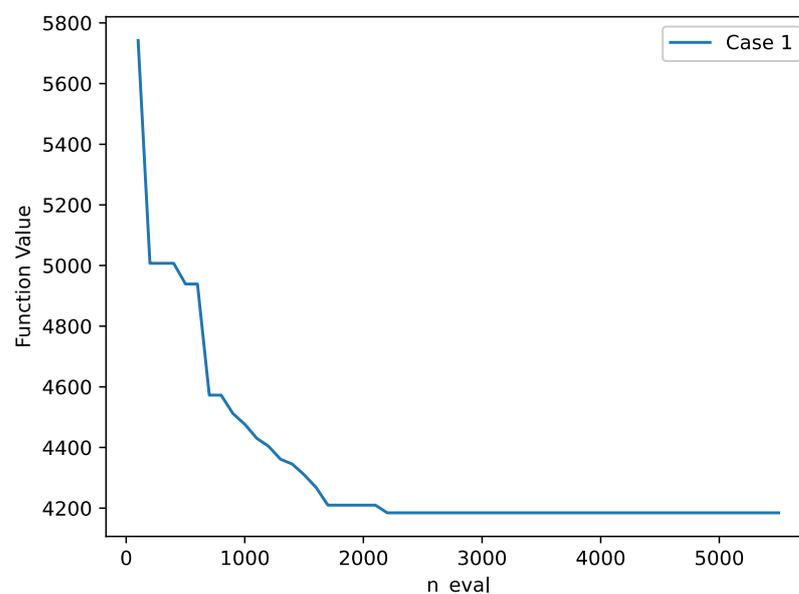


Figure 9. Convergence plot for 18-bar structure.

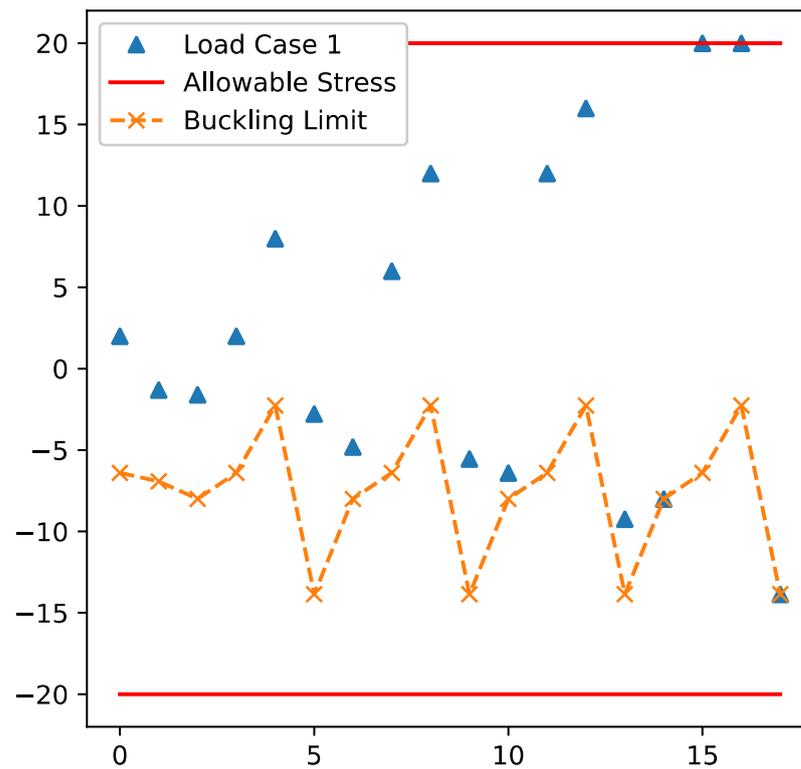


Figure 10. Stress constraint values with allowable limits for 18-bar structure.

4.4. 25-Bar Truss Structure

The 25-bar truss structure, displayed in Figure 11, is composed of a material with a mass density of 0.1 lb./in³ and an elastic modulus of 10,000 ksi. Structural members have been grouped into eight design variables due to the tower’s double symmetry along the x and y axes. Tables 3 and 4 outline the member groupings and their respective stress limitations. For consistency with other studies, the minimum and maximum cross-sectional values for all members were set to $A_{min} = 0.01 \text{ in}^2$ and $A_{max} = 3.4 \text{ in}^2$, respectively. The permissible nodal displacement values for all free nodes were restricted to ± 0.35 in in the x, y, and z directions. The benchmark has 124 non-linear design constraints (50 tensile/compressive stress and 12 displacement constraints) for each of the two loading conditions outlined in Table 4. Given the complexity of the problem, the ISRES algorithm is employed as the optimization tool of choice. With the careful selection of parameters and control settings, the algorithm undertakes the task of minimizing the truss weight while satisfying the extensive set of design constraints.

Table 3. Element grouping for 25-bar structure.

Element Group	Members	Comp. Stress (Kips)
A ₁	1	35.092
A ₂	2,3,4,5	11.590
A ₃	6,7,8,9	17.307
A ₄	10,11	35.092
A ₅	12,13	35.092
A ₆	14,15,16,17	6.759
A ₇	18,19,20,21	6.959
A ₈	22,23,24,25	11.802

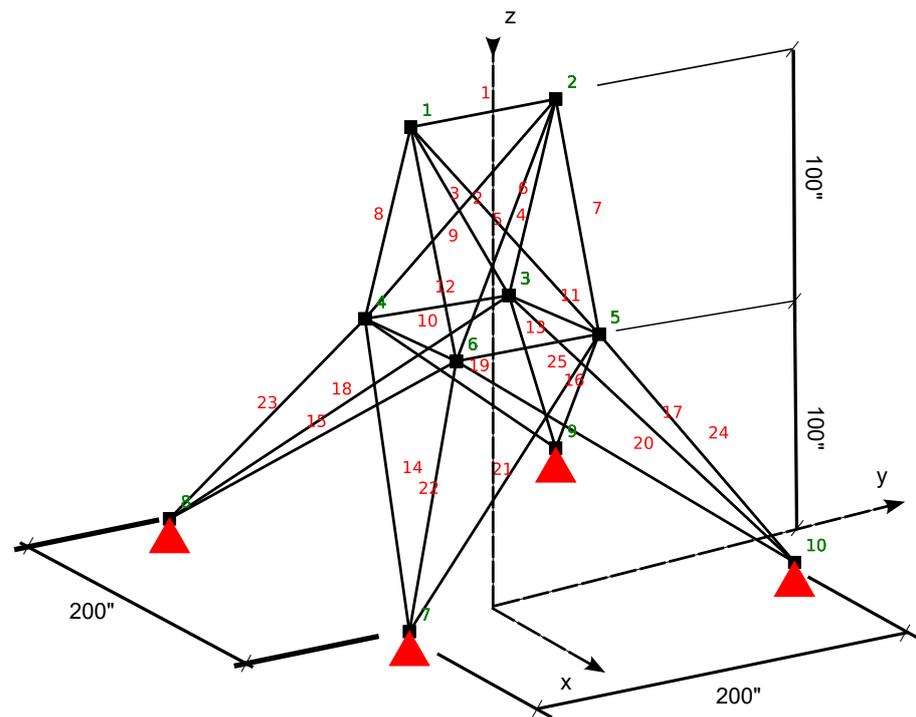


Figure 11. Schematic of the 25-bar structure.

Table 4. Loading conditions for the 25-bar truss structure.

Node	Condition I			Condition II		
	P_x	P_y	P_z	P_x	P_y	P_z
1	0	20	-5.0	1	10.0	-5.0
2	0	-20	-5.0	0	10.0	-5.0
3	0	0	0	0.5	0	0
6	0	0	0	0.5	0	0

Multiple runs of the ISRES algorithm are conducted to explore the solution space and identify diverse candidate solutions. The best-performing solutions are selected based on the objective function, which seeks to minimize the truss weight.

In addition to optimizing the truss sizing and layout, the ISRES results are compared with those obtained from various other optimization methods available in the literature. The comparative analysis encompasses a range of metaheuristic algorithms, gradient-based techniques, and other state-of-the-art approaches. The comparison considers the quality of the solutions and the computational efficiency of each method.

Furthermore, a comprehensive structural analysis is performed on the optimized truss designs to validate their compliance with the imposed constraints. A finite element analysis is utilized to assess stress distributions, member sizes, and overall structural stability for each of the loading conditions.

By conducting the comparison and verification, the study offers valuable insights into the performance of the ISRES algorithm in handling highly constrained truss-optimization problems. The findings contribute to a better understanding of the algorithm’s strengths and weaknesses, particularly in the context of complex structural designs.

Figure 12 provides a convergence plot of the ISRES algorithm applied to the 25-bar truss structure. The plot offers a graphical representation of the algorithm’s convergence behavior over the optimization iterations. The gradual reduction of the objective function values indicates the algorithm’s progress towards optimal solutions. Analyzing the plot

assists in understanding the convergence rate and identifying any convergence stagnation or rapid convergence trends, contributing to a deeper insight into the optimization process.

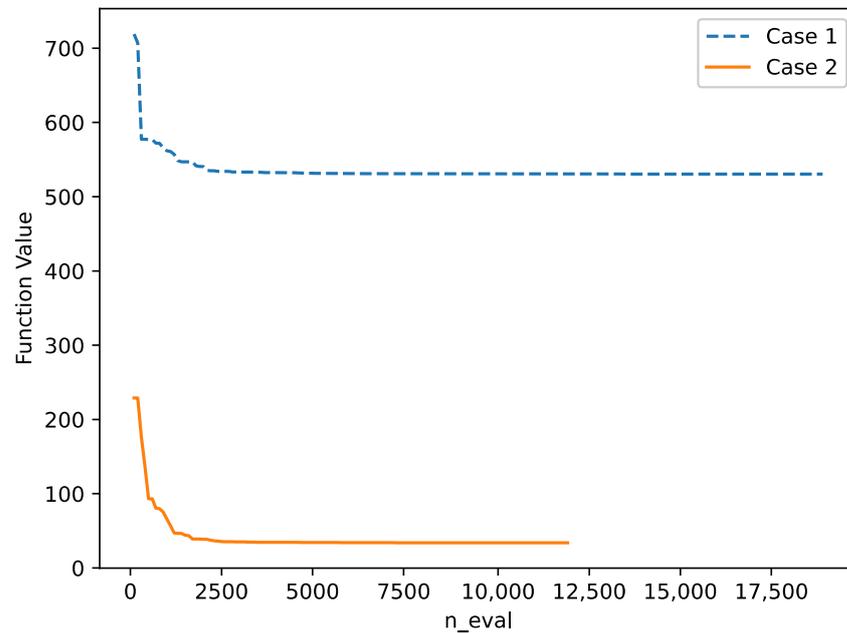


Figure 12. Convergence plot for 25-bar structure.

Figure 13 illustrates the displacement constraints along with their allowable limits for load case 1 and load case 2. The figure provides a visual representation of how the optimized truss designs conform to the specified nodal displacement limits. By presenting the displacement distributions across the truss structure, this figure offers insights into the structural stability of the optimized designs. It highlights areas where displacement constraints are met or exceeded, aiding in the identification of regions that may require further refinement to ensure the overall reliability of the truss configurations.

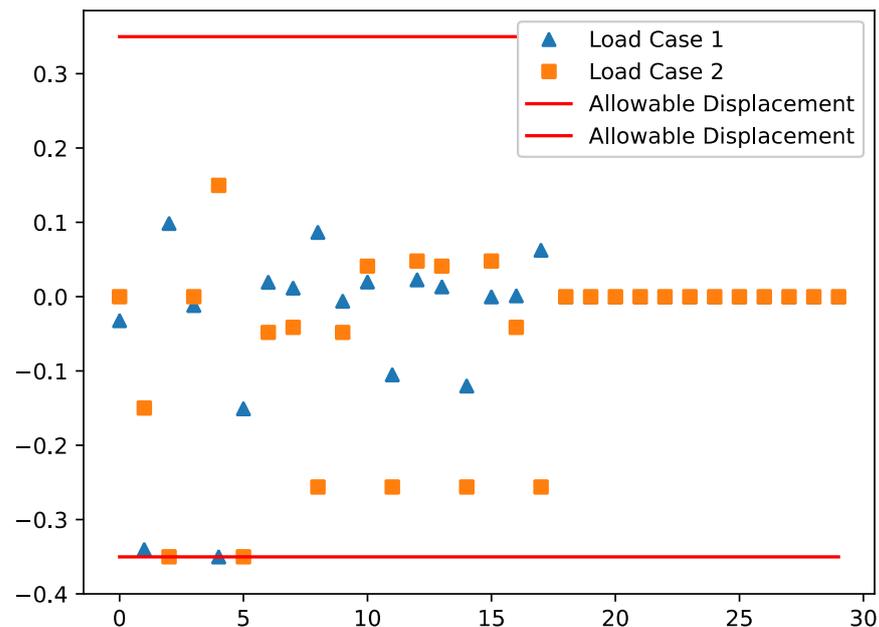


Figure 13. Displacement constraint values with allowable limits for 25-bar structure.

Figure 14, on the other hand, displays the stress constraints and their allowable limits for both load case 1 and load case 2. This figure visually represents how the optimized truss designs adhere to the stress limitations imposed on the members. By presenting stress distributions, the figure provides valuable insights into whether the members experience tensile or compressive stresses within acceptable bounds. This aids in the evaluation of the structural integrity and safety of the optimized truss configurations.

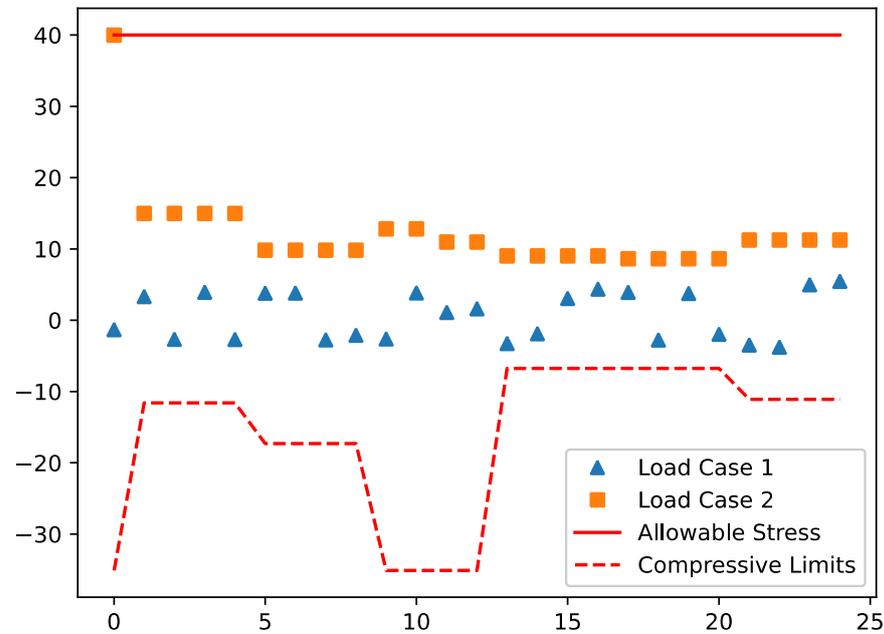


Figure 14. Stress constraint values with allowable limits for 25-bar structure.

4.5. 72-Bar Truss Structure

The schematic in Figure 15 shows 72-bar benchmark spatial truss structure. The structure consists of 16 independent design variables (as seen in Table 5) which are grouped together due to its structural symmetry. It is considered to be made out of material with a mass density of 0.1 lb./in³ and an elastic modulus of 10,000 ksi. Under both of the loading conditions, the truss tower has 320 non-linear design constraints (72 tension/compression and 16 displacement). The maximum design variable (Amax) is set to 4.0 in² and the minimum design variable (Amin) is set to 0.1 in² for case 1 and 0.01 in² for case 2. The displacement in the topmost nodes (17, 18, 19, and 20) are limited to ±0.25 in in the x and y direction, and the normal member stresses are limited to 25 ksi for both tension and compression, as given in Table 6. Multiple runs of the ISRES algorithm are conducted to generate diverse candidate solutions. The best-performing solutions are then selected based on the objective function, aiming to minimize the truss weight while satisfying all the imposed constraints.

Table 5. Element grouping for 72-bar structure.

Element Group	Members	Element Group	Members
A ₁	1,2,3,4	A ₉	37,38,39,40
A ₂	5,6,7,8,9,10,11,12	A ₁₀	41,42,43,44,45,46,47,48
A ₃	13,14,15,16	A ₁₁	49,50,51,52
A ₄	17,18	A ₁₂	53,54
A ₅	19,20,21,22	A ₁₃	55,56,57,58
A ₆	23,24,25,26,27,28,29,30	A ₁₄	59,60,61,62,63,64,65,66
A ₇	31,32,33,34	A ₁₅	67,68,69,70
A ₈	35,36	A ₁₆	71,72

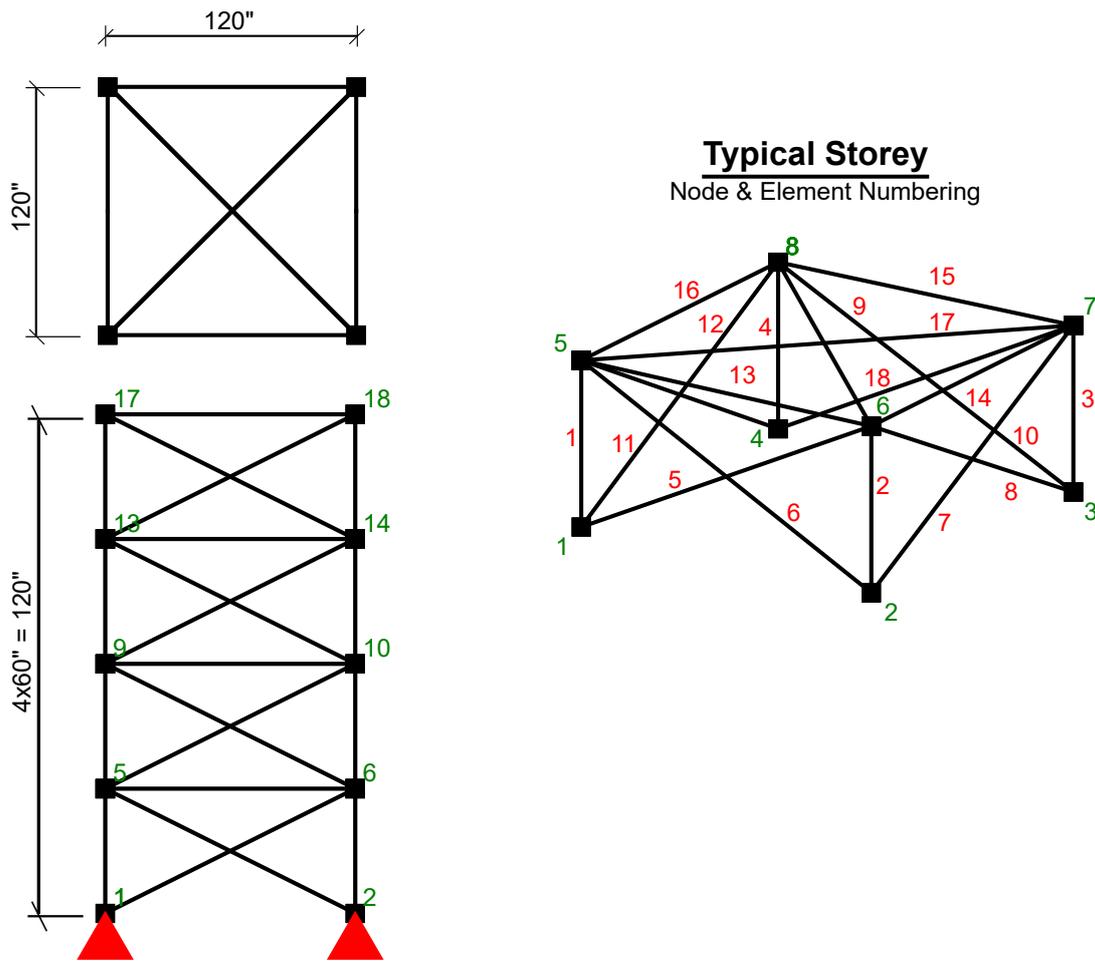


Figure 15. Schematic of the 72-bar structure.

Table 6. Loading conditions for the 72-bar truss structure.

Node	Condition I			Condition II		
	P _x	P _y	P _z	P _x	P _y	P _z
17	5.0	5.0	−5.0	0	0	−5.0
18	0	0	0	0	0	−5.0
19	0	0	0	0	0	−5.0
20	0	0	0	0	0	−5.0

For comprehensive evaluation, the results obtained from the ISRES algorithm are compared with solutions achieved through various other optimization methods reported in the literature. The comparison encompasses a wide range of metaheuristic algorithms, gradient-based techniques, and other state-of-the-art approaches. The comparison considers the quality of the solutions and the computational efficiency of each method.

Furthermore, the optimized truss designs undergo a rigorous finite element analysis to validate their structural integrity and stability. The structural analysis evaluates stress distributions, member sizes, and overall performance under both loading conditions.

By conducting the comparison and verification, the study contributes valuable insights into the effectiveness of the ISRES algorithm in handling complex spatial truss-optimization problems. The findings inform engineers and designers of the algorithm’s applicability and performance, particularly in challenging structural design scenarios.

In conclusion, the study successfully demonstrates the efficacy of the ISRES algorithm in optimizing the sizing and layout of the 72-bar benchmark spatial truss structure. The

comparison with other optimization methods and the structural analysis validation bolster the confidence in the optimized truss designs. These results have significant implications for engineering and architectural applications, guiding practitioners in selecting appropriate optimization strategies for complex spatial truss-design challenges. The research advances the field of optimization techniques in structural engineering, enabling the creation of efficient and reliable truss designs for a diverse range of real-world applications.

Figure 16 provides a comprehensive representation of the allowable displacements for every element in the 72-bar structure, along with the allowable limits for each element, for both load case 1 and load case 2. By illustrating these allowable displacement limits on a per-element basis, the figure offers a detailed view of how the optimized truss designs adhere to the specified displacement restrictions for varying loading scenarios. This insightful visualization aids in the assessment of structural stability and safety across the entire truss structure.

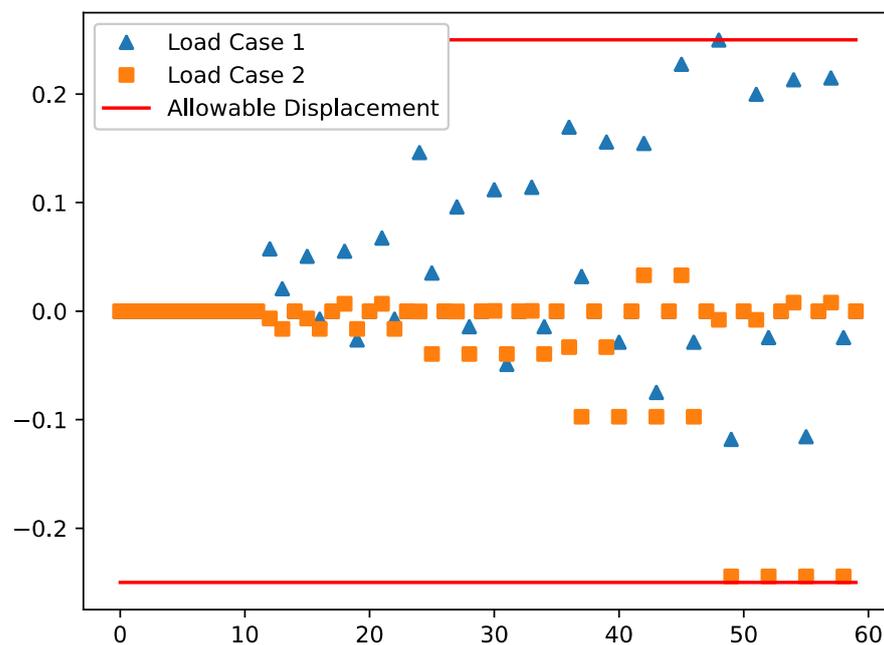


Figure 16. Displacement constraint values with allowable limits for 72-bar structure.

5. Results

In this paper, we have investigated the application of the Improved Stochastic Ranking Evolution Strategy (ISRES) algorithm for the sizing and layout optimization of truss benchmark structures. The truss benchmark problems, ranging from 10-bar to 72-bar structures, were subjected to various loading conditions and design constraints. The main objective of the optimization was to minimize the truss weight while satisfying stress and displacement constraints.

The results obtained from the ISRES algorithm demonstrated its effectiveness in finding optimal solutions for the truss sizing and layout problems. For each truss benchmark structure, the ISRES algorithm efficiently explored the solution space and converged to well-performing designs that minimized the truss weight while adhering to all the imposed constraints. The optimization process successfully handled the complexity of the problem, which included multiple design variables, stress constraints, and nodal displacement limits.

Through comparison with other optimization methods reported in the literature, the ISRES algorithm showcased competitive performance in terms of solution quality and computational efficiency. The algorithm's ability to find diverse candidate solutions and its robustness in dealing with highly constrained problems were particularly noteworthy.

The validation of the optimized truss designs through a finite element analysis further confirmed their structural integrity and stability. The designs were verified to meet stress

and displacement limits, ensuring the safety and reliability of the final truss configurations. Optimized results for all structures are given in Tables 7–13 for all cases.

Table 7. Optimized results for the 10-bar planar truss (case 1).

Design Variable (in ²)	Li et al. [49]	Wu and Chow [5]	Camp et al. [50]	Awad [21]	Sadollah et al. [51]	This Study
A ₁	30.569	26.50	28.92	30.501	30.00	30.6
A ₂	0.1	1.62	0.10	0.1	1.62	0.1
A ₃	22.974	16.00	24.07	23.198	22.90	22.8
A ₄	15.148	14.20	13.96	15.247	16.90	15.4
A ₅	0.1	1.80	0.1	0.1	1.62	0.1
A ₆	0.547	1.62	0.56	0.55511	1.62	0.6
A ₇	7.493	5.12	7.69	7.4562	7.97	7.5
A ₈	21.159	16.00	21.95	21.035	22.90	21.1
A ₉	21.556	18.50	22.09	21.526	22.90	21.5
A ₁₀	0.1	2.38	0.1	0.1	1.62	0.1
Weight (lb)	5061.03	4376.20	5076.31	5060.85	5507.75	5072.528

Table 8. Optimized results for the 10-bar planar truss (case 2).

Design Variable (in ²)	Kaveh et al. [49]	Wu and Chow [5]	Lee and Geem [52]	Awad et al. [21]	Sadollah et al. [51]	This Study
A ₁	23.299	30.50	23.25	23.62	29.50	23.5
A ₂	0.1	0.50	0.102	0.1	0.10	0.1
A ₃	25.682	16.50	25.73	25.434	24.0	25.0
A ₄	14.510	15.00	14.51	14.351	15.0	14.5
A ₅	0.1	0.10	0.100	0.10003	0.10	0.1
A ₆	1.969	0.10	1.997	1.9701	0.50	2.0
A ₇	12.149	0.50	12.21	12.339	7.50	12.5
A ₈	12.360	18.00	12.61	12.712	21.50	13.0
A ₉	20.869	19.50	20.36	20.346	21.50	20.2
A ₁₀	0.1	0.50	0.1	0.1	0.10	0.1
Weight (lb)	4679.15	42.17.30	4668.8	4677.06	5067.33	4687.9

Table 9. Comparison of optimal designs for the 15-bar planar truss structure.

Design Variable (in ²)	Rahami et al. [1]	Gholizadeh [2]	Dede and Ayvaz [3]	Tang et al. [4]	Hwang and He [53]	This Study
A ₁	1.081	1.174	1.081	1.081	0.954	0.954
A ₂	0.539	0.539	0.954	0.539	1.081	0.539
A ₃	0.287	0.347	0.141	0.287	0.440	0.111
A ₄	0.954	0.954	1.081	0.954	1.174	0.954
A ₅	0.539	0.954	0.539	0.954	1.488	0.539
A ₆	0.141	0.141	0.347	0.22	0.270	0.347
A ₇	0.111	0.141	0.111	0.111	0.270	0.111
A ₈	0.111	0.111	0.174	0.111	0.347	0.111
A ₉	0.539	1.174	0.141	0.287	0.220	0.174
A ₁₀	0.440	0.141	0.270	0.220	0.440	0.44
A ₁₁	0.539	0.440	0.220	0.440	0.220	0.44
A ₁₂	0.270	0.440	0.141	0.440	0.440	0.174
A ₁₃	0.220	0.141	0.440	0.111	0.347	0.174
A ₁₄	0.141	0.141	0.347	0.220	0.270	0.347
A ₁₅	0.287	0.347	0.141	0.347	0.220	0.111
X ₂	101.5775	102.2873	100.0042	133.612	118.346	101.2513
Y ₂	227.9112	240.5050	241.0473	234.52	225.209	240.7576
Y ₂	134.7956	112.5840	118.8228	100.449	119.046	132.8862
Y ₃	128.2206	108.0428	100.0829	104.738	105.086	129.3964
Y ₄	54.8630	57.7952	50.0000	73.762	63.375	55.1655
Y ₆	−16.4484	6.4299	3.1411	−10.067	−20.0	−19.5015
Y ₇	−16.4484	1.8006	−9.6997	−1.339	−20.0	10.1465
Y ₈	54.8572	57.7987	46.8963	50.402	57.722	52.1898
Weight (lb)	76.6854	77.6153	76.6519	79.820	104.573	71.86

Table 10. Optimized results for the 18-bar planar truss.

Design Variable (in ²)	Imai and Schmit [54]	Rahami et al. [1]	Mortazavi et al. [55]	Sonmez [56]	Awad [21]	This Study
A ₁	9.998	12.75	14.25	10.000	10.000	10.000
A ₂	21.65	18.5	11.75	21.651	21.651	21.651
A ₃	12.50	4.75	6.00	12.500	12.500	12.500
A ₄	7.072	3.25	8.00	7.071	7.071	7.071
Weight (lb)	6430.0	4530.7	4520.99	6430.529	6430.529	6430.529

Table 11. Comparison of optimal designs for the 25-bar spatial truss structure.

Design Variable (in ²)	Li et al. [28]	Lee and Geem [52]	Rajeev and Krishnamoorthy [57]	Gandomi et al. [58]	Camp and Bichon [59]	This Study
A ₁	9.863	0.047	0.1	0.010	0.10	1.309
A ₂	1.798	2.022	1.8	2.054	0.30	2.806
A ₃	3.654	2.950	2.3	3.008	3.40	2.291
A ₄	0.100	0.010	0.2	0.010	0.10	0.951
A ₅	0.100	0.014	0.1	0.010	2.10	0.010
A ₆	0.596	0.688	0.8	0.679	1.00	0.107
A ₇	1.659	1.657	1.8	1.611	0.50	1.004
A ₈	2.612	2.663	3.0	2.678	3.40	3.398
Weight (lb)	627.08	544.38	546.01	544.99	484.85	530.676

Table 12. Optimized results for the 72-bar planar truss (case 1).

Design Variable (in ²)	Wu and Chow [5]	Li et al. [28]	Sadollah et al. [51]	Degertekin and Hayalioglu [60]	Awad et al. [21]	This Study
A ₁	1.5	2.1	1.9	1.8929	1.8812	1.963
A ₂	0.7	0.6	0.5	0.5160	0.52207	0.481
A ₃	0.1	0.1	0.1	0.0100	0.01003	0.010
A ₄	0.1	0.1	0.1	0.0100	0.010024	0.011
A ₅	1.3	1.4	1.3	1.2917	1.3045	1.233
A ₆	0.5	0.5	0.5	0.5176	0.51551	0.506
A ₇	0.2	0.1	0.1	0.0100	0.010011	0.011
A ₈	0.1	0.1	0.1	0.0100	0.010067	0.01
A ₉	0.5	0.5	0.6	0.5229	0.52429	0.538
A ₁₀	0.5	0.5	0.5	0.5193	0.51519	0.533
A ₁₁	0.1	0.1	0.1	0.0100	0.011122	0.010
A ₁₂	0.2	0.1	0.1	0.0997	0.10444	0.167
A ₁₃	0.2	0.2	0.2	0.1680	0.16738	0.161
A ₁₄	0.5	0.5	0.6	0.5359	0.53737	0.542
A ₁₅	0.5	0.3	0.4	0.4457	0.44244	0.478
A ₁₆	0.7	0.7	0.6	0.5818	0.57549	0.551
Weight (lb)	400.66	388.94	385.54	363.841	363.88	364.379

Table 13. Optimized results for the 72-bar planar truss (case 2).

Design Variable (in ²)	Wu and Chow [5]	Li et al. [28]	Sadollah et al. [51]	Bekdaş et al. [61]	Awad et al. [21]	This Study
A ₁	0.196	4.97	0.196	1.8899	1.8607	1.157
A ₂	0.602	1.228	0.563	0.5119	0.50626	0.610
A ₃	0.307	0.111	0.442	0.1000	0.1	0.01
A ₄	0.766	0.111	0.602	0.1000	0.1	0.01
A ₅	0.391	2.88	0.442	1.2702	1.2749	1.068
A ₆	0.391	1.457	0.442	0.5120	0.50952	0.599
A ₇	0.141	0.141	0.111	0.1000	0.1	0.01
A ₈	0.111	0.111	0.111	0.1000	0.10012	0.082
A ₉	1.800	1.563	1.266	0.5234	0.52872	0.369
A ₁₀	0.602	1.228	0.563	0.5165	0.52359	0.659
A ₁₁	0.141	0.111	0.111	0.1000	0.10014	0.01
A ₁₂	0.307	0.196	0.111	0.1000	0.10001	0.077
A ₁₃	1.563	0.391	1.800	0.1565	0.15628	0.1
A ₁₄	0.766	1.457	0.602	0.5457	0.54998	0.682
A ₁₅	0.141	0.766	0.111	0.4104	0.40132	0.467
A ₁₆	0.111	1.563	0.111	0.5678	0.58582	0.342
Weight (lb)	427.203	933.09	390.73	379.6172	379.68	379.98

6. Conclusions

In conclusion, this study demonstrates the efficacy of the Improved Stochastic Ranking Evolution Strategy (ISRES) algorithm for the sizing and layout optimization of truss benchmark structures. The results showcase the algorithm's ability to efficiently converge to optimal solutions, effectively minimizing truss weight while satisfying complex design constraints.

The ISRES algorithm proves to be a valuable optimization tool for structural engineers and designers, as it provides practical and reliable solutions for a wide range of truss-design problems. Its ability to handle multiple design variables, stress constraints, and nodal displacement limits make it a versatile and powerful tool for complex engineering optimization tasks.

Furthermore, the comparison with other optimization methods and the validation through a finite element analysis highlight the ISRES algorithm's competitiveness and robustness in tackling complex structural design challenges. The algorithm's performance indicates its potential for broader applications in various engineering and architectural contexts.

Overall, this research contributes to the advancement of optimization techniques in structural engineering, enabling the development of more efficient and cost-effective truss designs. The findings offer valuable guidance to practitioners, empowering them to make informed decisions when selecting appropriate optimization strategies for complex truss-design scenarios.

As future work, it may be beneficial to extend this study to include additional structural benchmarks and explore the application of the ISRES algorithm in other engineering optimization problems. Additionally, investigating the algorithm's performance on real-world truss-design projects and further fine-tuning its parameters for specific applications could yield even more promising results.

In summary, the ISRES algorithm proves to be a reliable and efficient tool for the sizing and layout optimization of truss benchmark structures. This study highlights its potential for enhancing engineering design processes, promoting the development of more sustainable and optimized truss solutions for practical applications.

Author Contributions: Conceptualization, M.S.A. and A.N.; methodology, A.N.; software, M.S.A.; validation, A.N., D.Y., E.E. and M.S.A.; writing—original draft preparation, A.N., D.Y., E.E. and M.S.A.; writing—review and editing, A.N., D.Y., E.E. and M.S.A.; visualization, D.Y. and M.S.A.; supervision, A.N. and E.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to ongoing studies.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kumar, S.; Tejani, G.G.; Pholdee, N.; Bureerat, S. Multi-Objective Passing Vehicle Search algorithm for structure optimization. *Expert Syst. Appl.* **2021**, *169*, 114511. [[CrossRef](#)]
2. Gholizadeh, S. Layout optimization of truss structures by hybridizing cellular automata and particle swarm optimization. *Comput. Struct.* **2013**, *125*, 86–99. [[CrossRef](#)]
3. Dede, T.; Ayvaz, Y. Combined size and shape optimization of structures with a new meta-heuristic algorithm. *Appl. Soft Comput. J.* **2015**, *28*, 250–258. [[CrossRef](#)]
4. Tang, W.; Tong, L.; Gu, Y. Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables. *Int. J. Numer. Methods Eng.* **2005**, *62*, 1737–1762. [[CrossRef](#)]
5. Wu, S.J.; Chow, P.T. Integrated discrete and configuration optimization of trusses using genetic algorithms. *Comput. Struct.* **1995**, *55*, 695–702. [[CrossRef](#)]
6. Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison-Wesley Professional: Boston, MA, USA, 1989.
7. Yang, X.S.; Deb, S. Cuckoo search via Lévy flights. In Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009; pp. 210–214. [[CrossRef](#)]

8. Erol, O.K.; Eksin, I. A new optimization method: Big Bang-Big Crunch. *Adv. Eng. Softw.* **2006**, *37*, 106–111. [[CrossRef](#)]
9. Kaveh, A.; Kaveh, A. Particle Swarm Optimization. In *Advances in Metaheuristic Algorithms for Optimal Design of Structures*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 1942–1948. [[CrossRef](#)]
10. Yang, X.-S. *Nature Inspired Metaheuristic Algorithms*; Luniver Press: Frome, UK, 2010.
11. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A New Heuristic Optimization Algorithm: Harmony Search. *Simulation* **2001**, *76*, 60–68. [[CrossRef](#)]
12. Ha, T.V.; Nguyen, Q.H.; Nguyen, T.T. A parallel differential evolution with cooperative multi-search strategy for sizing truss optimization. *Appl. Soft Comput.* **2022**, *131*, 109762. [[CrossRef](#)]
13. Vu-Huu, T.; Pham-Van, S.; Pham, Q.; Cuong-Le, T. An improved bat algorithms for optimization design of truss structures. *Structures* **2023**, *47*, 2240–2258. [[CrossRef](#)]
14. Miguel, L.F.F.; Lopez, R.H.; Miguel, L.F.F. Multimodal size, shape, and topology optimisation of truss structures using the Firefly algorithm. *Adv. Eng. Softw.* **2013**, *56*, 23–37. [[CrossRef](#)]
15. Khodadadi, N.; Çiftçioğlu, A.Ö.; Mirjalili, S.; Nanni, A. A comparison performance analysis of eight meta-heuristic algorithms for optimal design of truss structures with static constraints. *Decis. Anal. J.* **2023**, *8*, 100266. [[CrossRef](#)]
16. Assimi, H.; Jamali, A. A hybrid algorithm coupling genetic programming and Nelder–Mead for topology and size optimization of trusses with static and dynamic constraints. *Expert Syst. Appl.* **2018**, *95*, 127–141. [[CrossRef](#)]
17. Jafari, M.; Salajegheh, E.; Salajegheh, J. Optimal design of truss structures using a hybrid method based on particle swarm optimizer and cultural algorithm. *Structures* **2021**, *32*, 391–405. [[CrossRef](#)]
18. Degertekin, S.O.; Bayar, G.Y.; Lamberti, L. Parameter free Jaya algorithm for truss sizing-layout optimization under natural frequency constraints. *Comput. Struct.* **2021**, *245*, 106461. [[CrossRef](#)]
19. Azizi, M.; Aickelin, U.; Khorshidi, H.A.; Shishehgharkhaneh, M.B. Shape and size optimization of truss structures by Chaos game optimization considering frequency constraints. *J. Adv. Res.* **2022**, *41*, 89–100. [[CrossRef](#)] [[PubMed](#)]
20. Jawad, F.K.J.; Ozturk, C.; Dansheng, W.; Mahmood, M.; Al-Azzawi, O.; Al-Jemely, A. Sizing and layout optimization of truss structures with artificial bee colony algorithm. *Structures* **2021**, *30*, 546–559. [[CrossRef](#)]
21. Awad, R. Sizing optimization of truss structures using the political optimizer (PO) algorithm. *Structures* **2021**, *33*, 4871–4894. [[CrossRef](#)]
22. Dehghani, M.; Mashayekhi, M.; Sharifi, M. An efficient imperialist competitive algorithm with likelihood assimilation for topology, shape and sizing optimization of truss structures. *Appl. Math. Model.* **2021**, *93*, 1–27. [[CrossRef](#)]
23. Pierezan, J.; Coelho, L.D.S.; Mariani, V.C.; de Vasconcelos Segundo, E.H.; Prayogo, D. Chaotic coyote algorithm applied to truss optimization problems. *Comput. Struct.* **2021**, *242*, 106353. [[CrossRef](#)]
24. Pholdee, N.; Bureerat, S. Comparative performance of meta-heuristic algorithms for mass minimisation of trusses with dynamic constraints. *Adv. Eng. Softw.* **2014**, *75*, 1–13. [[CrossRef](#)]
25. Serpik, I. Discrete size and shape optimization of truss structures based on job search inspired strategy and genetic operations. *Period. Polytech. Civ. Eng.* **2020**, *64*, 801–814. [[CrossRef](#)]
26. Cao, H.; Qian, X.; Chen, Z.; Zhu, H. Enhanced particle swarm optimization for size and shape optimization of truss structures. *Eng. Optim.* **2017**, *49*, 1939–1956. [[CrossRef](#)]
27. Luh, G.C.; Lin, C.Y. Optimal design of truss-structures using particle swarm optimization. *Comput. Struct.* **2011**, *89*, 2221–2232. [[CrossRef](#)]
28. Li, L.J.; Huang, Z.B.; Liu, F. A heuristic particle swarm optimization method for truss structures with discrete variables. *Comput. Struct.* **2009**, *87*, 435–443. [[CrossRef](#)]
29. Gomes, H.M. Truss optimization with dynamic constraints using a particle swarm algorithm. *Expert Syst. Appl.* **2011**, *38*, 957–968. [[CrossRef](#)]
30. Kim, T.H.; Byun, J.I. Truss sizing optimization with a diversity-enhanced cyclic neighborhood network topology particle swarm optimizer. *Mathematics* **2020**, *8*, 1087. [[CrossRef](#)]
31. Carvalho, J.P.G.; Carvalho, É.C.R.; Vargas, D.E.C.; Hallak, P.H.; Lima, B.S.L.P.; Lemonge, A.C.C. Multi-objective optimum design of truss structures using differential evolution algorithms. *Comput. Struct.* **2021**, *252*, 106544. [[CrossRef](#)]
32. Dang, K.D.; Nguyen-Van, S.; Thai, S.; Lee, S.; Luong, V.H.; Lieu, Q.X. A single step optimization method for topology, size and shape of trusses using hybrid differential evolution and symbiotic organisms search. *Comput. Struct.* **2022**, *270*, 106846. [[CrossRef](#)]
33. Nguyen-Van, S.; Nguyen, K.T.; Luong, V.H.; Lee, S.; Lieu, Q.X. A novel hybrid differential evolution and symbiotic organisms search algorithm for size and shape optimization of truss structures under multiple frequency constraints. *Expert Syst. Appl.* **2021**, *184*, 115534. [[CrossRef](#)]
34. Nguyen-Van, S.; Nguyen, K.T.; Dang, K.D.; Nguyen, N.T.T.; Lee, S.; Lieu, Q.X. An evolutionary symbiotic organisms search for multiconstraint truss optimization under free vibration and transient behavior. *Adv. Eng. Softw.* **2021**, *160*, 103045. [[CrossRef](#)]
35. Tang, H.; Huynh, T.N.; Lee, J. A novel adaptive 3-stage hybrid teaching-based differential evolution algorithm for frequency-constrained truss designs. *Structures* **2022**, *38*, 934–948. [[CrossRef](#)]
36. Kao, C.Y.; Hung, S.L.; Setiawan, B. Two Strategies to Improve the Differential Evolution Algorithm for Optimizing Design of Truss Structures. *Adv. Civ. Eng.* **2020**, *2020*, 8741862. [[CrossRef](#)]

37. Ho-Huu, V.; Nguyen-Thoi, T.; Truong-Khac, T.; Le-Anh, L.; Vo-Duy, T. An improved differential evolution based on roulette wheel selection for shape and size optimization of truss structures with frequency constraints. *Neural Comput. Appl.* **2018**, *29*, 167–185. [[CrossRef](#)]
38. Panagant, N.; Bureerat, S. Truss topology, shape and sizing optimization by fully stressed design based on hybrid grey wolf optimization and adaptive differential evolution. *Eng. Optim.* **2018**, *50*, 1645–1661. [[CrossRef](#)]
39. Assimi, H.; Jamali, A.; Nariman-Zadeh, N. Sizing and topology optimization of truss structures using genetic programming. *Swarm Evol. Comput.* **2017**, *37*, 90–103. [[CrossRef](#)]
40. Liu, J.; Xia, Y. A hybrid intelligent genetic algorithm for truss optimization based on deep neural network. *Swarm Evol. Comput.* **2022**, *73*, 101120. [[CrossRef](#)]
41. Noii, N.; Aghayan, I.; Hajirasouliha, I.; Kunt, M.M. A new hybrid method for size and topology optimization of truss structures using modified ALGA and QPGA. *J. Civ. Eng. Manag.* **2017**, *23*, 252–262. [[CrossRef](#)]
42. Dong, T.; Chen, S.; Huang, H.; Han, C.; Dai, Z.; Yang, Z. Large-Scale Truss Topology and Sizing Optimization by an Improved Genetic Algorithm with Multipoint Approximation. *Appl. Sci.* **2022**, *12*, 407. [[CrossRef](#)]
43. Delyová, I.S.; Frankovský, P.; Bocko, J.; Trebuňa, P.; Živčák, J.; Schürger, B.; Janigová, S. Sizing and Topology Optimization of Trusses Using Genetic Algorithm. *Materials* **2021**, *14*, 125–132. [[CrossRef](#)] [[PubMed](#)]
44. Chen, S.Y.; Shui, X.F.; Huang, H. Improved genetic algorithm with two-level approximation using shape sensitivities for truss layout optimization. *Struct. Multidiscip. Optim.* **2017**, *55*, 1365–1382. [[CrossRef](#)]
45. Zhou, P.; Du, J.; Lü, Z. Interval analysis based robust truss optimization with continuous and discrete variables using mix-coded genetic algorithm. *Struct. Multidiscip. Optim.* **2017**, *56*, 353–370. [[CrossRef](#)]
46. Assimi, H.; Jamali, A.; Nariman-Zadeh, N. Multi-objective sizing and topology optimization of truss structures using genetic programming based on a new adaptive mutant operator. *Neural Comput. Appl.* **2019**, *31*, 5729–5749. [[CrossRef](#)]
47. Runarsson, T.P.; Yao, X. Stochastic ranking for constrained evolutionary optimization. *IEEE Trans. Evol. Comput.* **2000**, *4*, 284–294. [[CrossRef](#)]
48. Runarsson, T.P.; Yao, X. Search biases in constrained evolutionary optimization. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2005**, *35*, 233–243. [[CrossRef](#)]
49. Li, L.J.; Huang, Z.B.; Liu, F.; Wu, Q.H. A heuristic particle swarm optimizer for optimization of pin connected structures. *Comput. Struct.* **2007**, *85*, 340–349. [[CrossRef](#)]
50. Camp, C.; Pezeshk, S.; Cao, G. Optimized Design of Two-Dimensional Structures Using a Genetic Algorithm. *J. Struct. Eng.* **1998**, *124*, 551–559. [[CrossRef](#)]
51. Sadollah, A.; Bahreininejad, A.; Eskandar, H.; Hamdi, M. Mine blast algorithm for optimization of truss structures with discrete variables. *Comput. Struct.* **2012**, *102–103*, 49–63. [[CrossRef](#)]
52. Lee, K.S.; Geem, Z.W. A new structural optimization method based on the harmony search algorithm. *Comput. Struct.* **2004**, *82*, 781–798. [[CrossRef](#)]
53. Hwang, S.F.; He, R.S. A hybrid real-parameter genetic algorithm for function optimization. *Adv. Eng. Inform.* **2006**, *20*, 7–21. [[CrossRef](#)]
54. Bojczuk, D.; Mróz, Z. Configuration Optimization of Trusses. *J. Struct. Div.* **1981**, *107*, 745–756. [[CrossRef](#)]
55. Mortazavi, A.; Toğan, V.; Nuhoglu, A. Weight minimization of truss structures with sizing and layout variables using integrated particle swarm optimizer. *J. Civ. Eng. Manag.* **2017**, *23*, 985–1001. [[CrossRef](#)]
56. Sonmez, M. Artificial Bee Colony algorithm for optimization of truss structures. *Appl. Soft Comput. J.* **2011**, *11*, 2406–2418. [[CrossRef](#)]
57. Rajeev, S.; Krishnamoorthy, C.S. Discrete optimization of structures using genetic algorithms. *J. Struct. Eng.* **1992**, *118*, 1233–1250. [[CrossRef](#)]
58. Gandomi, A.H.; Talatahari, S.; Yang, X.-S.; Deb, S. Design optimization of truss structures using cuckoo search algorithm. *Struct. Des. Tall Spec. Build.* **2013**, *22*, 1330–1349. [[CrossRef](#)]
59. Camp, C.V.; Bichon, B.J. Design of Space Trusses Using Ant Colony Optimization. *J. Struct. Eng.* **2004**, *130*, 741–751. [[CrossRef](#)]
60. Degertekin, S.O.; Hayalioglu, M.S. Sizing truss structures using teaching-learning-based optimization. *Comput. Struct.* **2013**, *119*, 177–188. [[CrossRef](#)]
61. Bekdaş, G.; Yücel, M.; NiGdeli, S.M. Evaluation of Metaheuristic-Based methods for optimization of truss structures via various algorithms and Lévy flight modification. *Buildings* **2021**, *11*, 49. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.