Available online at www.eccomasproceedia.org
Eccomas Proceedia COMPDYN (2025) 5548-5561

ECCOMAS
Proceedia

COMPDYN 2025
10th ECCOMAS Thematic Conference on
Computational Methods in Structural Dynamics and Earthquake Engineering
M. Papadrakakis, M. Fragiadakis (eds.)
Rhodes Island, Greece, 15-18 June 2025

COMPARATIVE ANALYSIS OF THREE METAHEURISTIC ALGORITHMS FOR TRUSS SIZE AND TOPOLOGY OPTIMIZATION

Sofia Noreña-Salazar[1], Luis A. Lara-Valencia[2], and Leandro Zapata-Longas[2]

[1] Universidad Nacional de Colombia
Carrera 80 N.°65-223, Núcleo Robledo. Medellín. Colombia
snorena@unal.edu.co

[2] Universidad Nacional de Colombia
Carrera 80 N.°65-223, Núcleo Robledo. Medellín. Colombia
{lualarava, lzapatalo}@unal.edu.co

Abstract

The efficient design of structures is essential to reduce material costs, enhancing structural performance, and meeting the rigorous demands of fields like structural engineering. This paper presents the size and topology optimization of two benchmark truss structures that have been extensively discussed in the literature. The central aim of this study is to reduce the structural weight of these trusses while satisfying a set of important constraints. Specifically, the optimization process considers factors such as strength requirements, displacement limitations, and kinematic stability conditions. To achieve this, matrix analysis techniques were implemented to deal with more practical considerations in the modeling process. The optimization was performed using three metaheuristic algorithms: Teaching-Learning-Based Optimization (TLBO), Stochastic Paint Optimizer (SPO), and the Flow Direction Algorithm (FDA). The computational performance of TLBO, SPO, and FDA is evaluated on benchmark truss structures. Aspects such as accuracy, convergence rate, and computational efficiency are analyzed. Each algorithm is shown to be effective in solving these optimization problems, highlighting distinct advantages over each other.

Keywords: Truss optimization, Teaching Learning Based Optimization, Flow Direction Algorithm, Stochastic Paint Optimization, weight minimization.

ISSN:2623-3347
Peer-review under responsibility of the organizing committee of COMPDYN 2025.
doi: 10.7712/120125.128 2.24562

1 INTRODUCTION

In recent decades, the pursuit of an efficient truss structure design has been a fundamental objective in civil engineering, particularly in applications where achieving lighter structures without compromising performance is essential. The need to develop structurally reliable, sustainable, and cost-effective trusses has driven the early development of various optimization methodologies, which have been widely applied by numerous authors. [1].

As a result, truss optimization can be carried out by adjusting their size, layout or topology. Size optimization focuses on selecting the most suitable cross-sectional areas for the structural members. Layout optimization aims to determine the optimal nodal coordinates of the joints to achieve the best structural shape. Topology optimization seeks the most efficient structural configuration by removing unnecessary members [2]. Since the primary goal is to develop a lighter structure, the objective function to be evaluated is the weight of the system. Since all of these methodologies align with this objective, they can be effectively combined to address more complex optimization challenges. [3].

To approach this problem, optimization algorithms are classified into deterministic and stochastic categories. While deterministic methods provide exact solutions, their applicability is often limited in large-scale cases. In contrast, stochastic algorithms, particularly metaheuristics, efficiently explore the solution space, making them well-suited for complex optimization problems [4]-[5]. This study evaluates three prominent metaheuristics: Teaching-Learning-Based Optimization (TLBO), Stochastic Paint Optimizer (SPO) and Flow Direction Algorithm (FDA).

Building on these concepts, the present investigation explores the size and topology optimization of truss structures using these algorithms. Their performance is evaluated through a series of numerical experiments on two benchmark problems: a 10-bar, and a 25-bar truss. The analysis examines their effectiveness in reducing structural weight while satisfying strength, displacement, and kinematic stability constraints. The results aim to provide a comparative perspective on these algorithms, highlighting their advantages and potential limitations in structural optimization**.**

2 THEORETICAL FOUNDATION

2.1 Size optimization of trusses

Sizing optimization of trusses aims to minimize structural weight while ensuring compliance with stress and displacement constraints. When the truss geometry is fixed, the problem reduces selecting appropriate cross-sectional areas for each member to achieve minimal weight without compromising structural integrity. This goal, formally known as the objective function OF , can be mathematically expressed as:

$$W = \sum_{e=1}^{Nm} \rho_e L_e A_e \tag{1}$$

Subject to:

$$\begin{aligned} \sigma^L &\le \sigma_e \le \sigma^U \\ \delta^L &\le \delta_c \le \delta^U \\ A^L &\le A_e \le A^U \end{aligned} \tag{2}$$

Where W is the truss weight, Nm is the number of elements, ρ_e, L_e, and A_e represent the mass density, length and cross-sectional area of each member, respectively. The cross-sectional area is the design variable and is constrained by predefined boundary limits A^L and A^U . Additionally, the design must satisfy stress σ_e and deflections δ_c limits: $\sigma^L \le \sigma_e \le \sigma^U$ and $\delta^L \le \delta_c \le \delta^U$. In cases where a truss solution exceeds these bounds, a penalty function ψ^k is applied, multiplying the original objective function to reflect the degree of constraint violation. The reformulated OF is presented below:

$$W = \left(\sum_{e=1}^{Nm} \rho_e L_e A_e \right) \psi^k \tag{3}$$

The penalty ψ^k for a truss k is the sum of stress and deflection penalties:

$$\psi^k = \left(1 + \phi_\sigma^k + \phi_\delta^k\right)^\zeta \tag{4}$$

Where ζ is a positive penalty exponent, which, based on other studies and for simplification, is set to 1 [6]. The total stress penalty ϕ_σ^k is calculated as the sum of the individual penalties for each member of the structure:

$$\phi_\sigma^k = \sum_{e=1}^{Nm} \phi_\sigma^e \tag{5}$$

For each member, the stress σ_e is compared to the maximum allowable limits $\sigma^{L,U}$. Penalty ϕ_σ^e is applied only if it exceeds the defined range; otherwise, is set to zero. This can be mathematically described as:

$$\text{If } \sigma^L \le \sigma_e \le \sigma^U \text{, then } \phi_\sigma^e = 0 \tag{6}$$

$$\text{If } \sigma_e < \sigma^L \text{ or } \sigma_e > \sigma^U \text{, then } \phi_\sigma^e = \left| \frac{\sigma_e - \sigma^{L,U}}{\sigma^{L,U}} \right|$$

The same procedure applies to deflection penalties ϕ_δ^k, considering that each node may experience deflections in x, y, z directions:

$$\phi_\delta^k = \sum_{c=1}^{Nc} \left[\phi_{\delta x}^c + \phi_{\delta y}^c + \phi_{\delta z}^c \right] \tag{7}$$

$$\text{If } \delta^L \le \delta_{c(x,y,z)} \le \delta^U \text{, then } \phi_{\delta(x,y,z)}^c = 0 \tag{8}$$

$$\text{If } \delta_{c(x,y,z)} < \delta^L \text{ or } \delta_{c(x,y,z)} > \delta^U \text{, then } \phi_{\delta(x,y,z)}^c = \left| \frac{\delta_{c(x,y,z)} - \delta^{L,U}}{\delta^{L,U}} \right|$$

2.2 Topology optimization of trusses

The objective function for topology optimization remains the same, with the cross-sectional area as the design variable. Consequently, this optimization approach addresses both size and topology simultaneously:

$$W = \sum_{e=1}^{Nm} \rho_e L_e A_e^{'} \tag{9}$$

Subject to:

Truss is kinetically stable and acceptable to the user

$$\sigma^L \leq \sigma_e \leq \sigma^U$$

$$\delta^L \leq \delta_c \leq \delta^U \tag{10}$$

$$A^L \leq A_e \leq A^U \text{ where } A_e^{'} = \begin{Bmatrix} A_e \rightarrow if \rightarrow A_e \geq \varepsilon \\ 0 \rightarrow if \rightarrow A_e < \varepsilon \end{Bmatrix}$$

In this case, the same constraints apply; however, there is a key distinction: since the goal is to remove elements, the cross-sectional areas can take two values: 0 or A_e. This is determined by a critical number named ε; if the cross-sectional area of a member falls below this value, it is automatically removed from the system. In this work, ε is assumed to be 0, which requires setting a negative lower limit for A^L to ensure a search space for element removal.

Another important consideration is that eliminating elements may also result in the removal of nodes, potentially affecting restricted or loaded nodes or resulting in a kinematically unstable solution. To prevent this, certain constraints must be imposed. In this way, any solution that proposes the removal of restricted or loaded nodes is discarded, as well as any truss that lacks kinematic stability. The latter can be identified by a singular stiffness matrix, meaning its determinant is zero, making further calculations impossible. Consequently, such solutions are also discarded.

Now, as in size optimization, the objective function is penalized whenever deflection or stress constraints are violated:

$$W = \sum_{e=1}^{Nm} \rho_e L_e A_e^{'} + c_f P \tag{11}$$

Where c_f is the weight factor of the penalty values, set to $c_f = 100000$ in this study to ensure a sufficiently large impact on the objective function, consistent with the values used in previous research [7]. Besides, the total penalty P is determined by the highest deflection or stress penalty value:

$$P = max\left\{ P_{e\_max(\delta)}, P_{e\_max(\sigma)} \right\} \tag{12}$$

Similarly, the deflection penalty $P_{e(\delta)}$ is given by the highest value among all truss members, where each value is either set to zero if the deflection remains within the proposed limits or normalized, as shown below:

$$P_{e(\delta)} = \begin{Bmatrix} \dfrac{\left|\delta_{c(x,y,z)}\right|}{\left|\delta^L\right|} \rightarrow if \rightarrow \delta_{c(x,y,z)} < \delta^L \\ 0 \rightarrow if \rightarrow \delta^L < \delta_{c(x,y,z)} < \delta^U \\ \dfrac{\left|\delta_{c(x,y,z)}\right|}{\left|\delta^U\right|} \rightarrow if \rightarrow \delta_{c(x,y,z)} > \delta^U \end{Bmatrix} \tag{13}$$

Similarly, the stresses penalty function $P_{e(\sigma)}$ follows the same approach:

$$P_{e(\sigma)} = \begin{cases} \dfrac{|\sigma_e|}{|\sigma^L|} \rightarrow if \rightarrow \sigma_e < \sigma^L \\ 0 \rightarrow if \rightarrow \sigma^L < \sigma_e < \sigma^U \\ \dfrac{|\sigma_e|}{|\sigma^U|} \rightarrow if \rightarrow \sigma_e > \sigma^U \end{cases} \tag{14}$$

2.3 Optimization process

The following sections describe the metaheuristic algorithms employed for truss weight optimization, highlighting their underlying principles, mathematical formulations, and distinctive features.

2.3.1 Teaching Learning Based Optimization

Teaching-Learning-Based Optimization (TLBO), proposed by Rao et al. [8] in 2011, is a population-based algorithm originally designed for solving mathematical and constrained mechanical design problems. It is inspired by the educational process in a classroom, where the role of the teacher helps to enhance the overall knowledge of the students, promoting collective learning among individuals. [2].

The algorithm operates through two sequential phases: teaching and learning. In the teaching phase, individuals are evaluated based on their objective function values, and the one with the best performance is designated as the teacher. The remaining members adjust their positions by considering both the average knowledge of the class and the influence of the teacher, which is expected to enhance their learning. This process is mathematically expressed as:

$$X_{new,i} = X_i + r(X_{teacher} - TF \cdot X_{mean}) \tag{15}$$

Where TF is the teaching factor and can be either 1 or 2, and r is a random number within the range of 0 to 1. X_i is the existing solution of the student i. If the evaluation of $X_{new,i}$ in the objective function results in an improvement of the previous value, it is accepted; otherwise, it is rejected and X_i is retained. [9].

During the learning phase, students aim to improve their performance through the interaction and support from their classmates. Two individuals are randomly selected, a student will only acquire knowledge if the other individual has a higher level of it, meaning that they have better results in the objective function. This can be described mathematically as:

$$X_{new,i} = X_i + r(X_j - X_i) \ \text{ if } \ f(X_j) < f(X_i) \tag{16}$$

$$X_{new,i} = X_i + r(X_i - X_j) \ \text{ if } \ f(X_i) < f(X_j)$$

$X_{new,i}$ is accepted only if evaluating the objective function yields a better result than X_i.

2.3.2 Flow Direction Algorithm

Flow Direction Algorithm (FDA), proposed by Karami et al. in 2021, is a physics inspired algorithm that simulates the way a drop of water goes through a drainage basin during a rainfall event. It has been successfully implemented in engineering problems. [10].

The main idea is that a drop of water travels along the basin topography, the outlet basin which is the point where most drops converge is considered the global maximum. The amount of flow travels to another cell based on the height and slope of the 8 neighboring cells, this is, the D8 Method, represented in Figure 1. The initial position of the *i*th flow is computed through the following equation:

$$Flow\_X(i) = lb + rand(ub - lb) \quad (17)$$

Where lb and ub are the upper and lower limit of the decision variables, and $rand$ a function that throws a value between 0 and 1 with uniform distribution. The neighbor *j*th position is described by the following:

$$Neighbor\_X(j) = Flow\_X(i) + randn * \Delta \quad (18)$$

In this equation, Δ controls the search scope; to achieve a balance between local refinement and global exploration Δ is defined as:

$$\Delta = (rand * Xrand - rand * Flow\_X(i)) * \|Best\_X - Flow\_X(i)\| * W \quad (19)$$

Where W is a nonlinear weight with random number between 0 and infinite. Finally, the new position of the flow can be determined as:

$$Flow\_newX(i) = Flow\_X(i) + V * \frac{Flow\_X(i) - Neighbor\_X(j)}{\|Flow\_x(i) - Neighbor\_x(j)\|} \quad (20)$$

The new position is accepted only if it improves the objective function compared to the initial one.

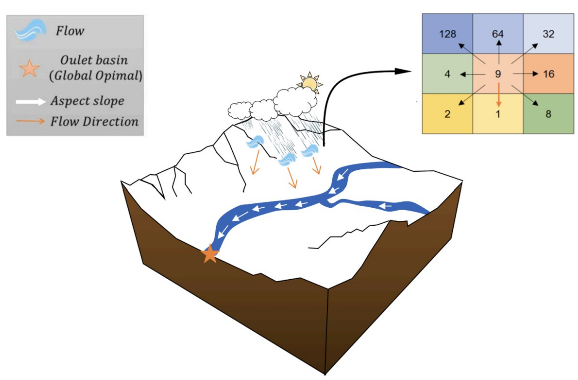

Figure 1: Outline of moving flow to outlet of basin and D8 method. [10].

2.3.3 Stochastic Paint Optimization

Stochastic paint optimization (SPO), proposed by Kaveh et al. in 2020, is an art-based optimization algorithm that applies color theory principles to paint mixing. It has been effectively applied to structural engineering problems, including the optimization of planar truss structures [11]. In SPO, the set of available colors represents the solution domain, while the different color combinations correspond to potential solutions. The optimization process of the algorithm relies on color combination techniques, structured around specific groups and rooted in the color wheel theory introduced in 1766.

Based on this framework, colors are categorized into three types: primary, secondary, and tertiary. Primary colors—yellow, blue, and red—serve as the foundation for all others, making them the most optimal solutions. Secondary colors, formed by mixing two primary colors, represent intermediate-quality solutions. Tertiary colors, derived from a combination of primary and secondary colors, are considered the least optimal. To efficiently explore and refine the solution space, SPO employs four combination techniques:

Analogous combination technique: Analogous color schemes combine colors that are near to each other on the color wheel. The result of this technique is defined as:

$$C_{new} = C_i + rand(C_{i+1} - C_{i-1}) \tag{21}$$

All C_i, C_{i+1}, C_{i-1}, belong to the same type of color. $rand$ is a random vector. C_{i+1} and C_{i-1} are other two random selected colors. This technique works as a local search phase.

Complementary combination technique: Complementary colors are opposite to each other on the wheel (e.g. red and green). The result of this technique is given by:

$$C_{new} = C_i + rand(C_{pi} - C_{ti}) \tag{22}$$

Where C_{pi}, C_{ti} are random selected colors from primary and tertiary type, respectively.

Triadic combination technique: A triadic color scheme is any three colors that are equally apart on the color wheel. The result obtained from this technique is defined as:

$$C_{new} = C_i + rand(\frac{C_{pi} + C_{si} + C_{ti}}{3}) \tag{23}$$

Where C_{pi}, C_{si} and C_{ti} are random selected colors from primary, secondary and tertiary type, respectively.

Tetradic combination technique: The solution obtained by this method is:

$$C_{new} = C_i + \frac{rand_1(C_{pi}) + rand_2(C_{si}) + rand_3(C_{ti}) + rand_4(C_{rand})}{4} \tag{24}$$

Different from other techniques, this one associates a random vector to each selected color from every category. Also, a fourth color is randomly selected from all kinds of colors.

3 NUMERICAL EXAMPLES

Benchmark truss problems are used to evaluate the performance of the optimization algorithms. In this case, a ten-bar truss and twenty-five-bar space truss problems are considered. For the size optimization of these structures, the basic tuning parameters of each algorithm—number of individuals and generations—must be defined. According to related studies [12], [13], typical values range from 50 to 300 for individuals and from 100 to 300 for generations. For this work, and based on a sensitivity analysis, it was determined that setting both parameters to 100 is enough to ensure TLBO, FDA and SPO convergence while avoiding unnecessary computational costs.

On the other hand, in the topological optimization problem, the ranges for individuals and generations differ slightly from the previous case. This is because the problem is more complex, increasing the computational effort per iteration. However, the search space is smaller due to the higher number of constraints, allowing convergence to be achieved with lower parameter values. Based on previous studies [14], the recommended range for this problem is between 10 and 50 individuals and between 50 and 100 generations. For this work, both parameters are set to 50.

3.1 Ten-bar truss

Figure 2 shows the ten-bar truss model and its discretization into nodes and elements.

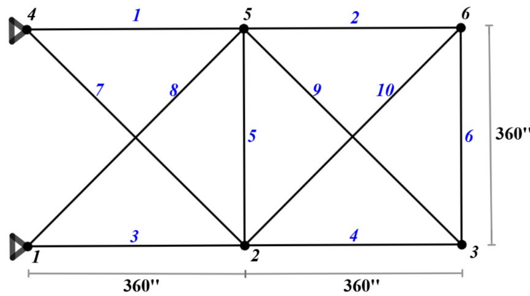

Figure 2: Ten-bar truss benchmark problem.

For size optimization, the variables required for the analysis are defined. In this structure, cross-sectional areas can range from 0.1 in.² to 35.0 in.². The material properties include a unit weight of 0.1 lb/in.³ and a modulus of elasticity of 10^7 psi. The design constraints impose a maximum allowable stress of ±25 ksi in any truss member and a maximum deflection of ±2.0 in. at any node (in both directions). Additionally, two points loads of 100 kips each are applied at nodes 2 and 3. In topology optimization, the same variables are kept, but the cross-sectional areas range from -3 in.² to 35 in.², where any negative value is automatically set to zero, indicating the removal of a member.

3.2 Twenty-five-bar space truss

Figure 3 presents the twenty-five-bar truss model and its discretization into nodes and elements.

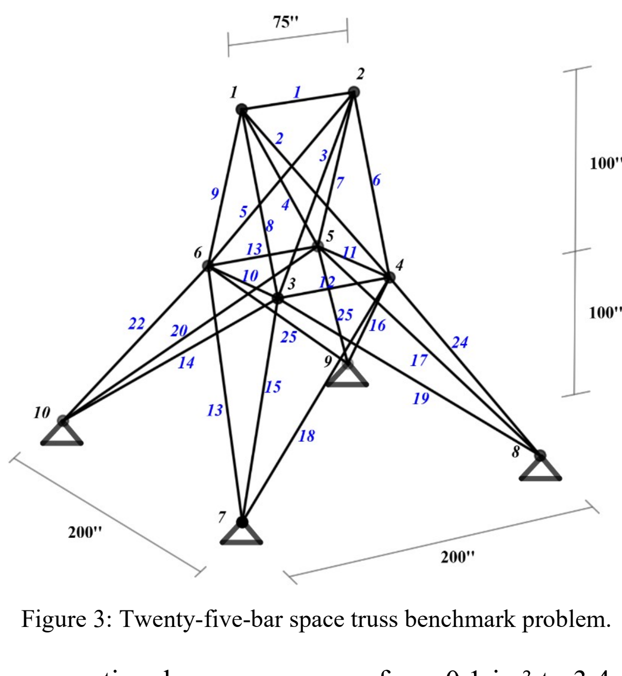

Figure 3: Twenty-five-bar space truss benchmark problem.

In this structure, cross-sectional areas can range from 0.1 in.² to 3.4 in.² for size optimization. The 25 bars are grouped into 8 categories based on their symmetry: (1) A_1 ,

(2) $A_2 - A_3 - A_4 - A_5$, (3) $A_6 - A_7 - A_8 - A_9$, (4) $A_{10} - A_{11}$, (5) $A_{12} - A_{13}$, (6) $A_{14} - A_{16} - A_{17} - A_{23}$, (7) $A_{18} - A_{19} - A_{20} - A_{21}$, and (8) $A_{15} - A_{22} - A_{24} - A_{25}$. The material properties include a unit weight of 0.1 lb/in.^{3} and a modulus of elasticity of 10^7 psi. The design constraints set a maximum allowable stress of 35 ksi for any truss member, with minimum stress values varying for each element group, listed in Table 1.

Element group	Members	Compression (ksi)
1	1	35.092
2	2,3,4,5	11.590
3	6,7,8,9	17.305
4	10,11	35.092
5	12,13	35.092
6	14,16,17,23	6.759
7	18,19,20,21	6.759
8	15,22,24,25	11.082

Table 1: Allowable compression stresses for the 25-bar-truss.

The maximum deflection is ±0.35 in. at any node (in all directions). Additionally, the load case applied to the structure is presented in Table 2.

Node	P_x (lb)	P_y (lb)	P_z (lb)
1	1,000	-10,000	-10,000
2	0	-10,000	-10,000
3	600	0	0
6	500	0	0

Table 2: Loading of 25-bar-truss.

For topology optimization, the cross-sectional areas range from -3 in² to 3 in², the maximum allowable stress is ±40 ksi in any truss member and the maximum deflection is ±0.35 in. at any node (in all directions).

4 RESULTS AND DISCUSSION

4.1 Ten-bar truss

4.1.1 Size optimization

Table 3 lists the best design obtained by each algorithm. Initially, it can be observed that members 2, 5, and 10 have the smallest cross-sectional areas in all cases. This suggests that these elements contribute minimally to the overall structural load-bearing capacity, enabling the optimization process to reduce their size without compromising stability or constraint limitations. In terms of structural weight, TLBO and SPO produced nearly identical results, 5105.88 lb and 5105.86 lb respectively, while FDA produced a higher weight. Although FDA demonstrated the fastest execution time, this resulted in a slight trade-off in solution accuracy. Comparing SPO and TLBO, the latter stands out as the most efficient choice, achieving a comparable objective function value in nearly half of the computation time required by SPO.

Additionally, cross-sectional areas of the members show minor variations across the algorithms, with TLBO and SPO producing similar results for most members, indicating that these algorithms converge to similar solutions. However, FDA assigns higher areas to members 3 and 8, suggesting a more conservative approach to maintaining structural integrity.

Variables		Cross-sectional areas (in^2)		
Element group	Members	TLBO	FDA	SPO
1	1	30.77	28.69	30.81
2	2	0.10	0.10	0.10
3	3	23.43	22.53	23.41
4	4	15.38	15.81	15.37
5	5	0.10	0.10	0.10
6	6	0.56	0.68	0.56
7	7	7.52	7.70	7.53
8	8	21.19	22.85	21.21
9	9	21.74	21.60	21.70
10	10	0.10	0.10	0.10
Weight (lb)		5105.88	5121.43	5105.86
Time (s)		2.39	1.91	4.68

Table 3: Design for 10-bar truss using size optimization.

Figure 5 illustrates the convergence process of the objective function for each optimization algorithm. All algorithms exhibit rapid convergence, reaching the optimal value within 50 iterations. However, SPO demonstrates particularly fast convergence, stabilizing an adequate result in less than 20 iterations. In contrast, FDA shows the slowest convergence and appears to stagnate at local minima. This behavior suggests that FDA may not be the most suitable choice for problems with time constraints, where fewer iterations are required, as it may not provide promising results. Lastly, TLBO shows acceptable performance, considering its fast execution and its ability to stabilize a satisfactory result after 30 iterations.

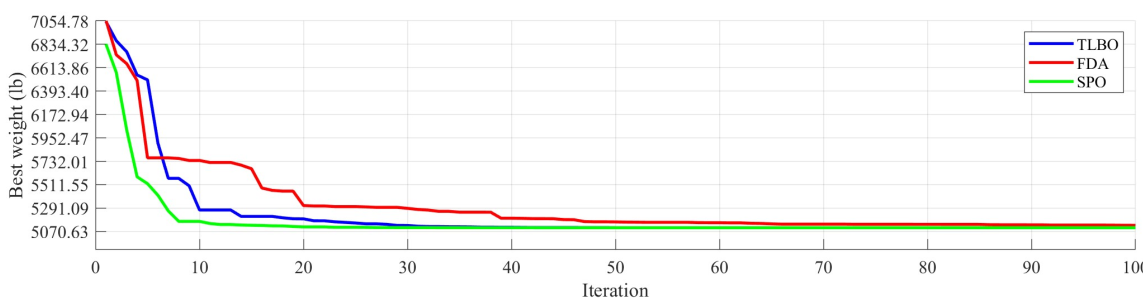

Figure 5: Convergence Comparison of Optimization Algorithms.

4.1.2 Size and topology optimization

Table 4 presents the optimal design proposed by each algorithm for the topology problem. Most of the members, which had the smallest cross-sectional areas in the size optimization, were eliminated in this case. TLBO and FDA were the most aggressive in removing members, eliminating members 2, 5, 6 and 7. In contrast, SPO removed the same members except for 6. This indicates that TLBO and FDA adopted a material-reduction strategy by eliminating entire members, while SPO appeared to prioritize maintaining a more consistent structural configuration.

Variables		Cross-sectional areas (in^2)		
Element group	Members	TLBO	FDA	SPO
1	1	30.09	29.62	34.92
2	2	REMOVED	REMOVED	REMOVED
3	3	24.97	27.27	20.75
4	4	17.28	13.50	15.33
5	5	REMOVED	REMOVED	REMOVED
6	6	REMOVED	REMOVED	0.60
7	7	10.03	13.74	6.50
8	8	17.58	19.05	20.79
9	9	23.83	24.18	20.48
10	10	REMOVED	REMOVED	REMOVED
Weight (lb)		5105.88	5435.02	5009.50

Table 4: Design for 10-bar truss using size and topology optimization.

This strategy aims to avoid creating a truss configuration that could lead to significant penalties due to constraint violations. Consequently, SPO achieves the lowest weight. This also explains why FDA design has the highest weight: the elimination of elements required increasing the cross-sectional areas of the remaining members. This adjustment led to violations of deflection and stress limits as well, raising the penalty function value. Figure 6 illustrates the ten-bar truss after the topological optimization process.

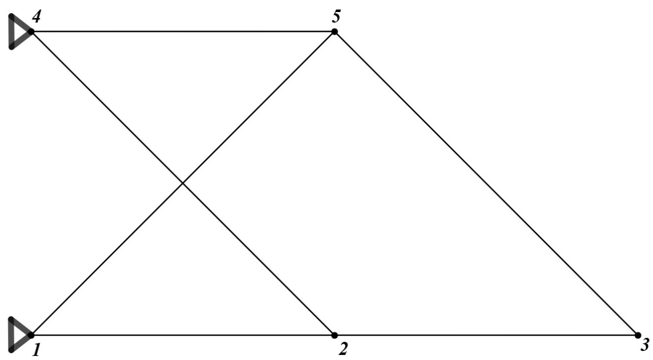

Figure 6: Best found topology for ten-bar truss.

4.2 Twenty-five-bar space truss

4.2.1 Size optimization

Table 5 lists the best designs obtained. While the weight of the structure is nearly identical across all cases, a notable difference is observed in the execution times. The SPO algorithm is significantly slower compared to TLBO and FDA, which have similar runtimes. Therefore, when time is a limiting factor, TLBO or FDA may be more suitable choices.

It is also noteworthy that the member groups from critical cases, such as groups 1, 3, 4, and 8, show identical results across all algorithms. This highlights the algorithms prioritization of members that are crucial for the overall stability of the truss, aiming to maintain their strength as high as possible, while reducing the cross-sections of less critical members to optimize the performance of the structure.

Variables		Cross-sectional areas (in^2)		
Element group	Members	TLBO	FDA	SPO
1	1	0.10	0.10	0.10
2	2,3,4,5	0.44	0.49	0.42
3	6,7,8,9	3.40	3.40	3.40
4	10,11	0.10	0.10	0.10
5	12,13	1.91	1.94	1.93
6	14,16,17,23	0.96	0.92	0.97
7	18,19,20,21	0.46	0.47	0.47
8	15,22,24,25	3.40	3.40	3.40
Weight (lb)		484.02	484.23	484.02
Time (s)		3.35	3.63	7.93

Table 5: Design for 25-bar truss using size optimization.

Figure 7 illustrates the convergence process of the objective function for each optimization algorithm. It can be observed that TLBO and SPO exhibit similar convergence, stabilizing their results before 30 iterations in both cases. SPO shows a small advantage with a more rapid initial slope, reaching convergence just after 10 iterations. In contrast, FDA demonstrates slower convergence, stabilizing only after 60 iterations and initially experiencing some stagnation at a local minimum. Considering both convergence and runtime, TLBO proves to be the most efficient algorithm overall.

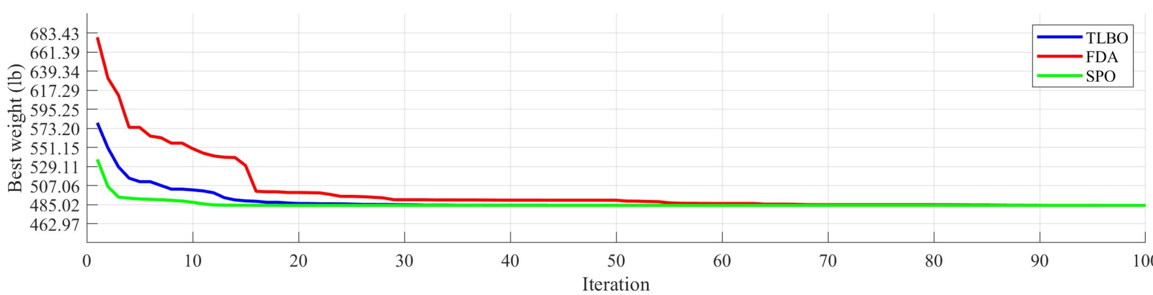

Figure 7: Convergence Comparison of Optimization Algorithms.

4.2.2 Size and topology optimization

The results of the topological optimization for the same truss design are presented in Table 6. In this case, the optimization process led to the removal of several members across all algorithms. For instance, group members 1, 4 and 5 were eliminated in TLBO and SPO, which indicates that these members were not necessary for the overall stability of the structure. In contrast, FDA demonstrates lower performance, removing only one element and resulting in the highest structure weight. Despite SPO removing more elements, the objective function did not decrease significantly, remaining at a value similar to FDA case. This could be attributed to a high penalty function value in the SPO case, caused by violations of the imposed constraints. Overall, these observations suggest that TLBO was the most efficient in identifying critical elements and eliminating non-essential members, achieving a lighter structure without compromising its integrity. Now, compared to the previous truss, this indicates that in problems with a larger search space, the design variables tend to differ more, clearly reflecting the strategies employed by each algorithm.

Variables		Cross-sectional areas (in^2)		
Element group	Members	TLBO	FDA	SPO
1	1	REMOVED	REMOVED	REMOVED
2	2,3,4,5	0.76	1.38	1.03
3	6,7,8,9	3.60	3.00	3.00
4	10,11	REMOVED	0.25	REMOVED
5	12,13	REMOVED	1.33	REMOVED
6	14,16,17,23	1.18	0.77	0.86
7	18,19,20,21	1.50	1.17	1.61
8	15,22,24,25	3.70	3.00	3.00
Weight (lb)		508.50	524.78	521.52

Table 6: Design for 25-bar truss using size and topology optimization.

Figure 8 presents the most optimal topology of the twenty-five-bar truss, where the remaining elements are the only ones essential for meeting all system requirements.

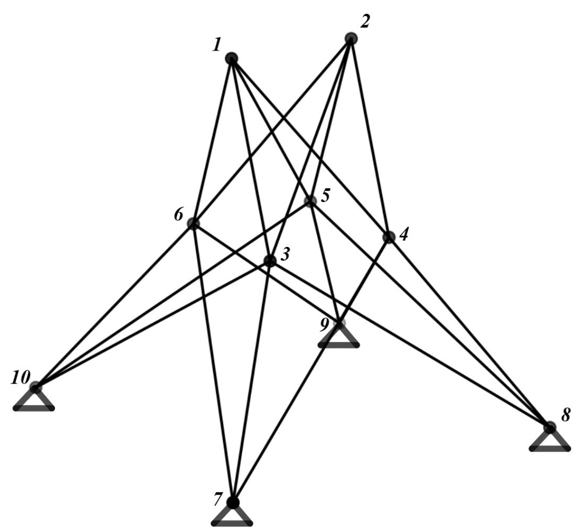

Figure 8: Best found topology for twenty-five-bar space truss.

5 CONCLUSIONS

It has been demonstrated that TLBO outperforms SPO and FDA in truss optimization problems, achieving the best balance between weight reduction and computational efficiency. While SPO showed faster convergence in smaller problems, its processing time was longer for larger structures. FDA, in contrast, exhibited slower convergence and generated designs with larger cross-sectional areas, leading to less efficient solutions. Overall, TLBO proved to be the most effective algorithm, consistently providing optimal results in both size and topology optimization across all truss configurations. Furthermore, its robustness in handling diverse problem sizes and its ability to maintain computational efficiency under varying constraints make it a reliable choice for real-world applications. These findings underline the importance of selecting the right algorithm based on the problem characteristics to achieve both optimal designs and efficient performance.

REFERENCES

[1] R. Awad, “Sizing optimization of truss structures using the political optimizer (PO) algorithm”, *Structures*, vol. 33, pp. 4871–4894, 2021.

[2] A. Mortazavi, “The Performance Comparison of Three Metaheuristic Algorithms on The Size, Layout and Topology Optimization of Truss Structures”, *Mugla J. Sci. Technol*., 2019.

[3] L. F. F. Miguel, R. H. Lopez y L. F. F. Miguel, “Multimodal size, shape, and topology optimization of truss structures using the Firefly algorithm”, *Advances Eng. Softw*., vol. 56, pp. 23–37, 2013.

[4] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Metaheuristic algorithms: A comprehensive review", *Elsevier eBooks,* pp. 185-231, 2018.

[5] A. K. Kar, “Bio inspired computing – A review of algorithms and scope of applications”, *Expert Syst. with Appl*., vol. 59, pp. 20–32, 2016.

[6] C. V. Camp y M. Farshchin, “Design of space trusses using modified teaching–learning based optimization”, *Eng. Struct.*, vol. 62-63, pp. 87–97, 2014.

[7] J.-P. Li, “Truss topology optimization using an improved species-conserving genetic algorithm”, *Eng. Optim.*, vol. 47, n. º 1, pp. 107–128, 2014.

[8] R. V. Rao, V. Savsani and D. P. Vakharia, “Teaching-learning-based optimization: An optimization method for continuous non-linear large-scale problems”. *Information Sciences*, vol. 183, 2012.

[9] A, Baghlani and M.H. Makiabadi, “Teaching-Learning-Based Optimization Algorithm for Shape and Size Optimization of Truss Structures with Dynamic Frequency Constraints”. *Iranian journal of science and technology,* 2013.

[10] H. Karami, M. V. Anaraki, S. Farzin y S. Mirjalili, “Flow Direction Algorithm (FDA): A Novel Optimization Approach for Solving Optimization Problems”, *Comput. & Ind. Eng*., vol. 156, p. 107224, 2021.

[11] A. Kaveh, S. Talatahari y N. Khodadadi, “Stochastic paint optimizer: theory and application in civil engineering”, *Eng. with Comput*., 2020.

[12] S. K. Azad, M. Bybordiani, S. K. Azad y F. K. J. Jawad, “Simultaneous size and geometry optimization of steel trusses under dynamic excitations”, *Structural Multidisciplinary Optim.*, vol. 58, n. º 6, pp. 2545–2563, 2018.

[13] F. K. J. Jawad, M. Mahmood, D. Wang, O. AL-Azzawi y A. AL-JAMELY, “Heuristic dragonfly algorithm for optimal design of truss structures with discrete variables”, *Structures*, vol. 29, pp. 843–862, 2021.

[14] H. Kawamura, H. Ohmori y N. Kito, “Truss topology optimization by a modified genetic algorithm”, *Structural Multidisciplinary Optim.*, vol. 23, n. º 6, pp. 467–473, 2002.