mathematics

MDPI

Article

Enhanced Structural Design of Prestressed Arched Trusses through Multi-Objective Optimization and Multi-Criteria Decision-Making

Andrés Ruiz-Vélez [1], José García [2], Gaioz Partskhaladze [3], Julián Alcalá [1] and Víctor Yepes [1,*]

[1] Institute of Concrete Science and Technology (ICITECH), Universitat Politècnica de València, 46022 València, Spain; aruivel@doctor.upv.es (A.R.-V.); jualgon@upv.es (J.A.)
[2] Escuela de Ingeniería de Construcción y Transporte, Pontificia Universidad Católica de Valparaíso, Valparaíso 2362804, Chile; jose.garcia@pucv.cl
[3] Engineering and Construction Department, Faculty of Technologies, Batumi Shota Rustaveli State University, 35/32, Ninoshvili/Rustaveli Str., 6010 Batumi, Georgia; gizo.partskhaladze@bsu.edu.ge
* Correspondence: vyepesp@cst.upv.es

Abstract: The structural design of prestressed arched trusses presents a complex challenge due to the need to balance multiple conflicting objectives such as structural performance, weight, and constructability. This complexity is further compounded by the interdependent nature of the structural elements, which necessitates a comprehensive optimization approach. Addressing this challenge is crucial for advancing construction practices and improving the efficiency and safety of structural designs. The integration of advanced optimization algorithms and decision-making techniques offers a promising avenue for enhancing the design process of prestressed arched trusses. This study proposes the use of three advanced multi-objective optimization algorithms: NSGA-III, CTAEA, and SMS-EMOA, to optimize the structural design of prestressed arched trusses. The performance of these algorithms was evaluated using generational distance and inverted generational distance metrics. Additionally, the non-dominated optimal designs generated by these algorithms were assessed and ranked using multiple multi-criteria decision-making techniques, including SAW, FUCA, TOPSIS, PROMETHEE, and VIKOR. This approach allowed for a robust comparison of the algorithms and provided insights into their effectiveness in balancing the different design objectives. The results of the study indicated that NSGA-III exhibited superior performance with a GD value of 0.215, reflecting a closer proximity of its solutions to the Pareto front, and an IGD value of 0.329, indicating a well-distributed set of solutions across the Pareto front. In comparison, CTAEA and SMS-EMOA showed higher GD values of 0.326 and 0.436, respectively, suggesting less convergence to the Pareto front. However, SMS-EMOA demonstrated a balanced performance in terms of constructability and structural weight, with an IGD value of 0.434. The statistical significance of these differences was confirmed by the Kruskal–Wallis test, with p-values of 2.50×10^{-15} for GD and 5.15×10^{-06} for IGD. These findings underscore the advantages and limitations of each algorithm, providing valuable insights for future applications in structural optimization.

Keywords: multi-objective optimization; multi-criteria decision-making; NSGA-III; CTAEA; SMS-EMOA; SAW; FUCA; TOPSIS; PROMETHEE; VIKOR

MSC: 90C11; 90C27; 90C29

Citation: Ruiz-Vélez, A.; García, J.; Partskhaladze, G.; Alcalá, J.; Yepes, V. Enhanced Structural Design of Prestressed Arched Trusses through Multi-Objective Optimization and Multi-Criteria Decision-Making. *Mathematics* **2024**, *12*, 2567. https://doi.org/10.3390/math12162567

Academic Editor: Hsien-Chung Wu

Received: 29 July 2024
Revised: 18 August 2024
Accepted: 19 August 2024
Published: 20 August 2024

1. Introduction

Optimization techniques are crucial in modern engineering practices. In structural engineering, design optimization focuses on maximizing the contribution and minimizing the consequences of structures in our environment. Single-objective approaches optimize one relevant feature of the problem, often related to economic cost and environmental or

social ramifications. Single-objective optimization (SOO) algorithms generate solutions that excel in either limiting or enhancing one structural characteristic [1]. However, complex engineering problems often involve multiple considerations of similar importance. In this context, multi-objective optimization (MOO) algorithms enable the simultaneous consideration of multiple criteria [2–5]. Additionally, structural design optimization frameworks integrating multi-criteria decision-making (MCDM) techniques for alternative scoring and ranking provide a robust approach to balancing various competing factors [6–10].

Prestressed structures achieve structural performance within limited material consumption. By effectively utilizing the elastic working range of materials, prestressed designs often reduce resource usage without compromising structural integrity. Their reduced weight and excellent load-bearing capacity make them ideal for long-spanning truss structures such as bridges, high-rise buildings, and roofs. These applications benefit from deflection control and enhanced stability intrinsic to prestressing [11–13]. However, designing prestressed truss elements presents unique challenges.

Traditional prestressed element design methods are complex and have several limitations [14–16]. Accurately defining the prestressing force is challenging, leading to difficulties in sizing truss elements and achieving optimal weight minimization. Different unloading forces in the top and bottom chords after prestressing require larger cross-sections than necessary during actual load-bearing phases, increasing the self-weight and cost of the truss [17,18]. The reliance on iterative processes for force estimation often results in imprecise and time-consuming calculations, further complicating material efficiency and cost-effectiveness.

Partskhaladze et al. [19] outline an SOO approach targeting weight minimization for a new prestressed arched truss element. Their work thoroughly elaborates on how prestressed structures extend the elastic capabilities of materials, resulting in improved load-carrying characteristics. It also addresses the complexities and limitations of traditional design methods, introducing a novel design approach using a stiffness matrix method for calculations coupled with a single-objective simulated annealing (SA) optimization algorithm [20]. The weight minimization results highlight the effectiveness of this design method and underscore weight reduction as a strategy for material efficiency in these structural elements.

While effective for targeted weight reduction, this novel SOO approach can be enhanced to better address the complex nature of structural engineering problems. MOO is particularly relevant for considering multiple performance metrics essential for efficient structural design. Additionally, a robust design framework that integrates MOO and MCDM techniques provides a comprehensive structural design optimization and ranking approach.

This paper explores the performance of the non-dominated sorting genetic algorithm III (NSGA-III), the two-archive evolutionary algorithm (CTAEA), and the hypervolume measure–evolutionary multi-objective optimization algorithm (SMS-EMOA) [21–23]. The multi-objective optimization (MOO) problem includes the two-stage load-bearing requirements of the prestressed truss element, aiming to maximize structural performance in both non-prestressed and prestressed states, as shown in Figure 1. Building on previous studies focused on weight minimization, this research introduces a third objective to reduce self-weight. Additionally, a fourth objective addresses constructability issues, making assembly easier and enhancing safety in construction.

To address these complex and often conflicting objectives, an optimization framework is proposed. Within this framework, different objective functions are integrated, and three specific optimization algorithms—NSGA-III, CTAEA, and SMS-EMOA—are applied. The interconnection of these algorithms is key, as it is recognized that no single algorithm is universally optimal; each has its own strengths and weaknesses. By structuring the problem within this flexible framework, the optimization process leverages the complementary capabilities of these algorithms, tailoring the approach to the specific needs of the design and ensuring a more balanced and effective solution.

Following the optimization process, the framework integrates several multi-criteria decision-making (MCDM) techniques, such as simple additive weighting (SAW), fair un choix adéquat (FUCA), technique for order of preference by similarity to ideal solution (TOPSIS), preference ranking organization method for enrichment evaluation (PROMETHEE), and "visekriterijumska optimizacija i kompromisno resenje" (VIKOR), combined with an entropy-based strategy for objective criteria weighting [24–29]. The interconnection between the optimization algorithms and these MCDM techniques is crucial, as the MCDM methods assist in the decision-making process by evaluating and ranking the optimized designs based on multiple criteria. This synergy ensures that the most balanced and practical solutions are identified, taking into account the various trade-offs that are important for real-world applications.

Additionally, the framework includes a series of analytical visualizations that provide insights into the variability in different variables within the Pareto front. These visualizations help in understanding how different optimization strategies affect the design, offering a clearer picture of the trade-offs involved. This framework can be applied not only to prestressed trusses but also to various structural designs, making it a powerful tool for optimization and decision-making in structural engineering.

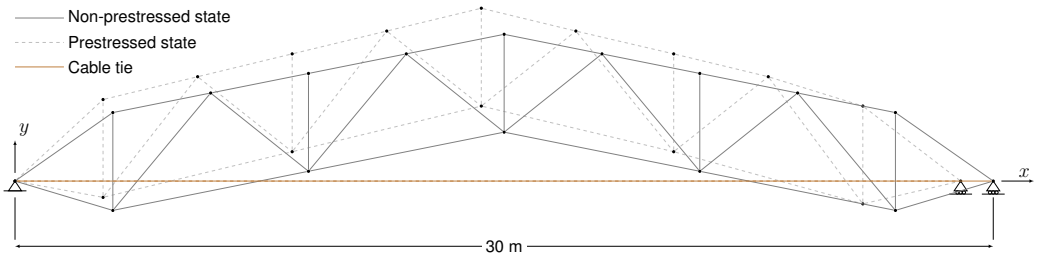

Figure 1. Multi-state illustration of the truss structure: solid line–non-prestressed state; dashed line–prestressed state; and solid colored line–cable tie element.

This research contributes to modern engineering practices by implementing an integrated framework for prestressed design leveraging MOO algorithms and MCDM techniques [30–32]. The results advance the understanding and application of prestressed trusses by addressing the complexities of traditional design methods and incorporating comprehensive evaluation metrics [33,34]. The framework, which includes objectives for weight minimization, load-bearing performance, and constructability, ensures a balanced and practical design process. These innovations enhance structural performance, reduce resource consumption, and improve constructability and safety, demonstrating practical implications for more efficient and sustainable construction practices.

2. Methods

This section presents a clear outline of the problem employed to evaluate the effectiveness of optimization algorithms within the combined MOO and MCDM multi-state design framework. It also comprehensively describes the optimization algorithms used, explains the criteria weighting process, and details the decision-making techniques applied in this study.

2.1. Optimization Problem Overview

This paper broadens the scope from the SOO in Partskhaladze et al. [19], introducing the MOO structural design optimization of a prestressed arched truss element. The MOO problem involves optimizing multiple conflicting objectives simultaneously. Each solution vector $\vec{X}$ comprising n decision variables represents a potential design within the solution space and is expressed via Equation (1).

$$\vec{X} = (x_1, x_2, \ldots, x_n) \tag{1}$$

Throughout the optimization process, specific parameters remain fixed, such as physical constants, boundary conditions, and other essential values. Section 2.1.1 delineates the decision variables and parameter values considered in the study.

The k objective functions $f(\vec{X})$ aim to either maximize or minimize specific outcomes. This study targets four objective functions, generally expressed via Equation (2), where f_1 and f_2 are to be maximized, and f_3 and f_4 are minimized. Section 2.1.3 details the objective functions and the computation approach used in this study.

$$optimize(f(\vec{X})) = \begin{cases} max(f_i(\vec{X})) & \text{if } i \in (1,2) \\ min(f_i(\vec{X})) & \text{if } i \in (3,4) \end{cases} \tag{2}$$

The optimization process is subject to m constraints $g(\vec{X})$, typically expressed through Equation (3). Section 2.1.2 outlines the constraints that define the feasible solution space for the MOO problem in this study.

$$g_m(\vec{X}) \leq 0 \tag{3}$$

In multi-objective optimization, the aim is to effectively explore the solution space to identify a set of balanced solutions, known as the Pareto front [35]. The Pareto front consists of non-dominated solutions, where a solution $\vec{X}_1$ is said to dominate ($\succ$) another solution $\vec{X}_2$ if it satisfies the conditions outlined in Equation (4).

$$\vec{X}_1 \succ \vec{X}_2 \iff \begin{cases} f_i(\vec{X}_1) \geq f_i(\vec{X}_2) & \text{for all } i = 1, 2, \ldots, k \\ f_i(\vec{X}_1) > f_i(\vec{X}_2) & \text{for at least one } i \end{cases} \tag{4}$$

The set of non-dominated solutions forms the Pareto front, representing the trade-offs between different objectives. The multi-objective algorithms detailed in Section 2.2 are designed to identify balanced solutions, effectively highlighting possible alternatives to be evaluated through decision-making strategies.

2.1.1. Variables and Parameters

The truss structure comprises a rigid primary framework and a tie-member cable. The mathematical model, depicted in Figure 2, includes 30 structural elements and 16 joints. These elements are arranged within the upper chord (elements 1–5 and 16–20), lower chord (elements 6–8 and 21–23), lattice elements (elements 9–15 and 24–29), and the tie member (element 30).

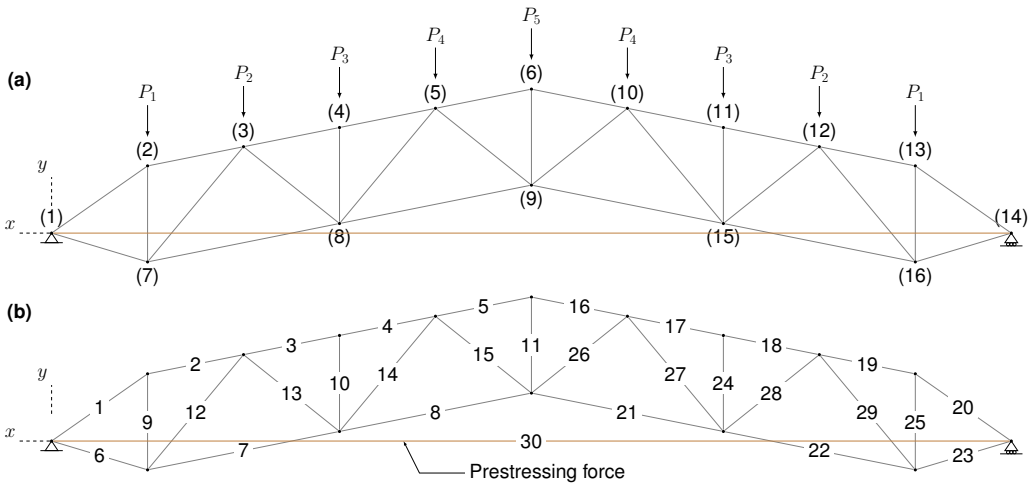

Figure 2. Prestressed arched truss illustration: (**a**) nodes (1)–(16) and load parameters P_1–P_5; and (**b**) truss elements 1–29, cable tie 30, and prestressing force variables.

Support nodes with restrained degrees of freedom are located at positions 1 and 14, strut nodes at 7 and 16, and loading nodes at elements 2–6 and 16–20. The slender part of the structure is the tie-member cable, designated as segment 30. Truss elements are constructed from equal-leg angles in a back-to-back configuration, with cross-sectional and linear mass characteristics detailed in Table 1 and illustrated in Figure 3.

Table 1. Cross-sectional characteristics of equal-leg angles used in the truss elements, configured back-to-back, in accordance with GOST 8509–93 [36].

b (mm)	t (mm)	A (cm^2)	i_x (cm)	i_y (cm)	m (kg/m)
50	5	4.88	1.53	2.45	3.77
63	5	6.13	1.94	2.96	4.81
70	5	6.86	2.16	3.22	3.38
75	6	8.78	2.30	3.44	6.89
80	6	9.38	2.47	3.65	7.36
90	6	10.60	2.78	4.03	8.33
90	7	12.30	2.77	4.06	9.64
100	7	13.80	3.08	4.44	10.80
100	8	15.60	3.07	4.47	12.20
110	8	17.20	3.39	4.87	13.50
125	8	19.70	3.87	5.46	15.50
125	9	22.00	3.86	5.48	17.30
140	9	24.70	4.34	6.09	19.40
140	10	27.30	4.33	6.11	21.50
160	10	31.40	4.96	6.91	24.70
160	11	34.40	4.95	6.93	27.00
160	16	49.10	4.89	7.03	38.50
180	11	38.80	5.60	7.74	30.50
180	12	42.20	5.59	7.76	33.10
200	12	47.10	6.22	8.55	37.00
200	14	54.60	6.20	8.60	42.80
200	16	62.00	6.17	8.64	48.70
220	16	68.60	6.81	9.42	53.80
250	16	78.40	7.76	10.6	61.50
250	20	97.00	7.71	10.7	76.10

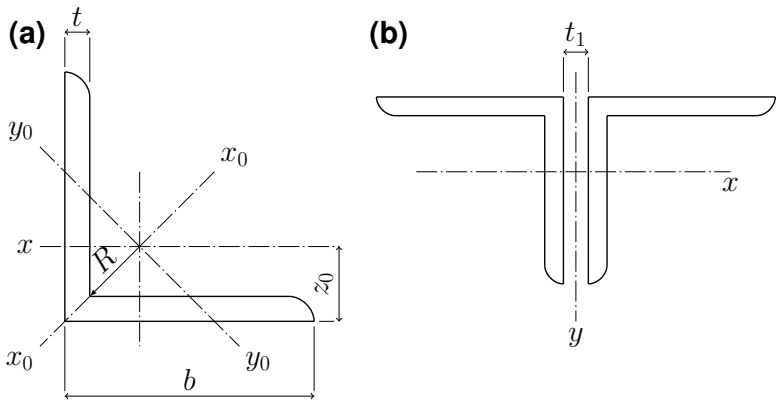

Figure 3. Truss elements profile geometry: (**a**) cross-sectional dimensions of equal-leg angles; and (**b**) back-to-back cross-sectional configuration, in accordance with GOST 8509-93 [36].

Aligning with previous work [19], the primary structure is S245 low-carbon hot-rolled mild steel, while the tie member uses high-strength steel. The cable properties are outlined in Table 2. The truss elements have a design resistance of $R_y = 24$ kN/cm^2 and an elastic modulus of $E = 21,000$ kN/cm^2. The cable has a design resistance of $R_u = 200$ kN/cm^2 and an $E = 16,000$ kN/cm^2 elastic modulus. Table 3 also details the dead weight and live load values considered in the study.

Table 2. Cross-sectional characteristics of the steel cable used for the tie member of the truss, in accordance with GOST 3068-66 [37].

Φ_c (cm)	A_c (cm^2)	m_c (kg/m)	R_c (kN)
3.80	6.61	5.86	907.50
4.20	8.15	7.24	1120
4.60	9.87	8.75	1350
5.10	11.74	10.45	1610
5.50	13.81	12.25	1895
5.90	16.01	14.20	2200
6.30	18.37	16.30	2525
6.80	20.90	18.55	2825

Table 3. Nodal load values for the computational model, including live-load, dead-load, and application elements [19].

	Unit	P_1	P_2	P_3	P_4	P_5
Live-load	(kN)	57.60	57.60	83.40	91.80	30.00
Dead-load	(kN)	85.40	85.40	73.00	90.70	30.00
Element	(num)	2, 13	3, 12	4, 11	5, 10	6

The sections of the 29 truss elements, the cable section, and the prestressing force are the 31 decision variables modified to generate different solutions during optimization. The positions of the nodes, the joints, the loads, and the sectional characteristics are pre-established known parameters that remain fixed during optimization.

2.1.2. Computational Model and Constraints

The internal stress for each member of the prestressed arched truss, including the tie cable, is determined using the displacement method. This study follows the geometry, boundary conditions, dead loads, and live loads in Partskhaladze et al. [19], for a truss spanning 30 meters. An object-oriented programming (OOP) Python 3 proprietary computational model was developed and utilized to execute the displacement method and implement the MOO algorithms and MCDM strategies [38].

The displacement method is conducted under the two-dimensional planar hypothesis, where the structure stiffness matrix (K_s) and nodal forces (F) are known, and the primary unknowns are the displacements (U) at each node. Equation (5) illustrates the general equation of this method.

$$K_s \cdot U = F \tag{5}$$

The nodal displacements in the computational model are defined by two degrees of freedom: horizontal (u_x) and vertical (u_y) displacements. Assembling the structure's stiffness matrix necessitates the preliminary computation of each truss element's stiffness matrix (K_e). Each K_e is computed first by incorporating a member length (L) per Equation (6).

$$L = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{6}$$

Next, the nodal position is determined using the element's cosine (c) and sine (s) values in global coordinates, following Equations (7) and (8).

$$c = \frac{(x_2 - x_1)^2}{L} \tag{7}$$

$$s = \frac{(y_2 - y_1)^2}{L} \tag{8}$$

Finally, the sectional characteristics, including elastic modulus (E) and cross-sectional area (A), are integrated, with K_e being computed according to Equation (9). The cable K_e contains values corresponding to its structural behavior.

$$K_e = \frac{EA}{L} \cdot \begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix} \tag{9}$$

Once K_e is computed for all members, the structure's K_s is assembled, considering the connections and positioning. The computational model then constructs the F vector and generates the reduced K_s and F by considering the restrained degrees of freedom based on the boundary conditions at nodes 1 and 14. Finally, the resulting system of linear equations is solved to determine the nodal displacements. With U known, the internal forces in each element are calculated directly using the element stiffness relationships.

A multi-state constraint verification (M-SCV) system evaluates the structural capability and constraint compliance of each design generated during the MOO process. Algorithm 1 outlines the procedure applied to each solution during optimization to determine its feasibility. Solutions are classified as feasible (multi-state structural compliance) or unfeasible (non-compliance in non-prestressed, prestressed, or both states).

Algorithm 1 M-SCV algorithm implementation

Function M-SCV(*model*)
Input: *model* – Python computational model
Output: *S_1*, *S_2* – Non-prestressed and prestressed compliance vectors
S_1 ← array of size len(*model.elements*), initialized to "non-compliant"
S_2 ← array of size len(*model.elements*), initialized to "non-compliant"
Conduct **Stress analysis** on *model.global*
for each *state* in *model.states* **do**
  for *i*, *element* in enumerate(*model.elements*) **do**
    if non-prestressed state **then**
      Initiate **non-prestressed analysis** on *model.element*
      S_1_check ← *element.state_1_stress* ≤ *element.compute_resistance*()
      if *S_1_check* **then**
        S_1[*i*] ← "compliant"
      end if
    else if prestressed state **then**
      Initiate **prestressed analysis** on *model.elements*
      S_2_check ← *element.state_2_stress* ≤ *element.compute_resistance*()
      if *S_2_check* **then**
        S_2[*i*] ← "compliant"
      end if
    end if
  end for
end for
return *S_1*, *S_2*

The outputs of the constraint verification process within the computational model are vectors with lengths equal to the number of model members. A null value indicates compliance, while a non-null value (1 in this study) signifies non-compliance. Summing the validation vector reveals the number of elements that do not comply with the truss structure's non-prestressed or prestressed states.

2.1.3. Objective Functions

The MOO for this problem incorporates four distinct objective functions, aiming to improve safety and operational performance in construction projects that integrate prestressed elements. Two of these functions aim to maximize structural performance in

both non-prestressed and prestressed states. The third objective focuses on minimizing the structure's weight, aligning with previous work to ensure efficient resource utilization and enhance the structure's environmental profile. The fourth objective addresses constructability issues.

The first objective function evaluates the ability of the truss's structural elements to withstand loads in the non-prestressed state. Ensuring the resistance of prestressed structures before applying prestressing is crucial for worker safety in construction engineering. The structure must be capable of bearing loads beyond the widened elastic range provided by prestressing. This study aims to maximize the non-prestressed resistance capacity of the truss, setting this as a key objective within the MOO framework. This structural performance is calculated using Equation (10), which sums the indicator function δ_i for non-compliant elements and divides it by the total number of elements n.

$$f_1(\vec{X}) = \frac{1}{n}\sum_{i=1}^{n}\delta_i(\vec{X}) \tag{10}$$

Prestressing enables the truss structure to withstand the live loads applied to the structural assembly by introducing stresses that counteract the expected load effects. This study establishes a second objective function to maximize the effectiveness of prestressing in improving the truss's performance. This objective function specifically evaluates the structural performance in the prestressed state, ensuring that the prestressed elements can adequately support the anticipated live loads. The evaluation uses the approach outlined in Equation (11), which also employs the abovementioned indicator function δ_i.

$$f_2(\vec{X}) = \frac{1}{n}\sum_{i=1}^{n}\delta_i(\vec{X}) \tag{11}$$

The objective functions f_1 and f_2 use an indicator function δ that returns 1 for non-compliant elements and 0 for compliant ones. This shared mechanism, expressed through Equation (12), ensures consistency in evaluating compliance across different structural states within the OOP multi-objective algorithm deployment.

$$\delta_i(\vec{X}) = \begin{cases} 1 & \text{if the } i\text{-th element is non-compliant} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

A third objective function aims to minimize the truss weight, aligning with previous SOO applications for the structural typology under study. By reducing the structure's weight, this approach minimizes resource consumption, thereby lowering economic costs and enhancing the overall efficiency and sustainability of the prestressed arched truss design. This objective is computed using Equation (13) as the sum of the product of the length L_i and the linear mass m_i across the n elements in the structural configuration.

$$f_3(\vec{X}) = \sum_{i=1}^{n} L_i \cdot m_i(\vec{X}) \tag{13}$$

A fourth objective function targets the constructability of the prestressed arched truss by minimizing the number of different cross-sectional profiles used in the structure's elements. This objective focuses on improving construction efficiency and ease of assembly. This approach simplifies manufacturing and assembly processes by reducing the variety of cross-sectional profiles, reducing construction complexity and enhancing safety. This objective is calculated using Equation (14) as the sum of the unique cross-sectional profiles in the structural configuration, with the uniqueness of each profile determined by the indicator function U_i in Equation (15).

$$f_4(\vec{X}) = \sum_{i=1}^{n} U_i(\vec{X}) \tag{14}$$

$$U_i(\vec{X}) = \begin{cases} 1 & \text{if } p_i \notin (p_1, p_2, \ldots, p_{i-1}) \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

These four objective functions collectively provide a robust and innovative approach to optimizing prestressed arched truss designs, balancing performance, safety, and constructability. Including objectives that maximize structural performance in non-prestressed and prestressed states ensures the truss can withstand loads throughout its life cycle. Minimizing the structure's weight enhances resource efficiency and sustainability profile. Finally, focusing on constructability improves assembly processes. This MOO enhances operational performance and supports safety and viability constraints. This paper presents a novel contribution by integrating these diverse objectives into a cohesive optimization problem.

2.2. Optimization Algorithms

This section outlines a framework for multi-objective optimization that integrates the NSGA-III, CTAEA, and SMS-EMOA algorithms. In Section 2.2.1, the initial setup is introduced, which includes a repair operator common to all three methods, ensuring a uniform approach to managing solutions. Section 2.2.2 describes the execution phase for each of the NSGA-III, CTAEA, and SMS-EMOA algorithms. These algorithms utilize the SBX crossover with defined probabilities and distribution indices, along with polynomial mutation, also specified by probability and distribution indices. Figure 4 presents a flowchart of the standardized processes and methodologies within this framework, offering a detailed overview of the optimization workflow.

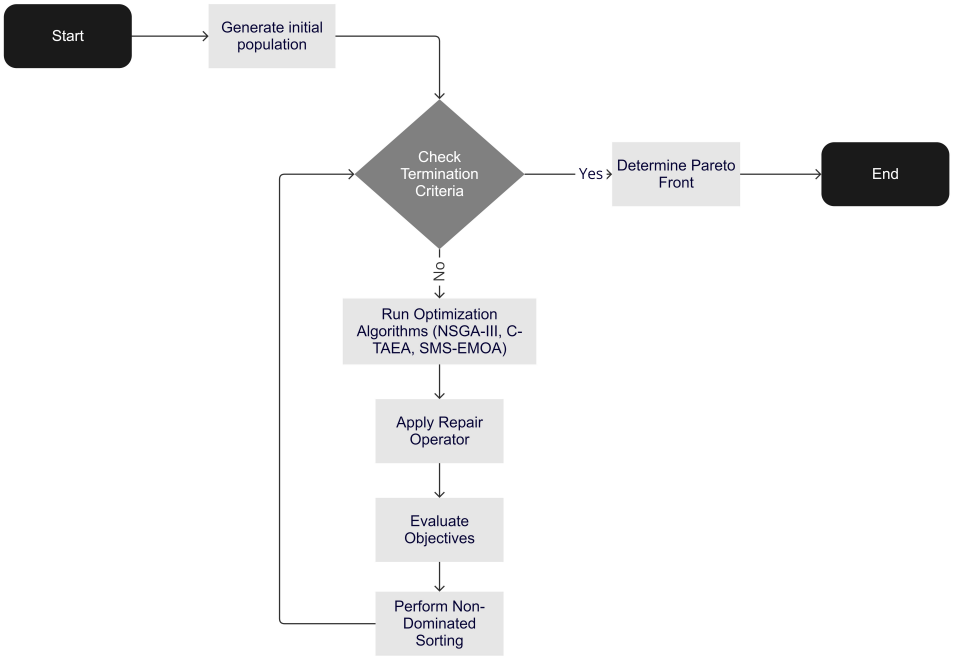

Figure 4. Diagram showing the optimization framework for NSGA-III, CTAEA, and SMS-EMOA algorithms.

2.2.1. Probabilistic Repair Operator

Algorithm 2 describes a repair operator that enhances solutions within an optimization algorithm by using statistical measures, such as the mean and median, alongside a probabilistic approach. During each iteration, a probability parameter β is used to decide which repair strategy will be applied. If a randomly generated number is less than β, the median is chosen for discrete variables (such as Choice and Integer), while the mean is applied to continuous (Real) variables. The median is particularly useful for discrete

variables as it mitigates the impact of outliers and reflects the most frequently occurring values in the dataset. Conversely, for continuous variables, the mean is used to calculate an average value, effectively capturing the central tendency within the population.

Algorithm 2 Statistical repair operator

Function StatRepair(S, β)
Input: S, β
Output: *repaired_S*
repaired_S ← copy of S
avg ← mean of S along axis 0
med ← median of S along axis 0
for each solution s in *repaired_S* **do**
  for each variable index i in s **do**
    type ← variable type at index i
    if rand $< \beta$ **then**
      if *type* is *Choice* or *Integer* **then**
        medValue ← med[i]
        if *type* is *Choice* **then**
          $s[i]$ ← closest value to *medValue* in *type.choice*
        else
          $s[i]$ ← round *medValue* to nearest integer within *type.bounds*
        end if
      else if *type* is *Real* **then**
        $s[i]$ ← clip avg[i] within *type.bounds*
      end if
    else
      Standard repair is applied based on variable type and bounds
    end if
  end for
end for
return *repaired_S*

2.2.2. Multi-Objective Optimization Algorithms

In the field of multi-objective optimization, different algorithms offer distinct advantages depending on the complexity and dimensionality of the problem. One of the most widely recognized algorithms is NSGA-II, known for its effectiveness in solving problems with two or three objectives by maintaining a balance between convergence and diversity of solutions. However, as the number of objectives increases, NSGA-II's performance diminishes due to its limited capability in maintaining diversity across a high-dimensional Pareto front. This limitation necessitates the exploration of more advanced algorithms capable of handling many-objective optimization problems, which are common in complex engineering designs like prestressed arch trusses.

When addressing the complex optimization challenges associated with the design of prestressed arch trusses, it is essential to select algorithms that can handle multiple, often conflicting objectives. The chosen algorithms must be capable of maintaining diversity across the solution space, efficiently handling constraints, and optimizing multiple objectives simultaneously. Different algorithms offer distinct advantages depending on the specific nature of the problem. For instance, some algorithms excel in managing many-objective problems by ensuring a well-distributed Pareto front, while others are more effective in handling constraints or maximizing specific performance metrics, such as the hypervolume.

In this study, three specific optimization algorithms—NSGA-III, CTAEA, and SMS-EMOA—were selected based on these considerations. NSGA-III was chosen for its ability to maintain diversity in high-dimensional objective spaces, CTAEA was selected for its efficiency in handling constrained optimization problems, and SMS-EMOA was chosen for its focus on maximizing the quality of the Pareto front through hypervolume optimization. Together, these algorithms offer a comprehensive and robust approach to solving the multi-objective optimization problem inherent in the design of prestressed arch trusses. The specific characteristics and advantages of each selected algorithm are detailed in the following sections.

To implement these optimization algorithms effectively, the selection of appropriate genetic operators is crucial. In this study, the simulated binary crossover (SBX) operator was employed as the crossover mechanism, complemented by polynomial mutation (PM). The SBX operator was chosen for its capability to generate diverse offspring by simulating the behavior of binary crossover within real-coded genetic algorithms. This operator is particularly beneficial in multi-objective optimization as it balances the need for exploration and exploitation, which is essential for achieving a well-distributed Pareto front. The parameters for SBX, including the crossover probability (*crossover_prob*) and crossover index (η), were carefully selected to optimize the algorithm's performance, especially in addressing the complex design requirements of prestressed arch trusses. The combination of SBX and PM ensured a thorough exploration of the solution space while maintaining diversity, a critical factor in handling the high-dimensional objective spaces characteristic of this study.

NSGA-III: As discussed in this section, it addresses these challenges through its incorporation of reference points and enhanced selection mechanisms, which allow it to manage a larger number of objectives effectively. As shown in Algorithm 3, NSGA-III combines selection, crossover, and mutation techniques. This algorithm uses simulated binary crossover (SBX) and polynomial mutation (PM) to generate a varied set of offspring, while predefined reference directions ensure a well-distributed array of solutions across multiple objectives. This method enhances NSGA-III's ability to effectively explore and exploit the multi-dimensional objective space, distinguishing it from the earlier NSGA-II. The selection strategy based on reference points is critical, as it directs the population toward a Pareto front that comprehensively represents the entire objective space, a key aspect in many-objective optimization. The algorithm's systematic approach, as detailed in the accompanying pseudocode, illustrates how it achieves a thorough and diverse set of solutions.

Algorithm 3 NSGA-III

Input: *pop_size* (population size), *ref_dirs* (reference directions), *mutation_prob* (mutation probability), *mutation_index* (mutation distribution index), *crossover_prob* (crossover probability), *crossover_index* (crossover distribution index)
Initialize population P with *pop_size* individuals
Evaluate initial population P
Compute reference points *ref_dirs*
Set generation counter $gen = 0$
while termination criteria not satisfied **do**
    Choose parent individuals from P through a tournament-based selection process
    Perform SBX crossover utilizing *crossover_prob* and *crossover_index*
    Apply polynomial mutation with *mutation_prob* and *mutation_index*
    Assess the offspring population Q
    Combine populations: $R = P \cup Q$
    Execute non-dominated sorting on R to generate fronts $F_1, F_2, \ldots$
    Choose the next generation population P based on reference point selection
    Update generation counter: $gen = gen + 1$
end while
return Population P that approximates the Pareto front

CTAEA: The CTAEA, as described in Algorithm 4, uses a two-archive approach to enhance the search process in many-objective optimization tasks. The algorithm employs adaptive differential evolution (ADE) for generating offspring and uses two separate archives to maintain convergence and diversity. This method ensures an effective exploration and exploitation of the objective space, facilitating the attainment of a well-distributed Pareto front. The detailed steps of the algorithm are encapsulated in the pseudocode provided.

Algorithm 4 CTAEA

Input: population size pop_size, mutation factor F, crossover probability CR
Initialize the population P with pop_size individuals
Set up the convergence archive CA and diversity archive DA
Assess the initial population P
Set the generation counter to $gen = 0$
while termination criteria are not fulfilled **do**
  Choose parents from P via tournament selection
  Apply ADE using the mutation factor F and crossover probability CR
  Evaluate the resulting offspring population Q
  Update the archives CA and DA with the offspring Q
  Merge archives and population: $R = P \cup CA \cup DA$
  Execute non-dominated sorting on R to form fronts $F_1, F_2, \ldots$
  Choose the next generation population P from R
  Increment $gen = gen + 1$
end while
return The population P that best approximates the Pareto front

SMS-EMOA: The SMS-EMOA, as described in Algorithm 5, incorporates selection, crossover, and mutation strategies to handle many-objective optimization tasks. This algorithm leverages SBX and PM for offspring generation and employs a hypervolume-based selection mechanism to guide the search towards the Pareto front. The hypervolume indicator helps maintain a balance between convergence and diversity, ensuring a well-represented set of solutions. The following pseudocode provides a detailed overview of the algorithm's steps.

Algorithm 5 SMS-EMOA

Input: population size pop_size, crossover probability $crossover\_prob$, mutation probability $mutation\_prob$, crossover distribution index $crossover\_index$, mutation distribution index $mutation\_index$
Create the initial population P with pop_size individuals
Assess the initial population P
Initialize the generation counter $gen = 0$
while termination criteria are not satisfied **do**
  Choose parents from P using tournament selection
  Apply SBX crossover with $crossover\_prob$ and $crossover\_index$
  Perform PM mutation with $mutation\_prob$ and $mutation\_index$
  Evaluate the offspring Q
  Merge populations: $R = P \cup Q$
  Conduct non-dominated sorting on R to generate fronts $F_1, F_2, \ldots$
  Choose the next generation population P using hypervolume-based selection
  Increment $gen = gen + 1$
end while
return The population P that best approximates the Pareto front

2.3. Evaluation and Decision-Making

Numerous factors beyond the scope of this paper can significantly influence decision-making outcomes when assessing performance and safety considerations. This study acknowledges these limitations and evaluates the effectiveness of different decision-making algorithms, identifying variations and similarities in their outcomes within the specific considerations outlined.

The MOO generates a set of non-dominated solutions representing well-balanced designs. This paper evaluates these efficient alternatives by objectively scoring and ranking them via decision-making strategies. The MCDM problem is structured with a decision matrix $X = r_{ij}$, comprising m alternatives $A_i = \{A_1, A_2, \ldots, A_m\}$ evaluated for n criteria. Each element r_{ij} of the decision matrix indicates the performance of alternative i for criterion j. This study employs the SAW, FUCA, TOPSIS, PROMETHEE, and VIKOR MCDM algorithms [31]. Criteria weighting, computed via an entropy theory-based strategy detailed in Section 2.3.1, serves as input for these algorithms, outlined in Section 2.3.2. A comparative analysis of the results from each strategy assesses their effectiveness in achieving robust and objective structural design selection.

2.3.1. Entropy Theory-Based Criteria Weighting

Entropy is a fundamental concept that quantifies a system's degree of disorder or randomness by measuring the number of possible microstates corresponding to a macrostate. The criteria weights W_j are calculated using an entropy-based approach [24], ensuring objectivity in the decision-making process and eliminating potential subjective biases. In this study, higher entropy levels indicate more uncertain information, affecting the decision-making algorithms' effectiveness.

The first step involves computing the normalized decision matrix X' via Equation (16) to level criteria with different units and orders of magnitude.

$$X' = r_{ij} \langle \sum_{i=1}^{m} r_{ij} \rangle^{-1} \tag{16}$$

After normalization, each criterion's entropy E_j and degree of divergence D_j are calculated through Equations (17) and (18).

$$E_j = \frac{-1}{\ln\langle m \rangle} \left\langle \sum_{i=1}^{m} r_{ij} \cdot \ln\langle r_{ij} \rangle \right\rangle \tag{17}$$

$$D_j = 1 - E_j \tag{18}$$

The final step is to compute W_j by normalizing D_j across all criteria per Equation (19). These objective criteria weights are then used as input for the decision-making algorithms in Section 2.3.2.

$$W_j = \frac{D_j}{\sum_{j=1}^{n} D_j} \tag{19}$$

2.3.2. Multi-Criteria Decision-Making Techniques

This study evaluated the efficacy of the NSGA-III, CTAEA, and SMS-EMOA optimization algorithms in identifying high-quality optima within the complex solution space, focusing on multi-stage structural performance, weight reduction, and constructability [14–16]. This section details the five decision-making techniques applied to the MCDM problem [25–29]. The cohesive evaluation of their outputs allows a robust assessment of the non-dominated solutions for similar mixed-integer programming (MIP) problems targeting sustainable infrastructure [31].

SAW: Algorithm 6 details the implementation of the SAW technique [25], which uses the decision matrix and entropy criteria weighting as inputs and is executed in three steps. First, the normalized decision matrix X' is computed, accounting for both maximization and minimization criteria intrinsic to the MOO problem in this study. Next, the algorithm calculates the S_i score for each alternative by summing the products of each element r_{ij} and the corresponding entropy weight W_j across all criteria. Finally, the alternatives are ranked in descending order based on their scores S.

Algorithm 6 SAW algorithm implementation

Function SAW(X, W)
Input: X, W
Output: S, *ranking*
$S \leftarrow$ vector of zeros with size n
$X' \leftarrow$ normalize X along axis 0
for each alternative i in n **do**
    $S_i \leftarrow 0$
    for each criterion j in m **do**
        $r_{ij} \leftarrow X'_{ij}$
        $S_i \leftarrow S_i + W_j \cdot r_{ij}$
    end for
end for
ranking $\leftarrow$ sort descending S
return S, *ranking*

FUCA: Algorithm 7 details the FUCA technique [26], which begins by sorting alternatives in descending order for maximization criteria (structural performance in non-prestressed and prestressed states) and in ascending order for minimization criteria (structure weight and constructability issues). The score S_i is then calculated by summing the products of the rankings R_{ij} and the entropy weights W_j for all criteria. Finally, the alternatives are sorted in ascending order based on the S scores, resulting in the final rankings for this technique.

Algorithm 7 FUCA algorithm implementation

Function FUCA(X, W)
Input: X, W
Output: S, *ranking*
$S \leftarrow$ vector of zeros with size n
for each alternative i in n **do**
    for each criterion j in m **do**
        $r_{ij} \leftarrow X_{ij}$
        $Q \leftarrow$ sort ascending X_j
        $R_{ij} \leftarrow r_{ij}$ index within Q
        $S_i \leftarrow S_i + W_j \cdot R_{ij}$
    end for
end for
ranking $\leftarrow$ sort descending S
return S, *ranking*

TOPSIS: Algorithm 8 describes the TOPSIS technique [27]. Initially, the normalized decision matrix X' and the normalized weighted matrix T_{ij} are computed, considering both maximization and minimization criteria. Next, the algorithm picks the ideal solution A^+ by selecting the highest values for maximization criteria, the lowest values for minimization criteria within T_{ij}, and the least favorable solution A^- by selecting the opposite. The nature of the TOPSIS technique involves calculating each alternative's proximity to the ideal solution and its distance from the least favorable one, computing each alternative's Euclidean distances D_i^+ to A^+ and D_i^- to A^-. The similarity index S_i is then evaluated as the distance ratio to the least favorable solution. Finally, the alternatives are sorted in descending order based on S.

Algorithm 8 TOPSIS algorithm implementation

Function TOPSIS(X, W)
Input: X, W
Output: S, *ranking*
$S \leftarrow$ vector of zeros with size n
$X' \leftarrow$ normalize X along axis 0
for each criterion j in m **do**
    $T'_j \leftarrow X'_j \cdot W_j$
end for
$A^+, A^- \leftarrow \max(T'_j)$ and $\min(T'_j)$ for each criterion j in m
$D^+, D^- \leftarrow$ vectors of zeros with size n
for each alternative i in n **do**
    for each criterion j in m **do**
        $D_i^+ \leftarrow D_i^+ + (T'_{ij} - A_j^+)^2$
        $D_i^- \leftarrow D_i^- + (T'_{ij} - A_j^-)^2$
    end for
    $D_i^+ \leftarrow (D_i^+)^{0.5}$ distance to D^+
    $D_i^- \leftarrow (D_i^-)^{0.5}$ distance to D^-
    $S_i \leftarrow D_i^- / (D_i^- + D_i^+)$
end for
ranking $\leftarrow$ sort descending S
return S, *ranking*

PROMETHEE: The PROMETHEE technique utilizes pairwise preference aggregation to convert the decision matrix into an explicit hierarchy. Algorithm 9 illustrates the PROMETHEE algorithm [28], which begins by constructing a preference matrix through pairwise comparisons of alternatives. The algorithm in this paper considers both maximization and minimization criteria (where higher and lower values are preferred, respectively) and integrates a preference function incorporating criteria weights W to quantify the preference degree between alternatives. The positive and negative flows, Φ^+ and Φ^-, are then computed from the preference matrix, representing how much each alternative is favored or opposed in all comparisons. The scoring S equivalent to the net flow Φ_i for each alternative assesses overall performance by offsetting its positive flow against its negative flow. Finally, the alternatives are sorted in descending order based on S.

Algorithm 9 PROMETHEE algorithm implementation

Function: PROMETHEE(X, W)
Input: X, W
Output: S, *ranking*
$S \leftarrow$ vector of zeros with size n
$F \leftarrow$ matrix of zeros with size $n \times n$
for each alternative i in n **do**
  for each alternative j in n **do**
    if $i \neq j$ **then**
      $P_{ij} \leftarrow \max(0, X_i - X_j) \cdot W$
      $F_{ij} \leftarrow \sum P_{ij}$
    end if
  end for
end for
$\Phi^+, \Phi^- \leftarrow$ vectors of zeros with size n
for each alternative i in n **do**
  for each alternative j in n **do**
    $\Phi_i^+ \leftarrow \Phi_i^+ + F_{ij} \cdot (n-1)^{-1}$ positive flow
    $\Phi_i^- \leftarrow \Phi_i^- + F_{ji} \cdot (n-1)^{-1}$ negative flow
  end for
  $\Phi_i \leftarrow \Phi_i^+ - \Phi_i^-$ net flow
  $S_i \leftarrow \Phi_i$
end for
ranking $\leftarrow$ sort descending S
return S, *ranking*

VIKOR: Algorithm 10 describes the VIKOR technique [29]. First, the algorithm calculates X' and identifies the best (f^+) and worst (f^-) performance values for each criterion, considering both maximization and minimization criteria. Two metrics are then evaluated for each alternative: the utility U_i, calculated as the weighted sum of the distances from an alternative to the optimal solution, normalized by the total range of criterion values; and the regret R_i, which identifies the maximum of these normalized distances to indicate the furthest deviation from the optimal in the worst-case scenario. The U_i and R_i metrics are then combined into a compound index S_i using the balance coefficient v (set to 0.5 in this study) to balance overall group utility against individual regret. Finally, the alternatives are sorted in ascending order based on S.

Algorithm 10 VIKOR algorithm implementation

Function VIKOR(X, W)
Input: X, W
Output: S, *ranking*
$U, R, S \leftarrow$ vectors of zeros with size n
$X' \leftarrow$ normalize X along axis 0
$f^+, f^- \leftarrow \max(X'_j)$ and $\min(X'_j)$ for each criterion j in m
for each alternative i in n **do**
  for each criterion j in m **do**
    $U_i \leftarrow U_i + W_j \cdot (f_j^+ - X'_{ij}) \cdot (f_j^+ - f_j^-)^{-1}$
    $R_i \leftarrow \max(R_i, W_j \cdot (f_j^+ - X'_{ij}) \cdot (f_j^+ - f_j^-)^{-1})$
  end for
end for
$U^* \leftarrow \min(U)$, $U^- \leftarrow \max(U)$
$R^* \leftarrow \min(R)$, $R^- \leftarrow \max(R)$
for each alternative i in n **do**
  $S_i \leftarrow v \cdot (U_i - U^*)/(U^- - U^*) + (1-v) \cdot (R_i - R^*)/(R^- - R^*)$
end for
ranking $\leftarrow$ sort ascending S
return S, *ranking*

3. Results

This section presents the results of the optimization using three algorithms: SMS-EMOA , NSGA-III, and CTAEA. In the subsection *Algorithm Comparisons* , the optimization values for four objective functions are detailed: structural capacity before tensioning, structural capacity after tensioning, structure weight, and constructability issues. Descriptive statistics are analyzed, and radar charts are presented, showing both individual solutions and the means and standard deviations for each algorithm.

The subsection *Performance Indicator Analysis* assesses the algorithms' effectiveness by employing two metrics: generational distance (GD) and inverted generational distance (IGD). These metrics measure how well the generated solutions converge towards and are distributed along a known Pareto front. Finally, in the subsection *Multi-Criteria Decision Analysis for Structural Design Optimization*, the non-dominated solutions are evaluated using a range of multi-criteria decision-making (MCDM) algorithms. This analysis compares the structural properties of the solutions, such as the cross-sectional areas of various elements and the corresponding prestressing forces.

3.1. Algorithm Comparisons

The CTAEA algorithm was utilized to define the probabilities for the crossover and mutation operators. The tuning process for the operators within the CTAEA algorithm was carried out systematically in two distinct phases, using the hypervolume metric as the main criterion for evaluation. In the first phase, the crossover probability was tested across various values—0.01, 0.1, 0.2, 0.3—to determine settings that could improve the optimization process. Similarly, the mutation probability was examined across a range of values—0.01, 0.02, 0.03, 0.04—to determine its impact on generating new solutions. During this exploratory phase, the crossover η value was fixed at five to maintain consistency in the distribution tightness of solutions.

Advancing to the second phase, a more focused exploitation strategy was employed, narrowing down the crossover probability to finer values—0.08, 0.1, 0.12—to optimize the frequency of crossover events. Concurrently, the mutation probability was refined to values—0.015, 0.02, 0.025—to further enhance solution diversity. This phase was essential for identifying the exact parameter configurations that optimize the hypervolume metric, achieving an ideal balance between the convergence of the Pareto front and the maintenance of diversity within the solution set.

This section presents the optimization results of the three algorithms: CTAEA, NSGA-III, and SMS-EMOA. Table 4 details the optimization values for each of the four objective functions: structural capacity before tensioning, structural capacity after tensioning, structure weight, and constructability issues. Subsequently, the descriptive statistics in Table 5 provide a statistical analysis to help understand the results. In Figure 5, two types of visualizations are presented: radar charts displaying all individual solutions and radar charts showing the means and standard deviations for each algorithm. The metrics of the solutions were normalized using a MinMax scale based on all solutions. Specifically, the structural capacity before tensioning (f_1) was transformed to a logarithmic scale, and the structure weight (f_3) and constructability issues (f_4) were converted to one minus their normalized values to reflect their minimization objectives. The axes in the radar charts represent the four objective functions: structural capacity before tensioning (f_1), structural capacity after tensioning (f_2), structure weight (f_3), and constructability issues (f_4). The shaded areas in the individual solution charts indicate the variability in the solutions, while the solid lines and shaded areas in the mean charts represent the averages and the ranges of one standard deviation, respectively.

Table 4. Multi-objective optimization results for different algorithms: CTAEA, NSGA-III, and SMS-EMOA.

Algorithm	f_1	f_2	f_3	f_4
CTAEA	8.519124×10^{13}	7.903836×10^{0}	3.932445×10^{1}	1.700000×10^{1}
	8.519124×10^{13}	1.698375×10^{1}	3.521146×10^{1}	1.600000×10^{1}
	8.519124×10^{13}	7.471966×10^{0}	3.042398×10^{1}	2.000000×10^{1}
	3.407360×10^{14}	7.601503×10^{0}	4.144611×10^{1}	1.500000×10^{1}
	4.097467×10^{1}	4.476063×10^{1}	3.344251×10^{1}	1.900000×10^{1}
	2.776707×10^{1}	6.558122×10^{0}	2.394779×10^{1}	1.600000×10^{1}
	4.256668×10^{13}	6.823006×10^{0}	2.789407×10^{1}	1.700000×10^{1}
	8.512460×10^{13}	7.917770×10^{0}	2.879679×10^{1}	1.900000×10^{1}
	1.735717×10^{1}	7.726929×10^{0}	2.642775×10^{1}	1.800000×10^{1}
NSGA-III	6.389343×10^{13}	1.028564×10^{1}	2.596566×10^{1}	1.800000×10^{1}
	1.277579×10^{14}	9.974873×10^{0}	3.878275×10^{1}	1.700000×10^{1}
	1.185833×10^{1}	7.033907×10^{0}	2.875980×10^{1}	1.600000×10^{1}
	1.703067×10^{14}	8.633605×10^{0}	2.942017×10^{1}	1.800000×10^{1}
	4.259562×10^{13}	3.994824×10^{1}	2.564713×10^{1}	1.800000×10^{1}
	4.256169×10^{13}	4.945458×10^{1}	3.212230×10^{1}	1.700000×10^{1}
	8.519124×10^{13}	6.481887×10^{0}	2.452360×10^{1}	1.700000×10^{1}
	3.748801×10^{1}	4.377684×10^{0}	2.388024×10^{1}	1.800000×10^{1}
	4.259562×10^{13}	7.576271×10^{0}	2.378082×10^{1}	2.000000×10^{1}
SMS-EMOA	2.042585×10^{1}	1.283917×10^{1}	3.122146×10^{1}	1.600000×10^{1}
	8.509959×10^{13}	8.575811×10^{1}	2.669033×10^{1}	1.800000×10^{1}
	1.276972×10^{14}	7.516954×10^{2}	3.039425×10^{1}	1.900000×10^{1}
	8.511064×10^{13}	9.880472×10^{0}	2.857700×10^{1}	1.700000×10^{1}
	6.388163×10^{13}	1.241284×10^{1}	3.123967×10^{1}	1.600000×10^{1}
	8.510857×10^{13}	8.374851×10^{0}	2.981695×10^{1}	1.400000×10^{1}
	8.518104×10^{13}	6.713652×10^{0}	2.921615×10^{1}	1.900000×10^{1}
	1.064536×10^{13}	2.236685×10^{1}	3.371193×10^{1}	1.500000×10^{1}
	8.516618×10^{13}	7.344048×10^{0}	2.452646×10^{1}	1.700000×10^{1}
	9.358560×10^{13}	9.039729×10^{0}	3.623062×10^{1}	1.600000×10^{1}

Table 5. Descriptive statistics of multi-objective optimization results for different algorithms: CTAEA, NSGA-III, and SMS-EMOA.

Algorithm	Metric	f_1	f_2	f_3	f_4
CTAEA	Mean	8.519124×10^{13}	1.421000×10^{1}	3.185000×10^{1}	1.733000×10^{1}
	Median	8.519124×10^{13}	7.600000×10^{0}	3.042000×10^{1}	1.700000×10^{1}
	Std Dev	1.130000×10^{14}	1.374000×10^{1}	6.160000×10^{0}	1.410000×10^{0}
	Min	1.740000×10^{1}	6.560000×10^{0}	2.395000×10^{1}	1.500000×10^{1}
	Max	3.407360×10^{14}	4.476000×10^{1}	4.145000×10^{1}	2.000000×10^{1}
NSGA-III	Mean	6.389343×10^{13}	1.731000×10^{1}	2.812000×10^{1}	1.778000×10^{1}
	Median	4.259562×10^{13}	9.970000×10^{0}	2.876000×10^{1}	1.800000×10^{1}
	Std Dev	5.420000×10^{13}	1.575000×10^{1}	7.770000×10^{0}	1.410000×10^{0}
	Min	1.185833×10^{1}	4.380000×10^{0}	2.378000×10^{1}	1.600000×10^{1}
	Max	1.703067×10^{14}	4.945000×10^{1}	3.878000×10^{1}	2.000000×10^{1}
SMS-EMOA	Mean	8.509959×10^{13}	9.160000×10^{1}	2.948000×10^{1}	1.650000×10^{1}
	Median	8.509959×10^{13}	9.880000×10^{0}	2.982000×10^{1}	1.600000×10^{1}
	Std Dev	1.140000×10^{14}	2.252200×10^{2}	2.730000×10^{0}	1.890000×10^{0}
	Min	2.042585×10^{1}	6.710000×10^{0}	2.453000×10^{1}	1.400000×10^{1}
	Max	1.276972×10^{14}	7.517000×10^{2}	3.623000×10^{1}	1.900000×10^{1}

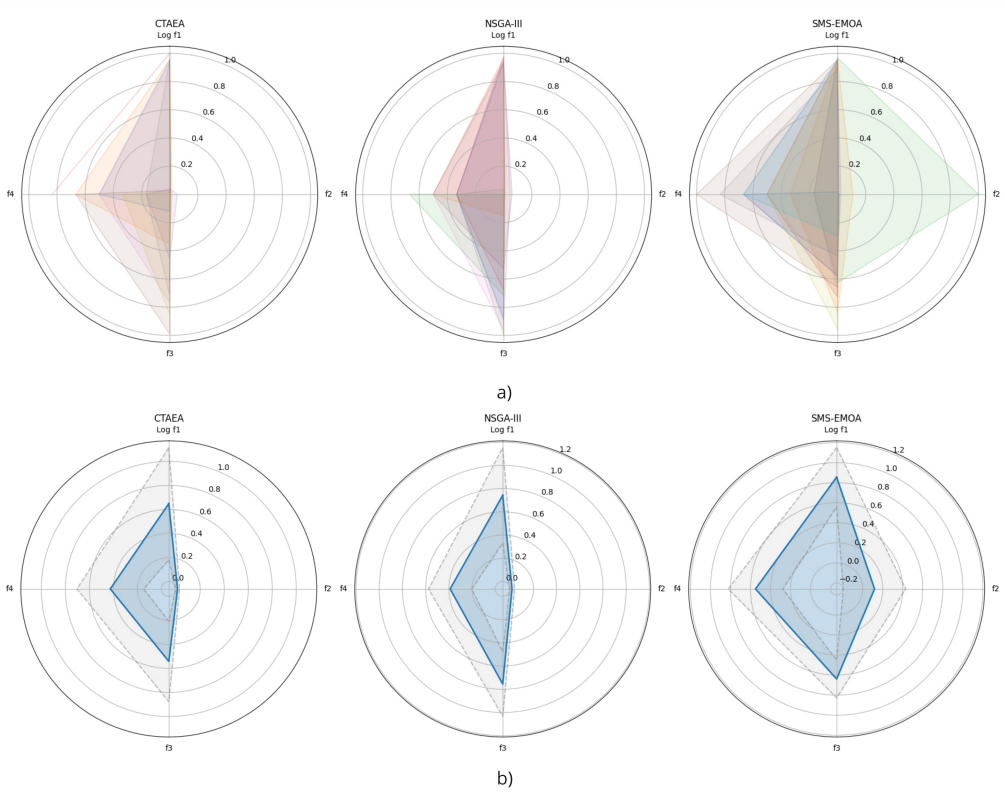

Figure 5. Radarchart visualization for the MOO results for each algorithm: (**a**) individual non-dominated solutions; and (**b**) mean and standard deviation values.

The CTAEA algorithm exhibits a high variability in structural capacity before tensioning, with a maximum value of 3.41×10^{14} and a significant standard deviation of 1.13×10^{14}. This suggests that CTAEA can generate solutions with a wide range of structural capacities. On the other hand, NSGA-III shows a lower mean (6.39×10^{13}) and a lower standard deviation (5.42×10^{13}), indicating more consistent solutions. SMS-EMOA, while having a similar mean (8.51×10^{13}), has a considerably high standard deviation (1.14×10^{14}), suggesting high variability in its solutions.

CTAEA presents significant variability in structural capacity after tensioning, with a standard deviation of 13.74 and a maximum value of 44.76. This indicates that some CTAEA solutions are significantly better after tensioning. NSGA-III, with a more balanced structural capacity after tensioning, has a mean of 17.31 and a standard deviation of 15.75, showing less variability. SMS-EMOA displays an extremely high mean (91.60) and a very large standard deviation (225.22), suggesting high variability and some exceptionally good solutions after tensioning.

In terms of structure weight, CTAEA has a mean of 31.85 with a standard deviation of 6.16, indicating relatively consistent solutions. NSGA-III shows a lower mean (28.12) and a standard deviation of 7.77, suggesting good consistency. SMS-EMOA, with a mean of 29.48 and a low standard deviation (2.73), also produces fairly consistent solutions.

Regarding constructability issues, CTAEA has a mean of 17.33 and a standard deviation of 1.41, indicating good consistency. NSGA-III is similar, with a mean of 17.78 and a standard deviation of 1.41, demonstrating consistent solutions. SMS-EMOA, with a lower mean (16.50) and a standard deviation of 1.89, indicates slightly less consistent solutions.

In conclusion, CTAEA produces solutions with high variability in structural capacity but consistent results in terms of weight and constructability issues. NSGA-III generates balanced and consistent solutions across all aspects. SMS-EMOA shows high variability in structural capacity after tensioning but consistent results in weight and constructability issues, which could be advantageous in specific scenarios where variability in structural capacity is acceptable or desirable.

From the radar charts and the analysis of the normalized data, Figure 5, it is observed that the CTAEA algorithm exhibits high variability in structural capacity before tensioning, suggesting it can generate solutions with a wide range of structural capacities. However, it shows consistency in terms of weight and constructability issues. The NSGA-III algorithm produces more balanced and consistent solutions across all metrics, which may be preferable for applications requiring stability. On the other hand, SMS-EMOA shows high variability in structural capacity after tensioning but generates consistent results in weight and constructability issues. Therefore, CTAEA is useful for exploring a wide range of structural capacities, NSGA-III offers balanced and consistent solutions, and SMS-EMOA could be beneficial in scenarios where variability in structural capacity is acceptable or desirable.

3.2. Performance Indicator Analysis

In this research, a systematic experimental setup was employed to assess the performance of three multi-objective optimization algorithms: CTAEA, NSGA-III, and SMS-EMOA. The experiment consisted of 30 independent runs, with each algorithm producing a varying number of points using a set of predefined parameters. This methodology enabled a direct comparison of the algorithms' effectiveness under uniform parameter conditions. To maintain consistency across different objective functions, a min-max normalization was applied to each function individually. This involved standardizing the units of the variables across all data points, including those on the Pareto front. For every experiment, the points generated by each algorithm were normalized by scaling their values between the minimum and maximum observed within the dataset. This normalization ensured that the varied scales were converted into a common metric, allowing for consistent comparisons across different optimization scenarios.

To assess the effectiveness of the optimization algorithms more thoroughly, two performance metrics were introduced: generational distance (GD) and inverted generational distance (IGD). These metrics evaluated the quality of the solutions produced by the algorithms, focusing on their convergence and diversity relative to a known Pareto front.

The **generational distance (GD)** measures the average Euclidean distance between each solution in a given set and the closest point on the Pareto front, providing an indication of how well the algorithm's solutions converge. In our experiment, the size of set A varied, representing the number of points obtained in each run. The GD is computed using the following equation:

$$GD(A) = \left(\frac{1}{|A|} \sum_{i=1}^{|A|} (d_i^2) \right)^{\frac{1}{2}}$$

where d_i represents the Euclidean distance between the ith solution a_i in set A and the closest point on the Pareto front Z. The sets A and Z are defined as follows:

$$A = \{a_1, a_2, \ldots, a_{|A|}\}, \quad Z = \{z_1, z_2, \ldots, z_{|Z|}\}$$

This metric is crucial for evaluating how close the generated solutions are to the ideal points on the Pareto front.

The **inverted generational distance (IGD)** is a metric that reverses the concept of generational distance by measuring the distance from each point in Z, the Pareto front, to the nearest point in A. In our experiment, set A included a variable number of points generated in each run. The IGD is computed using the following equation:

$$IGD(Z) = \left(\frac{1}{|Z|} \sum_{i=1}^{|Z|} (\hat{d}_i^2) \right)^{\frac{1}{2}}$$

where $\hat{d}_i$ represents the Euclidean distance (using $p = 2$) from a point z_i on the Pareto front to its closest reference point in set A. The set A is defined as $A = \{a_1, a_2, \ldots, a_{|A|}\}$, which

represents the solutions generated in each run. This metric is essential for assessing how well the generated solutions cover the Pareto front.

Advantages of each metric:

- The **GD** is especially valuable for evaluating how closely the generated solutions approach the Pareto front. A smaller GD value implies that the solutions produced by the algorithm are nearer to the optimal set, indicating better convergence performance.
- The **IGD** provides a comprehensive view of both the convergence and diversity of the solutions. It not only measures the proximity of the solutions to the Pareto front but also evaluates how evenly the solutions are spread across the front. A lower IGD value suggests that the solutions effectively cover the Pareto front with good distribution as well as proximity.

By utilizing both GD and IGD metrics, this study offers a thorough assessment of the algorithms' capability to produce solutions that are not only close to but also well spread across the ideal set of solutions, highlighting the subtle performance distinctions among the algorithms in multi-objective optimization tasks.

To assess the performance of the optimization algorithms, descriptive statistics were calculated to provide basic insights into the distribution of results for each algorithm. These statistics included the mean, maximum, minimum, standard deviation, and median values, which were key for understanding the central tendency and spread of the data. Additionally, the non-parametric Kruskal–Wallis test was applied to determine whether there were statistically significant differences between the algorithms. This test was selected due to its strength in dealing with non-normally distributed data and its suitability for comparing more than two groups without assuming equal variances, making it an ideal choice for the ordinal data produced in this experiment.

The results of the Kruskal–Wallis test, presented in Table 6, revealed statistically significant differences between the algorithms, with p-values of 2.50×10^{-15} for the generational distance (GD) and 5.15×10^{-06} for the inverted generational distance (IGD). These highly significant p-values indicate that the differences observed in the GD and IGD metrics were not due to random variation but reflected genuine performance discrepancies among the algorithms. Specifically, the GD results suggest that the proximity of solutions to the Pareto front varied significantly across the algorithms, while the IGD results highlight notable differences in how well the solutions covered the Pareto front. This statistical analysis underscores the distinct advantages and limitations of each algorithm in generating optimal solutions within the defined multi-objective optimization framework.

The combination of GD and IGD metrics provides a robust framework for evaluating multi-objective optimization algorithms. The statistical significance of the differences between the algorithms, as revealed by the Kruskal–Wallis test, confirms that these metrics effectively differentiate the performance of the algorithms. Specifically, NSGA-III demonstrated superior convergence with lower GD values, indicating its solutions were closer to the Pareto front. However, when considering the IGD metric, NSGA-III also showed a balanced performance in terms of both convergence and diversity, ensuring a good spread of solutions across the Pareto front. This comprehensive evaluation highlights NSGA-III's strengths in generating well-converged and diverse solutions, making it a robust choice for multi-objective optimization tasks.

Table 6. Results of the GD and IGD for different algorithms: CTAEA, NSGA-III, and SMS-EMOA.

Experiment	GD			IGD		
	CTAEA	**NSGA-III**	**SMS-EMOA**	**CTAEA**	**NSGA-III**	**SMS-EMOA**
1	0.343259	0.243163	0.562466	0.415568	0.334083	0.534358
2	0.344080	0.213318	0.590454	0.384573	0.357843	0.455668
3	0.259630	0.272583	0.486718	0.354933	0.436259	0.349575
4	0.332514	0.212139	0.454232	0.373068	0.292265	0.423739
5	0.354127	0.227103	0.521676	0.372890	0.357248	0.533556
6	0.309129	0.236694	0.314915	0.407252	0.419472	0.572944
7	0.393313	0.170852	0.597626	0.509682	0.398380	0.630672
8	0.421196	0.173322	0.306982	0.426333	0.332707	0.369995
9	0.268765	0.181077	0.504147	0.344514	0.339513	0.450241
10	0.284359	0.184755	0.448456	0.328917	0.308511	0.545038
11	0.286714	0.193047	0.364076	0.324218	0.283482	0.353318
12	0.260416	0.220492	0.421789	0.277172	0.306831	0.380599
13	0.322161	0.238490	0.456986	0.325745	0.316086	0.451070
14	0.400086	0.296006	0.393422	0.495041	0.392059	0.366927
15	0.283593	0.259695	0.469320	0.324543	0.362978	0.392514
16	0.351758	0.183666	0.326615	0.419597	0.287520	0.338755
17	0.331178	0.198670	0.359512	0.327612	0.286142	0.333001
18	0.334168	0.225676	0.396315	0.393625	0.349529	0.311955
19	0.322582	0.209546	0.342055	0.341323	0.284754	0.330602
20	0.309153	0.218193	0.345856	0.299896	0.284260	0.319379
21	0.362974	0.230623	0.561288	0.611660	0.378574	0.546087
22	0.264373	0.177711	0.424309	0.297895	0.294437	0.462823
23	0.348124	0.202473	0.379133	0.345787	0.344871	0.406306
24	0.242443	0.209284	0.362349	0.356406	0.262120	0.373122
25	0.274540	0.220155	0.461544	0.296070	0.304126	0.370325
26	0.344041	0.185727	0.461015	0.382598	0.284607	0.495314
27	0.386205	0.175720	0.297417	0.405136	0.312236	0.400061
28	0.322235	0.231037	0.481121	0.357127	0.326930	0.415491
29	0.382607	0.245432	0.348565	0.526730	0.340903	0.367277
30	0.353866	0.223426	0.643884	0.384533	0.317316	0.759258
Min	0.242443	0.170852	0.297417	0.277172	0.262120	0.311955
Max	0.421196	0.296006	0.643884	0.611660	0.436259	0.759258
Mean	0.326453	0.215336	0.436141	0.380348	0.329868	0.434666
Kruskal–Wallis *p*-value		2.50×10^{-15} **GD**			5.15×10^{-6} **IGD**	

3.3. Multi-Criteria Decision Analysis for Structural Design Optimization

This paper examined the performance of the CTAEA, NSGA-III, and SMS-EMOA multi-objective algorithms. Sections 3.1 and 3.2 thoroughly evaluate each algorithm's capability to generate balanced solutions that achieve equilibrium among conflicting objectives within a complex solution space. This section focuses on the results concerning the main structural characteristics of the non-dominated optimal solutions resulting from the MOO and the scoring and ranking of these balanced solutions using several decision-making algorithms detailed in Section 3.3.

Traditional and SOO design consider criteria either in isolation or sequentially. In contrast, the novel design approach in this study integrates multiple factors simultaneously within a framework integrating MOO and MCDM. Conventional prestressed arch truss design processes do not support this simultaneous consideration. The experimental results in this paper demonstrate the effectiveness of a multi-objective framework, enabling a more nuanced and comprehensive optimization of design parameters. This study seeks to move beyond the traditional and SOO approaches via a novel integrated methodology.

The cross-sectional area is a critical characteristic in structural analysis as it directly influences the load-bearing capacity and stability of the arched truss. Table 7 presents the

average cross-sectional values for the upper chord (elements 1–5 and 16–20), lower chord (elements 6–8 and 21–23), lattice elements (elements 9–15 and 24–29), and the tie member (element 30), along with the associated prestressing force for the non-dominated solutions.

Table 7. Element cross-sectional area results for different algorithms: A_1 to A_9 for CTAEA; A_{10} to A_{18} for NSGA-III; and A_{19} to A_{28} for SMS-EMOA.

Alternative	Prest. Force (kN)	Cable Tie (cm^2)	Upper Chord (cm^2)	Lower Chord (cm^2)	Lattice Elem. (cm^2)
A_1	586.71	18.37	34.08	39.26	33.43
A_2	596.43	13.81	22.21	34.66	27.52
A_3	631.25	11.74	25.11	34.81	32.84
A_4	636.02	9.87	26.58	34.10	33.76
A_5	565.26	18.37	40.37	28.40	33.29
A_6	444.86	11.74	25.17	38.28	25.17
A_7	254.70	18.37	35.31	36.12	23.18
A_8	397.66	9.87	21.35	32.29	28.74
A_9	916.31	13.81	42.99	57.95	32.80
A_{10}	370.60	9.87	26.55	34.27	26.75
A_{11}	476.54	16.01	37.56	29.96	28.79
A_{12}	370.84	16.01	25.01	21.42	33.70
A_{13}	575.16	13.81	28.56	34.78	31.31
A_{14}	597.28	9.87	39.12	26.21	25.47
A_{15}	583.16	16.01	24.48	16.60	28.42
A_{16}	484.74	16.01	21.09	39.26	38.10
A_{17}	995.20	8.15	19.27	45.23	34.63
A_{18}	637.81	20.9	35.61	18.84	33.46
A_{19}	349.80	6.61	28.71	39.48	45.42
A_{20}	575.16	13.81	34.11	45.64	33.77
A_{21}	992.25	13.81	25.71	32.04	31.68
A_{22}	916.78	8.15	28.90	27.29	29.09
A_{23}	725.01	8.15	36.41	30.35	28.57
A_{24}	617.31	16.01	35.10	30.13	28.49
A_{25}	446.36	16.01	32.10	32.74	27.46
A_{26}	825.72	11.74	25.38	21.34	35.40
A_{27}	630.45	9.87	28.69	36.85	27.91
A_{28}	674.55	18.37	38.08	35.10	27.26

Figure 6 illustrates the interdependencies in structural element sizing and the design implications by comparing the cross-sectional areas (in cm^2) of the upper and lower chords, lattice elements, and cable ties. Each dot in the scatter plots represents a specific design configuration obtained through MOO, with the trend line highlighting the overall relationship between the cross-sectional areas of the elements being compared. The shaded regions in the diagonal plots (the $n \times n$ positions) indicate the range and distribution of these configurations, enabling an assessment of how various sizing decisions influence the overall truss design.

The results indicated a slight positive correlation between the upper and lower chords, suggesting an interdependent element sizing to ensure balanced stiffness and strength. Lattice elements showed a positive correlation with the lower chord but a negative correlation with the upper chord, indicating that lattice elements were scaled more closely with the lower chord to maintain stability while allowing flexibility in the upper chord's design. Cable tie elements exhibited a positive correlation with the upper chord but a slight negative correlation with the lower chord and lattice elements, highlighting their specialized role in accommodating prestressing requirements within the lower section of the prestressed arched truss.

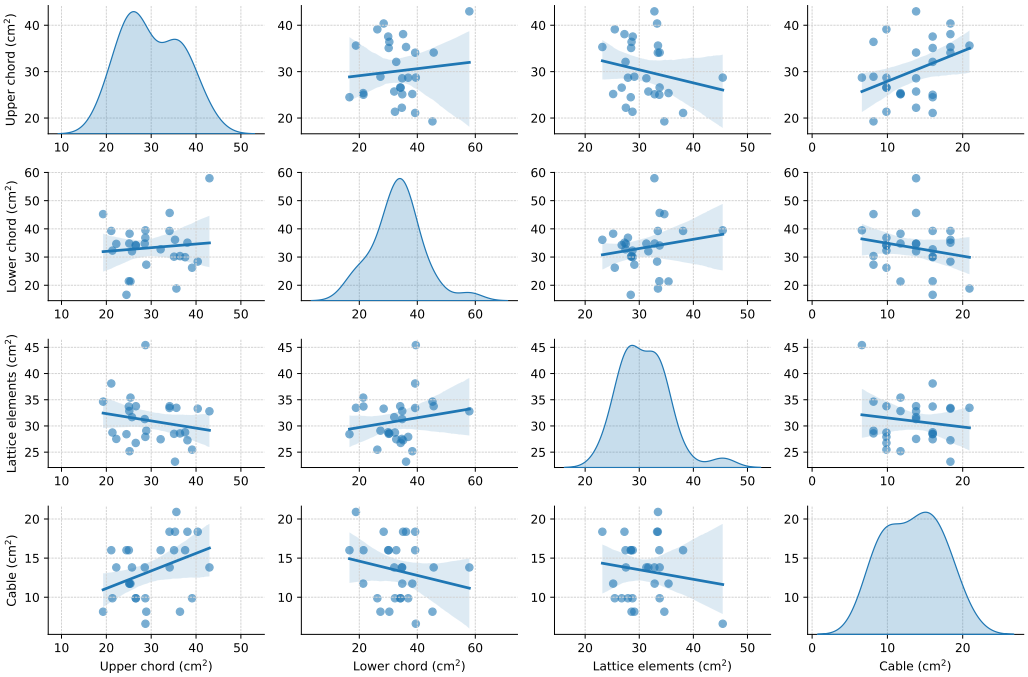

Figure 6. Non-dominated element sizing pair-plot interdependence analysis, with row and column positions: (1) upper chord; (2) lower chord; (3) lattice elements; and (4) cable tie member.

Figure 7 presents the distribution of prestressing forces (in kN) for the non-dominated optimal solutions achieved by the CTAEA, NSGA-III, and SMS-EMOA algorithms. Each violin plot represents the distribution and emphasizes the mean and median values for each MOO algorithm. The results for the CTAEA algorithm showed a relatively narrow distribution around the median value, indicating a consistent generation of solutions with similar prestressing force selection. In contrast, the NSGA-III algorithm displayed a slightly higher mean and median, suggesting a tight clustering of solutions around these slightly greater forces.

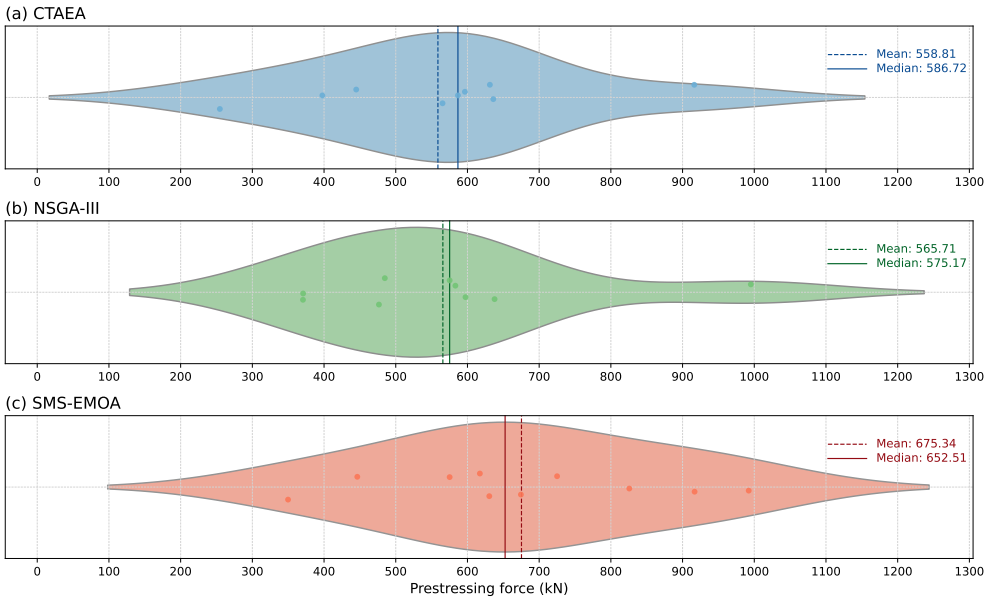

Figure 7. Non-dominated prestressing force violin plot: (**a**) CTAEA; (**b**) NSGA-III; and (**c**) SMS-EMOA.

The SMS-EMOA algorithm, however, exhibited a significantly higher mean prestressing force with a more dispersed distribution. This indicated that while SMS-EMOA could produce solutions with higher prestressing forces, there was more significant variability in the results, potentially offering a wider range of balanced solutions but with less predictability.

Figure 8 illustrates the effect of the prestressing force on the objective function results across all non-dominated solutions. Each subplot illustrates how an objective function varies with the prestressing force, with dots representing non-dominated solution data, lines indicating trends, and shaded areas representing variability through confidence intervals.

The prestressing force slightly enhanced the non-prestressed structural capacity f_1, potentially due to higher sizing requirements for greater prestressing forces. It did not significantly affect the structure's weight f_3. However, it slightly increased constructability issues f_4, presumably due to the need for larger sections in specific areas, such as the upper chord. Notably, a positive correlation existed between prestressing force and prestressed-state resistance, mainly concentrated around the mid-range of 600 kN. This behavior corresponded with the enlarged elastic range attained with greater prestressing. This was particularly evident in the A_{21} solution achieved by the SMS-EMOA algorithm.

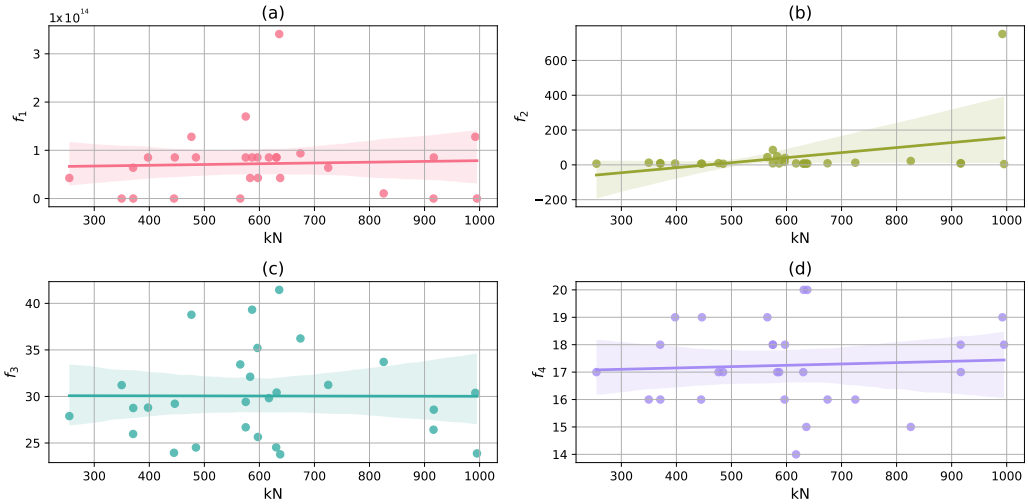

Figure 8. Prestressing force influence on objective function values across all non-dominated solutions: (**a**) f_1; (**b**) f_2; (**c**) f_3; and (**d**) f_4.

Table 8 presents the results for the MCDM problem, including criteria weighting via an entropy theory-based approach, normalized objective functions, and the corresponding scoring and ranking results for the non-dominated solutions. The scoring and ranking were conducted using the SAW, FUCA, TOPSIS, PROMETHEE, and VIKOR techniques for the balanced designs attained by the CTAEA, NSGA-III, and SMS-EMOA algorithms.

The criteria weighting, calculated via entropy theory, objectively assesses the importance of each criterion in the multi-criteria decision-making process, minimizing subjective biases for a balanced evaluation. The results showed higher weights for the structural capacity objectives. The prestressed state structural capacity f_2 was identified as the primary source of information within the MCDM process, followed by the non-prestressed state f_1. Constructability issues f_4 and structural self-weight f_3 received moderate and very similar weightings.

The scoring and ranking results using various MCDM techniques indicated a multi-faceted scenario where no algorithm consistently outperformed the others, highlighting the complexity of balancing conflicting objectives. Figure 9 represents the alternative ranking for each technique, clearly illustrating how each algorithm shows strengths in specific non-dominated solutions but may lag in others. This reflects the trade-offs inherent in the MOO problem in this paper. The radar chart compares the performance of the design alternatives (A_1 to A_{28}) across the five decision-making techniques. Each axis represents

one of the alternatives, and each line shows how a specific MCDM technique ranks across all non-dominated solutions.

Table 8. Multi-criteria decision-making for structural optimization results for different algorithms: A_1 to A_9 for CTAEA; A_{10} to A_{18} for NSGA-III; and A_{19} to A_{28} for SMS-EMOA.

	f_1	f_2	f_3	f_4	**Scores**					**Ranks**				
W_j	**0.2292**	**0.4840**	**0.1585**	**0.1282**	S_i^S	S_i^F	S_i^T	S_i^P	S_i^V	R_i^S	R_i^F	R_i^T	R_i^P	R_i^V
A_1	0.2500	0.0047	0.0000	0.5000	0.012916	16.85074	0.071882	–0.00763	0.918520	21	21	21	21	21
A_2	0.2500	0.0168	0.8338	0.3333	0.016630	10.60937	0.075152	–0.00730	0.744238	20	20	4	4	20
A_3	0.2500	0.0041	0.7061	1.0000	0.012738	18.92702	0.074052	–0.01265	0.931583	4	2	13	20	24
A_4	1.0000	0.0043	1.0000	0.1666	0.041782	14.96239	0.239817	+0.10735	0.610298	15	14	20	13	4
A_5	0.0000	0.0540	0.7866	0.8333	0.018321	14.61366	0.053725	–0.02510	1.000000	13	15	11	11	14
A_6	0.0000	0.0029	0.5335	0.3333	0.002706	20.27396	0.035312	–0.06431	0.669736	14	10	28	15	15
A_7	0.1249	0.0032	0.6387	0.5000	0.007641	20.55176	0.045388	–0.03935	0.748575	11	22	27	2	27
A_8	0.2498	0.0047	0.6628	0.8333	0.012914	16.34708	0.074960	–0.01550	0.830591	5	26	16	1	16
A_9	0.0000	0.0044	0.5996	0.6666	0.003181	19.31930	0.029751	–0.05810	0.827828	2	28	24	28	6
A_{10}	0.1875	0.0079	0.5873	0.6666	0.011475	12.51064	0.062053	–0.02791	0.731420	28	11	22	3	13
A_{11}	0.3749	0.0074	0.9290	0.5000	0.018591	12.77009	0.104729	+0.01249	0.857182	22	13	2	8	22
A_{12}	0.0000	0.0035	0.6618	0.3333	0.002899	21.32036	0.027050	–0.05908	0.760730	24	23	8	25	26
A_{13}	0.4998	0.0056	0.6794	0.6666	0.022870	13.01932	0.136464	+0.02280	0.699700	1	24	25	22	23
A_{14}	0.1250	0.0475	0.5788	0.6666	0.021188	11.26051	0.066044	–0.01835	0.613945	8	19	3	14	10
A_{15}	0.1249	0.0603	0.7514	0.5000	0.02507	11.84549	0.071162	–0.00695	0.638484	3	5	1	24	2
A_{16}	0.2500	0.0028	0.5488	0.5000	0.012337	17.20333	0.077831	–0.02368	0.644327	27	4	15	27	7
A_{17}	0.0000	0.0000	0.5317	0.6666	0.001813	22.39345	0.034005	–0.06291	0.781043	25	27	14	16	12
A_{18}	0.1250	0.0042	0.5290	1.0000	0.007951	18.14143	0.049548	–0.03868	0.847478	23	8	10	23	17
A_{19}	0.0000	0.0113	0.7274	0.3333	0.005272	14.58186	0.025158	–0.05275	0.804788	16	1	23	5	28
A_{20}	0.2497	0.1088	0.6066	0.6666	0.044740	9.110490	0.130015	+0.03217	0.387529	10	16	5	10	19
A_{21}	0.3747	1.0000	0.7053	0.8333	0.321856	7.758039	0.836178	+0.49541	0.000000	26	18	18	18	9
A_{22}	0.2497	0.0073	0.6569	0.5000	0.013717	12.52747	0.075759	–0.01734	0.706069	18	3	7	7	8
A_{23}	0.1874	0.0107	0.7279	0.3333	0.012344	13.03355	0.059425	–0.02403	0.710294	7	9	6	26	25
A_{24}	0.2497	0.0053	0.6900	0.0000	0.013104	13.84709	0.076015	–0.02142	0.568877	19	6	17	19	18
A_{25}	0.2499	0.0031	0.6739	0.8333	0.012428	20.66047	0.074604	–0.01584	0.843857	9	7	26	9	11
A_{26}	0.0312	0.0240	0.7938	0.1666	0.010376	12.56141	0.032106	–0.04055	0.709243	12	25	9	12	1
A_{27}	0.2499	0.0039	0.5489	0.5000	0.012687	16.34379	0.077928	–0.02312	0.640559	6	12	12	17	3
A_{28}	0.2746	0.0062	0.8609	0.3333	0.014334	12.62740	0.079368	–0.00769	0.790756	17	17	19	6	5

Alternatives A_7 and A_8 from CTAEA stood out due to their performance in the PROMETHEE method, securing the top and second ranks, respectively. Additionally, A_8 remained within the top five for the SAW technique but was poorly ranked by FUCA and exhibited mid-range performance in both TOPSIS and VIKOR. This variability indicated context-dependent alternatives with strengths for specific techniques and limitations when evaluated by other methods.

In contrast, some CTAEA alternatives, such as A_1 and A_5, displayed more consistent rankings across different techniques. A_1 consistently appeared in the lower rankings, indicating a general lack of competitiveness across all criteria. On the other hand, A_5 maintained mid-tier positions, reflecting balanced but non-exceptional performance across the board. This consistency could indicate non-dominated solutions with characteristics advantageous in decision contexts that value stability and predictability over specialization.

The NSGA-III algorithm demonstrated notable performance, particularly with alternative A_{15}, which ranked in the top five across four out of five techniques, including securing the top rank for TOPSIS. This non-dominated solution exhibited significant robustness and versatility across all techniques, making it a strong candidate under diverse evaluation methodologies. However, FUCA ranked A_{15} lower, favoring other alternatives such as CTAEA's designs. This discrepancy suggested that while the NSGA-III-generated solution excelled in most contexts, its suitability may be limited using FUCA's algorithmic approach.

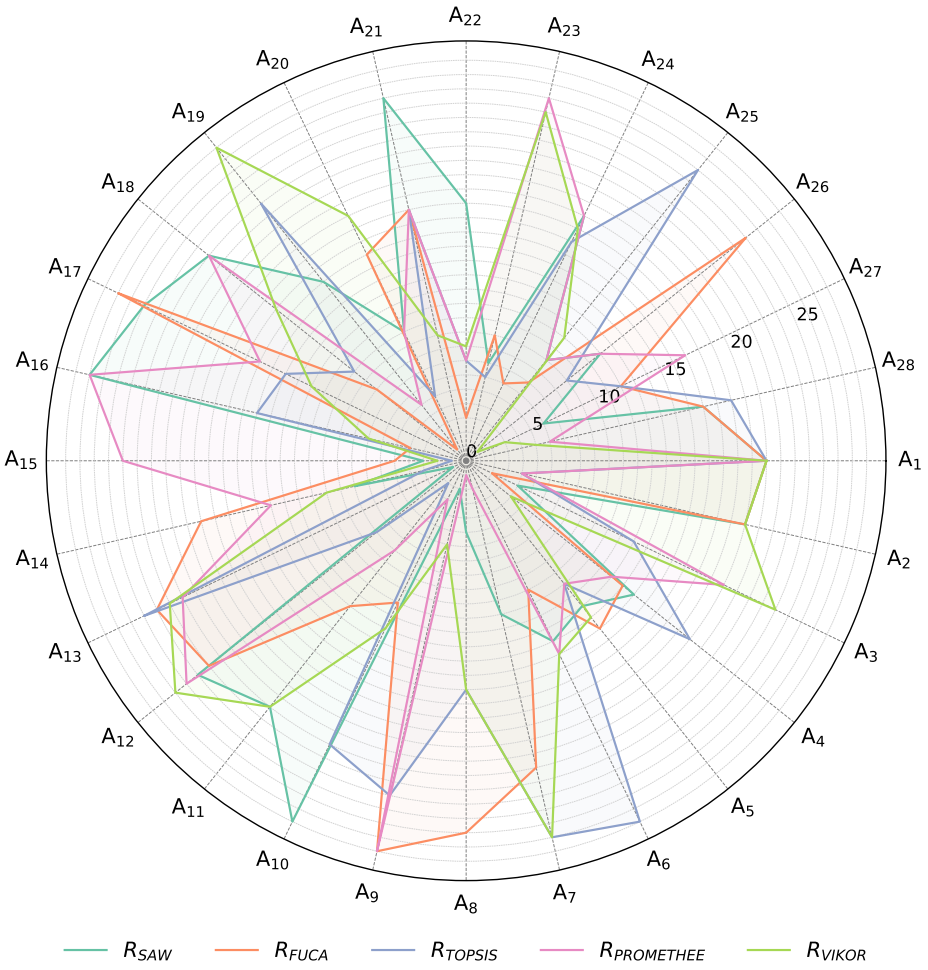

Figure 9. Alternative ranking radar chart for all non-dominated solutions across the MCDM techniques.

SMS-EMOA alternatives presented a diverse range of rankings. Alternative A_{19} was ranked highest by FUCA and performed well in PROMETHEE, yet it was poorly ranked by VIKOR, indicating a solid alignment with specific techniques but a lack of general transferability. Similarly, A_{26} showed strong performance with a top rank in VIKOR but mid and low rankings across other techniques, suggesting its strengths were evaluation-specific.

Across algorithms, NSGA-III and SMS-EMOA alternatives generally achieved higher ranks more consistently than CTAEA alternatives. NSGA-III's and SMS-EMOA's non-dominated alternatives frequently appeared in higher positions, indicating their robustness and adaptability. The stark contrast in rankings for some alternatives, such as CTAEA's A_8 and SMS-EMOA's A_{19}, underscores the importance of selecting and applying multiple MCDM techniques and evaluating top-ranked alternatives that better align with stakeholders' specific decision criteria and objectives for the MOO problem posed in this paper.

4. Conclusions

This study focused on the structural design optimization of a prestressed arched truss using a combination of multi-objective algorithms and decision-making techniques. The research integrated NSGA-III, CTAEA, and SMS-EMOA algorithms to optimize multi-state structural performance, weight, and constructability. The non-dominated optimal designs were evaluated and ranked using the MCDM techniques SAW, FUCA, TOPSIS, PROMETHEE, and VIKOR. The analysis of the results led to the following conclusions:

- The performance of the optimization algorithms was comprehensively evaluated using generational distance (GD) and inverted generational distance (IGD) metrics. NSGA-III demonstrated superior convergence, indicated by lower GD values, suggesting that its solutions were closer to the Pareto front. Additionally, NSGA-III showed a balanced performance in terms of diversity and convergence, as evidenced by lower IGD values, ensuring a good spread of solutions across the Pareto front.
- The element sizing results for the non-dominated designs indicate that the upper and lower chords, while interdependent, allow for targeted adjustments to meet specific structural demands.
- The profile selection in the upper and lower chords significantly influences the lattice elements, with the lower chord primarily affecting their sizing. Modifying the lattice design for specific applications can impact stress distribution and element sizing. Implementing topological optimization within the MOO process presents an avenue for subsequent research projects.
- Cable ties' sizing directly correlates with the upper chord and negatively correlates with the lower chord and lattice elements, highlighting their role in sustaining and distributing prestressing forces across the structure.
- A multifaceted approach involving multiple MCDM techniques is crucial for capturing a comprehensive view of each alternative's potential, balancing the strengths and weaknesses identified by each method to achieve the most suitable solution for structural design optimization.
- The selection of an alternative should be guided by the specific conditions of the decision-making process, using insights from multiple MCDM techniques to ensure a well-rounded and informed choice that aligns with stakeholders' specific interests and project objectives.

This study provides a comprehensive framework for optimizing prestressed arched truss designs, balancing multiple objectives through advanced optimization algorithms and robust decision-making techniques. A comprehensive comparison of the outcomes for the MOO and MCDM framework with traditional approaches is a relevant avenue for subsequent research projects requiring an extensive analysis beyond the scope of the present study. Future research should target this comparison and explore incorporating criteria such as environmental impact and life-cycle cost into the optimization and decision-making process to enhance the framework's scope and applicability within sustainable development. Moreover, adapting and deploying additional MOO and MCDM techniques could provide deeper insights and drive advancements in the optimization framework, contributing to its ongoing development and refinement.

Author Contributions: Conceptualization, A.R.-V., J.G. and G.P.; methodology, A.R.-V., J.G., G.P., J.A. and V.Y.; software, A.R-V. and J.G.; validation, A.R.-V., J.G., G.P., J.A. and V.Y.; formal analysis, A.R.-V.; investigation, A.R.-V. and J.G.; resources, A.R.-V., J.A. and V.Y.; data curation, A.R.-V.; writing—original draft preparation, A.R.-V.; writing—review and editing, J.G., J.A. and V.Y.; visualization, A.R.-V.; supervision, J.A. and V.Y.; project administration, V.Y.; funding acquisition, V.Y. All authors have read and agreed to the published version of the manuscript.

Funding: Grant PID2023-150003OB-I00 funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe". José García is supported by VINCI-DI:039.463/2024.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All the data used in the research can be found in the article.

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

Abbreviations

The following abbreviations are used in this manuscript:

SOO	Single-objective optimization
MOO	Multi-objective optimization
MCDM	Multi-criteria decision-making
NSGA-III	Non-dominated sorting genetic algorithm III
CTAEA	Two-archive evolutionary algorithm
SMS-EMOA	Hypervolume measure–evolutionary multi-objective optimization algorithm
SAW	Simple additive weighting
FUCA	Fair un choix adéquat
SLCA	Social life-cycle analysis
TOPSIS	Technique for order of preference by similarity to ideal solution
PROMETHEE	Preference ranking organization method for enrichment evaluation
VIKOR	Visekriterijumska optimizacija i kompromisno resenje
OOP	Object-oriented programming
M-SCV	Multi-state constraint verification
SBX	Simulated binary crossover
PM	Polynomial mutation
ADE	Adaptive differential evolution
MIP	Mixed-integer problem
GD	Generational distance
IGD	Inverted generational distance

References

1. Yepes, V.; Medina, J. Economic heuristic optimization for heterogeneous fleet VRPHESTW. *J. Transp. Eng.* **2006**, *132*, 303–311. [CrossRef]
2. Zhang, Z.; Cheng, X.; Xing, Z.; Gui, X. Pareto multi-objective optimization of metro train energy-saving operation using improved NSGA-II algorithms. *Chaos Solitons Fractals* **2023**, *176*, 114183. [CrossRef]
3. Rastegaran, M.; Beheshti Aval, S.; Sangalaki, E. Multi-objective reliability-based seismic performance design optimization of SMRFs considering various sources of uncertainty. *Eng. Struct.* **2022**, *261*, 114219. [CrossRef]
4. Blank, J.; Deb, K. pymoo: Multi-Objective Optimization in Python. *IEEE Access* **2020**, *8*, 89497–89509. [CrossRef]
5. Kabadayi, N.; Dehghanimohammadabadi, M. Multi-objective supplier selection process: A simulation–optimization framework integrated with MCDM. *Ann. Oper. Res.* **2022**, *319*, 1607–1629. [CrossRef]
6. Navarro, I.J.; Yepes, V.; Martí, J.V. A Review of Multicriteria Assessment Techniques Applied to Sustainable Infrastructure Design. *Adv. Civ. Eng.* **2019**, *2019*, 6134803. [CrossRef]
7. Ferdous, J.; Bensebaa, F.; Milani, A.S.; Hewage, K.; Bhowmik, P.; Pelletier, N. Development of a Generic Decision Tree for the Integration of Multi-Criteria Decision-Making (MCDM) and Multi-Objective Optimization (MOO) Methods under Uncertainty to Facilitate Sustainability Assessment: A Methodical Review. *Sustainability* **2024**, *16*, 2684. [CrossRef]
8. Chan, M.; Jin, H.; van Kan, D.; Vrcelj, Z. Developing an innovative assessment framework for sustainable infrastructure development. *J. Clean. Prod.* **2022**, *368*, 133185. [CrossRef]
9. Elkhayat, Y.; Mohamed, B.; Zayat, M.E.; Marzouk, M. Bibliometric analysis and visualization of sustainable infrastructure. *Innov. Infrastruct. Solut.* **2024**, *9*, 14. [CrossRef]
10. Rivero-Iglesias, J.M.; Puente, J.; Fernandez, I.; León, O. Integrated model for the assessment of power generation alternatives through analytic hierarchy process and a fuzzy inference system. Case study of Spain. *Renew. Energy* **2023**, *211*, 563–581. [CrossRef]
11. Belletti, B.; Gasperi, A. Behavior of Prestressed Steel Beams. *J. Struct. Eng.* **2010**, *136*, 1131–1139. [CrossRef]
12. Afshan, S.; Theofanous, M.; Wang, J.; Gkantou, M.; Gardner, L. Testing, numerical simulation and design of prestressed high strength steel arched trusses. *Eng. Struct.* **2019**, *183*, 510–522. [CrossRef]
13. Segal, E.M.; Rhode-Barbarigos, L.; Adriaenssens, S.; Filomeno Coelho, R.D. Multi-objective optimization of polyester-rope and steel-rope suspended footbridges. *Eng. Struct.* **2015**, *99*, 559–567. [CrossRef]
14. Hadjipantelis, N.; Gardner, L.; Wadee, M.A. Prestressed cold-formed steel beams: Concept and mechanical behaviour. *Eng. Struct.* **2018**, *172*, 1057–1072. [CrossRef]
15. Gesualdo, F.A.R.; Cunha, T.A.; Rezende, R.B. Numerical and experimental evaluation of a double inverted trussed beam reinforced with steel cable. *Constr. Build. Mater.* **2014**, *50*, 736–743. [CrossRef]
16. Wang, J.; Afshan, S.; Gardner, L. Axial behaviour of prestressed high strength steel tubular members. *J. Constr. Steel Res.* **2017**, *133*, 547–563. [CrossRef]
17. Keleş, M.; Artar, M.; Ergün, M. Investigation of temperature effect on the optimal weight design of steel truss bridges using Cuckoo Search Algorithm. *Structures* **2024**, *59*, 105819. [CrossRef]

18. Quan, G.; Ye, J.; Xu, N.; Qi, J.; Zhang, Z.; Wu, H.; Gao, B. Behaviour and design of prefabricated connections under combined bending and compression for free-form grid structures. *Structures* **2022**, *41*, 1763–1780. [CrossRef]
19. Partskhaladze, G.; Alcala, J.; Medzmariashvili, E.; Chavleshvili, G.; Surguladze, B.; Yepes, V. Heuristic Optimization of a New Type of Prestressed Arched Truss. *Materials* **2022**, *15*, 8144. [CrossRef] [PubMed]
20. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. [CrossRef]
21. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
22. Li, K.; Chen, R.; Fu, G.; Yao, X. Two-Archive Evolutionary Algorithm for Constrained Multiobjective Optimization. *IEEE Trans. Evol. Comput.* **2019**, *23*, 303–315. [CrossRef]
23. Beume, N.; Naujoks, B.; Emmerich, M. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *Eur. J. Oper. Res.* **2007**, *181*, 1653–1669. [CrossRef]
24. Zeleny, M. *Multiple Criteria Decision Making*; McGraw-Hill: New York, NY, USA, 1982.
25. Churchman, C.W.; Ackoff, R.L. An Approximate Measure of Value. *J. Oper. Res. Soc. Am.* **1954**, *2*, 172–187. [CrossRef]
26. Zakeri, S.; Chatterjee, P.; Konstantas, D.; Ecer, F. A comparative analysis of simple ranking process and faire un Choix Adéquat method. *Decis. Anal. J.* **2024**, *10*, 100380. [CrossRef]
27. Hwang, C.L.; Lai, Y.J.; Liu, T.Y. A new approach for multiple objective decision making. *Comput. Oper. Res.* **1993**, *20*, 889–899. [CrossRef]
28. Behzadian, M.; Kazemzadeh, R.; Albadvi, A.; Aghdasi, M. PROMETHEE: A comprehensive literature review on methodologies and applications. *Eur. J. Oper. Res.* **2010**, *200*, 198–215. [CrossRef]
29. Opricovic, S.; Tzeng, G.H. Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS. *Eur. J. Oper. Res.* **2004**, *156*, 445–455. [CrossRef]
30. Ruiz-Vélez, A.; García, J.; Alcalá, J.; Yepes, V. Sustainable Road Infrastructure Decision-Making: Custom NSGA-II with Repair Operators for Multi-Objective Optimization. *Mathematics* **2024**, *12*, 30. [CrossRef]
31. Ruiz-Vélez, A.; García, J.; Alcalá, J.; Yepes, V. Enhancing Robustness in Precast Modular Frame Optimization: Integrating NSGA-II, NSGA-III, and RVEA for Sustainable Infrastructure. *Mathematics* **2024**, *12*, 1478. [CrossRef]
32. Sánchez-Garrido, A.J.; Navarro, I.J.; Yepes, V. Multi-criteria decision-making applied to the sustainability of building structures based on Modern Methods of Construction. *J. Clean. Prod.* **2022**, *330*, 129724. [CrossRef]
33. Wang, Y.; Liu, Y.; Ding, K.; Wei, S.; Zhang, X.; Zhao, Y. Dynamic Optimization Method of Knowledge Graph Entity Relations for Smart Maintenance of Cantilever Roadheaders. *Mathematics* **2023**, *11*, 4833. [CrossRef]
34. Mallégol, A.; Khannoussi, A.; Mohammadi, M.; Lacarrière, B.; Meyer, P. Handling Non-Linearities in Modelling the Optimal Design and Operation of a Multi-Energy System. *Mathematics* **2023**, *11*, 4855. [CrossRef]
35. Pareto, V. *Manuale di Economia Politica*; Societa Editrice Libraria: Milano, Italy, 1906.
36. EASC. *GOST 8509–93*; Equal-Leg Angles: Specifications. Federal Agency on Technical Regulating and Metrology: Moscow, Russia, 1993.
37. EASC. *GOST 3068-66*; Steel Wire Ropes: Specifications. Federal Agency on Technical Regulating and Metrology: Moscow, Russia, 1966.
38. Van Rossum, G.; Drake, F.L. *Python 3 Reference Manual*; CreateSpace: Scotts Valley, CA, USA, 2009.