Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A domain-to-structure computational framework for automated truss topology generation and lightweight shape–size optimization

Shuai Yang
Anyang Normal University
Yu-Xin Lin
Northeast Forestry University
Yi-Bo Song
Anyang Normal University
Shi-Ke Zhang
Anyang Normal University
Meng-Meng Lian
Anyang Normal University
Xiao-Liang Zhou
zhouxiiiaoliang@163.com

Zhejiang A&F University

Research Article

Keywords: truss lightweight design, shape and size optimization, Delaunay triangulation, virtual-force relaxation, parametric finite element modeling

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11737255/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A domain-to-structure computational framework for automated truss topology generation and lightweight shape–size optimization

Shuai Yang[a,b], Yu-Xin Lin[c,a], Yi-Bo Song[a,b], Shi-Ke Zhang[a,b], Meng-Meng Lian[a,b], Xiao-Liang Zhou[d,*]

[a] *School of Smart City Engineering, Anyang Normal University, Anyang 455000, Henan, China*
[b] *Henan Province Engineering Technology Research Center of Digital-Intelligent Building and Low-Carbon Building Materials, Anyang Normal University, Anyang 455000, Henan, China*
[c] *School of Civil Engineering and Transportation, Northeast Forestry University, Harbin, 150040,China*
[d] *College of Landscape Architecture, Zhejiang A&F University, Zhejiang, 311300, China*

Abstract: The performance of truss lightweight design is fundamentally constrained by the initial node-member topology prescribed by the designer. Existing ground-structure methods require a complete enumeration of candidate nodes and connections a priori, while free-form generation and data-driven approaches depend on particular search strategies or training datasets that do not generalize reliably across arbitrary design domains. We present a domain-driven method that generates an initial truss graph directly from a prescribed geometric domain—without requiring a predefined topology. The method takes a signed distance function, a set of prescribed mechanical nodes, and a target member-length field as inputs, and constructs candidate truss configurations through one-sided virtual-force relaxation, boundary projection, and dynamic Delaunay-based nodal connectivity reconstruction. The resulting graph is automatically mapped onto a parametric finite-element model in COMSOL Multiphysics, where particle swarm optimization simultaneously optimizes the free-node coordinates and member cross-sectional areas while preserving the generated connectivity. Numerical examples across two- and three-dimensional design domains demonstrate that the procedure produces graphs that satisfy prescribed geometric and connectivity requirements and can be directly converted into solvable finite-element models. Over 20 independent optimization runs, the minimum feasible mass of the two-dimensional structure was reduced from 3276.8 kg to 1225.2 kg (-62.6%), and that of the three-dimensional structure from 11.2135 kg to 2.5995 kg (-76.8%), with all prescribed stress and displacement constraints satisfied. These results establish a computational pathway from an arbitrary design domain to a lightweight truss configuration-bridging the gap between geometric domain representation and automated structural optimization.

Keywords: truss lightweight design; shape and size optimization; Delaunay triangulation; virtual-force relaxation; parametric finite element modeling

[*] Corresponding author.
E-mail address: zhouxiiiaoliang@163.com(Xiao-Liang Zhou).

1 Introduction

Truss structures transmit loads primarily through the axial tension and compression of their constituent members. Owing to their high stiffness-to-mass ratios and well-defined load-transfer paths, they have been extensively employed in bridges, long-span roof systems, towers, spatial structures, and lightweight load-bearing systems. The objective of truss optimization is not merely to minimize material consumption, but rather to achieve a coordinated optimization of structural topology, nodal positions, and member dimensions while satisfying constraints on stress, displacement, stability, and geometric boundaries. Michell's theory [1] established the fundamental characteristics of minimum-material truss systems under specific conditions, whereas the ground-structure approach [2] formulated truss layout design as a tractable mathematical programming problem. Building upon these foundations, homogenization-based, material-distribution, and continuum topology optimization methods have progressively established systematic theoretical and numerical frameworks for structural optimization [3–5]. Existing reviews indicate that structural optimization has evolved from isolated sizing or topology problems towards multivariable, multi-constraint, multiscale, and data-driven design paradigms. Nevertheless, achieving an appropriate balance among design-space representation, engineering constraints, and computational cost remains a persistent challenge [6–8].

Continuum topology optimization enables the redistribution of material within a prescribed design domain and is therefore well suited to identifying dominant load-carrying paths that are not known a priori. Methods such as evolutionary structural optimization (ESO), bidirectional evolutionary structural optimization (BESO), the level-set method, the solid isotropic material with penalization (SIMP) approach, and mathematical programming have established relatively mature frameworks for topology evolution, boundary representation, and constraint treatment [9–17]. Among these, evolutionary approaches achieve structural transformation through the iterative removal and addition of finite elements; level-set methods provide an implicit boundary description that facilitates the representation of well-defined structural interfaces; and SIMP-based formulations, together with optimization algorithms such as the method of moving asymptotes (MMA), offer robust solution strategies for large-scale problems involving continuous design variables. Recent trust-region adaptations of MMA have further improved robustness for topology optimization problems with high nonlinearity and large-scale constraints [18]. Nevertheless, the outcomes of continuum topology optimization are typically represented as density fields or solid boundaries. When the target structure is a truss composed of ideally pin-jointed members, additional procedures are required, including skeleton extraction, node identification, member connectivity construction, intersection treatment, and cross-sectional assignment. These conversion procedures may alter the principal material paths identified in the continuum solution and introduce additional design discrepancies in terms of nodal positions, member count, and local stiffness. Consequently, although continuum topology optimization reduces the dependence on a predefined truss layout, it does not directly provide a node-member graph suitable for truss finite-element analysis.

In contrast to continuum-based approaches, the discrete ground-structure method directly treats the cross-sectional areas or existence states of candidate members as design variables, thereby yielding an explicit node-member topology. Computational frameworks such as GRAND and GRAND3 have extended the ground-structure approach to arbitrary

two- and three-dimensional design domains, while advances in large-scale linear programming, interior-point algorithms, and post-processing techniques have substantially enhanced the tractability of truss layout optimization [19–23]. Concurrently, engineering considerations including local buckling, kinematic stability, member intersection and overlap, failure probability, and material allocation have increasingly been incorporated into discrete topology optimization formulations [24–28]. Further studies have extended the coupled optimization of structural layout and geometric design variables, including simultaneous layout and shape optimization of thin-walled beams [29]. Nevertheless, the final solutions obtained using these methods remain constrained by the prescribed nodal set and candidate connectivity pattern. An inappropriate nodal distribution, excessive or insufficient local discretization density, and an overly large number of candidate long members may all affect both the dimensionality of the optimization problem and the attainable solution space. Thus, although the ground-structure method reduces the uncertainty associated with topology exploration, it also shifts part of the design decision-making process to the initial discretization stage. Its outcomes are therefore governed not only by the optimizer, but also by systematic restrictions imposed by the initial representation of nodes and candidate connections.

To alleviate the restrictions imposed by a fixed ground structure on the attainable design space, several studies have introduced nodal relocation, configuration grammars, dynamic form-finding, and free-form layout generation. Explicit geometry-based topology optimization provides a complementary route by directly parameterizing evolving structural components, including curved-component representations and three-dimensional stress-constrained formulations [30, 31]. At the lattice scale, Singh and Sanders developed an automated skeleton-based approach that manipulates nodal positions and strut connectivity to generate smooth and well-connected transitions between truss-based lattice configurations [32]. One study employed grammatical evolution to represent a variable number of continuously positioned nodes and used Delaunay triangulation to establish stable connectivity, demonstrating that enhanced geometric representational freedom can facilitate the discovery of structural configurations that are difficult to capture using a fixed ground structure [33]. A kernel-regression-based reinforcement learning approach formulated node addition, member connection, and cross-sectional selection as sequential decision-making processes, thereby mitigating the increase in search dimensionality caused by discretizing continuous action spaces [34]. AutoTruss first employed Monte Carlo tree search to identify valid configurations and subsequently applied reinforcement learning to refine the resulting layouts, alleviating the sparse-reward problem arising from the extremely low proportion of feasible structures within the complete search space [35]. In parallel, data-driven methods have been developed to exploit the inherent graph representation of truss structures and to accelerate the search for optimal configurations. Graph convolutional networks, deep neural networks, and multi-agent reinforcement learning have further exploited the inherent graph representation of truss structures to predict structural topologies, member sizes, or sets of non-dominated solutions [36–42]. Beyond graph-based methods, Mai et al. developed a deep unsupervised learning framework for constrained truss sizing optimization by parameterizing member cross-sectional areas using spatial coordinates [43]. At the multiscale level, Chen et al. employed coordinate-based neural representations to optimize structural topology while promoting connectivity be-

tween neighboring microstructures [44]. These studies demonstrate the potential of automated configuration generation and surrogate-based solution strategies. Their performance, however, generally depends on specific encoding schemes, search strategies, or training datasets. Related data-driven mechanics studies have highlighted both the cost of generating high-fidelity training data [45] and the importance of geometric diversity for irregular-geometry prediction [46]. Moreover, geometries, loading conditions, and nodal scales outside the training distribution still require independent verification of physical feasibility. Consequently, although free-form generation and data-driven methods broaden the available representations of structural configurations, they do not eliminate the need for explicit checks of geometric boundary conformity, nodal connectivity, and structural response.

In addition to topological representation, the coupling between nodal coordinates and member cross-sectional areas further governs the lightweighting potential of a truss structure. Nodal relocation simultaneously alters member lengths, orientations, internal-force distributions, and global stiffness, whereas variations in cross-sectional area directly affect structural mass, axial stiffness, and stress levels. Studies employing gradient-based, approximate-gradient, and hybrid evolutionary strategies have demonstrated that the simultaneous optimization of nodal positions and member cross-sections can substantially enlarge the design space compared with sizing optimization alone [47–52]. Related coupled-design strategies have also been reported in lightweight structural systems, involving joint optimization of topology, configuration, stiffener layout, and structural geometry [53, 54]. When structural responses are evaluated using an external finite-element model and reliable analytical sensitivities are difficult to obtain, gradient-free algorithms such as particle swarm optimization (PSO) have been widely adopted because of their straightforward encoding and adaptability to heterogeneous computational interfaces [55–60]. More recent variants have further addressed issues associated with premature convergence to local optima, discrete cross-sectional variables, and redundant finite-element evaluations [61–63]. Nevertheless, most existing shape-size optimization studies still begin with manually prescribed topologies or configurations adopted from the literature. They therefore focus primarily on the adjustment of continuous design parameters under a fixed connectivity pattern, while giving comparatively limited attention to how such connectivity can be generated automatically from a prescribed geometric design domain.

As truss optimization has become increasingly integrated with parametric finite-element analysis and automated model updating, structural geometry, cross-sectional parameters, and boundary conditions can be iteratively exchanged between external optimization routines and finite-element models [64–66]. Such frameworks provide a practical basis for evaluating structural responses in shape-size optimization. However, when nodes or member connections are added or removed during the optimization process, it may become difficult to maintain consistent element numbering, parameter indexing, and boundary-condition assignments within commercial finite-element software. Consequently, a trade-off remains between configurational freedom and the robustness of the finite-element interface. A fully fixed and regular ground structure facilitates parameter management, but may contain candidate members whose characteristic lengths are incompatible with the scale of the design domain. Conversely, continuously updating the topology enlarges the accessible design space, but increases the complexity of model reconstruction and data mapping.

Taken together, the preceding studies indicate that the central question addressed in this work is how to construct, from a prescribed geometric design domain and mechanically

constrained boundary nodes, an initial truss graph suitable for subsequent lightweight design without specifying a complete node-member topology a priori. The key idea proposed herein is to treat candidate nodes as particles subject to one-sided repulsive virtual forces, which naturally distribute them within the design domain and suppress locally clustered configurations, analogous to the equilibrium spacing achieved in physical particle systems. Conventional ground-structure methods require the prior enumeration of nodes and candidate members, whereas free-form generation and learning-based approaches generally depend on particular evolutionary encodings, tree-search strategies, training datasets, or sequential decision-making procedures. A further challenge lies in how the generated graph can be robustly mapped to a parametric finite-element model and subsequently incorporated into structural optimization. If members are continuously added or removed during the optimization process, changes in connectivity may substantially increase the effort required to maintain consistent element numbering, parameter indexing, and boundary-condition mapping within commercial finite-element software. It is therefore necessary to strike a balance between the representational freedom of the initial configuration and the maintainability of the finite-element model, and to establish a staged computational pathway linking initial truss-graph generation with subsequent lightweight optimization.

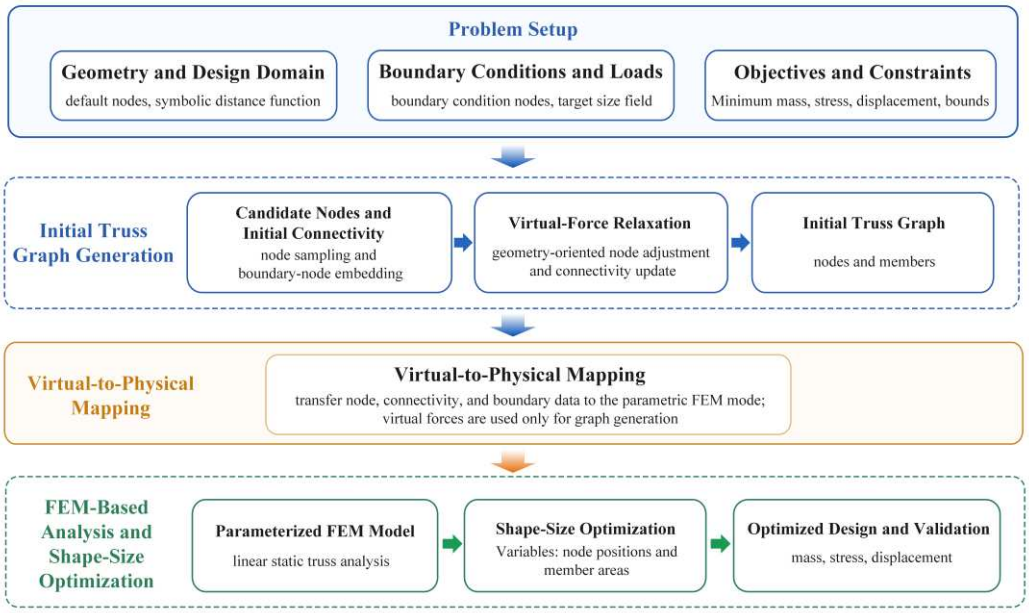


Figure 1: Overall computational workflow of the proposed staged framework for lightweight truss design.

To this end, this study proposes an initial truss-graph generation method based on virtual-force relaxation and, on this basis, develops a staged computational framework for the lightweight design of two- and three-dimensional truss structures. First, a candidate truss graph is generated from a prescribed design domain, predefined mechanical nodes, and a target member-length field through one-sided virtual repulsive forces, boundary projection, and dynamic Delaunay adjacency reconstruction. The generated graph is subsequently mapped to a parametric finite-element model. As illustrated in Fig. 1, the proposed framework proceeds through three sequential stages: initial truss-graph generation, parametric finite-element modelling, and PSO-based simultaneous shape-size optimization. The main contributions of this study are threefold. First, a virtual-force relaxation method

is proposed to generate candidate truss graphs directly from a prescribed design domain without requiring a predefined node-member topology. Second, a staged computational framework is established that decouples graph generation from optimization, thereby preserving finite-element model consistency throughout the optimization procedure. Third, the framework is validated through two- and three-dimensional numerical examples, demonstrating mass reductions of 62.6% and 76.8%, respectively, while satisfying all prescribed constraints. The remainder of this paper is organized as follows. Section 2 presents the problem definition and mathematical formulation. Section 3 details the initial truss-graph generation algorithm. Section 4 describes the parametric finite-element implementation and the PSO-based optimization procedure. Section 5 presents numerical examples and results. Section 6 concludes the paper.

2 Problem definition and mathematical formulation

This section formulates the initial truss-graph generation problem and the subsequent lightweight optimization problem within a common framework, marking the boundary between the virtual graph-generation stage and the physical finite-element analysis stage. The formulation provides a mathematical basis for the generation algorithm, the parametric finite-element implementation, and the numerical validation that follow.

2.1 Design domain and truss-graph representation

Let the design domain Ω be embedded in d-dimensional Euclidean space, with $d = 2$ for planar trusses and $d = 3$ for spatial trusses. Accordingly,

$$\Omega \subset \mathbb{R}^d \quad d = 2,3 \tag{1}$$

The domain boundary is represented by the signed distance function $\varphi(\boldsymbol{\xi})$:

$$\partial\Omega = \{\boldsymbol{\xi} \mid \varphi(\boldsymbol{\xi}) = 0\} \tag{2}$$

During graph generation, the supports, load points, and any geometric anchors are termed prescribed nodes and are denoted P_s, P_l, and P_a, respectively. The prescribed-node set is

$$P_{pre} = P_s \cup P_l \cup P_a \tag{3}$$

Prescribed-node coordinates remain fixed throughout virtual-force relaxation. All other candidate nodes are movable and their positions are updated by the virtual forces.

After nodal relaxation, boundary projection, and adjacency reconstruction, the candidate configuration is an undirected graph:

$$G = (P, B) \tag{4}$$

where P is the node set and B is the member-connectivity set. For n nodes and n_b members,

$$P = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\} \tag{5}$$

$$B = \{e_k = (i, j) \mid i, j \in \{1, 2, ..., n\}, i < j, k = 1, 2, .., n_b\} \tag{6}$$

2.2 Virtual graph generation and physical finite-element analysis

In the generation stage, movable node coordinates are iterated under virtual forces.

After each coordinate update, the Delaunay tessellation is recomputed and candidate connections are screened against the design domain. Node positions and edge set thus evolve jointly. The procedure can be abstracted as

$$\mathbf{F}_i^v(\mathbf{p}) \to 0 \tag{7}$$

where $\mathbf{F}_i^v$ is the resultant virtual force on node i and $\mathbf{p}$ is the vector of global nodal coordinates. This force is a purely artificial relaxation mechanism: it regulates node spacing and prevents local clustering of short members, and it does not correspond to any physical load or member axial force.

Once the candidate graph passes the prescribed checks, its node coordinates and connectivity are used to build a parametric finite-element model. The supports and load points are transferred to the physics interface, where the corresponding constraints and external loads are applied. Given the material properties and the cross-sectional area vector $\mathbf{A}$, the linear-elastic static response satisfies

$$\mathbf{K}(\mathbf{p},\mathbf{A})\mathbf{u} = \mathbf{f}^{ext} \tag{8}$$

where $\mathbf{K}$ is the global stiffness matrix, $\mathbf{u}$ is the nodal displacement vector, and $\mathbf{f}^{ext}$ is the external load vector.

2.3 Coupled shape–size optimization problem

For an initial truss graph $G = (P, B)$ that passes the finite-element solvability check, the optimization stage fixes B and treats both free-node coordinates and member cross-sectional areas as design variables. The initial configuration is then refined under the prescribed physical loading and constraints.

To separate the fixed prescribed nodes used during generation from the boundary conditions applied in finite-element analysis, the boundary-condition node set is defined as

$$P_{bc} = P_s \cup P_l \tag{9}$$

Boundary-condition node coordinates remain fixed. The remaining nodes, whose coordinates are shape design variables, constitute the free-node set

$$P_f = P \setminus P_{bc} \tag{10}$$

For n_f free nodes and n_b members, the complete design-variable vector is

$$\mathbf{x} = [\mathbf{q}, \mathbf{A}] \tag{11}$$

where $\mathbf{q}$ is the free-node coordinate vector and $\mathbf{A}$ is the member cross-sectional area vector. The global nodal coordinate vector $\mathbf{p}$ is assembled from $\mathbf{q}$ and the fixed boundary-condition node coordinates. For two-dimensional trusses,

$$\mathbf{q} = [x_1, y_1, x_2, y_2, ..., x_{nf}, y_{nf}] \tag{12}$$

The optimization objective is to minimize the total structural mass subject to the prescribed constraints. This optimization formulation incorporates constraints on member axial stress and nodal displacement, together with lower and upper bounds on the free-node coordinates and member cross-sectional area.

$$\begin{cases} \min \quad M(\mathbf{p}(\mathbf{q}),\mathbf{A}) = \sum_{k=1}^{n_b} \rho A_k L_k(\mathbf{p}(\mathbf{q})) \\ s.t. \quad |\sigma_k| \le \sigma_{\text{allow}} \quad k = 1,2,...,n_b \\ \qquad u_{\max} \le u_{\text{allow}} \\ \qquad \mathbf{q}_i^L \le \mathbf{q}_i \le \mathbf{q}_i^U \quad i = 1,2,...,n_f \\ \qquad A^L \le A_k \le A^U \quad k = 1,2,...,n_b \end{cases} \tag{13}$$

Here ρ is the material density, A_k is the cross-sectional area of member k, $L_k(\mathbf{p}(\mathbf{q}))$ is its length at the current nodal positions, σ_k is the axial stress in member k from the finite-element analysis, and $u_{\max}$ is the maximum nodal displacement magnitude. The quantities σ_{allow} and u_{allow} are the allowable stress and displacement, respectively; $\mathbf{q}_i^L$, $\mathbf{q}_i^U$, $\mathbf{A}^L$, and $\mathbf{A}^U$ are the corresponding lower and upper bounds.

3 Initial truss-graph generation via virtual-force relaxation

The quality of the initial truss graph determines the optimization search space, the efficiency of the finite-element model, and the ultimate mass of the lightweight design. Existing approaches either require a complete node-member topology a priori or rely on data-driven methods that generalize poorly to untrained geometries. We address this gap with a virtual-force relaxation procedure that takes as input a signed distance function defining the design domain, a set of prescribed mechanical nodes, and a target member-length field, and directly yields a valid truss graph without requiring prior enumeration of a ground structure (Fig. 2).

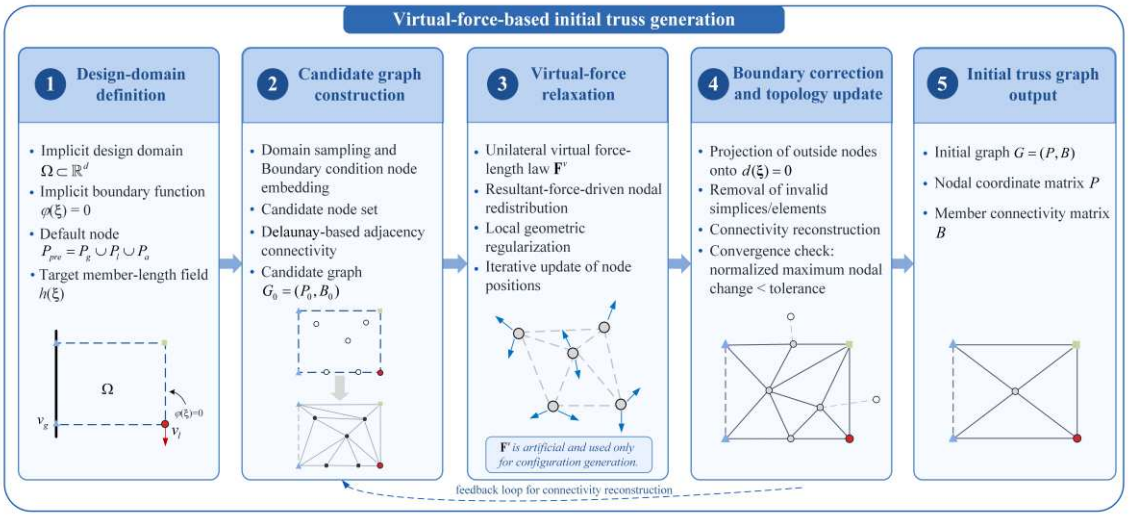


Figure 2: Workflow of initial truss-graph generation via virtual-force relaxation.

3.1 Implicit design domain, target member-length field, and candidate-node construction

We represent the design domain Ω by a signed distance function $\varphi(\xi)$, defined by the designer according to the prescribed geometric requirements:

$$\varphi(\xi)=\begin{cases}-d(\xi,\partial\Omega), & \xi\in\Omega\\ 0, & \xi\in\partial\Omega\\ d(\xi,\partial\Omega), & \xi\notin\Omega\end{cases} \tag{14}$$

where ξ is the spatial coordinate, $\partial\Omega$ is the domain boundary, and $d(\xi, \partial\Omega)$ is the minimum Euclidean distance to that boundary. Under Eq. (14), $\varphi(\xi) < 0$, $\varphi(\xi) = 0$, and $\varphi(\xi) > 0$ indicate interior, boundary, and exterior points, respectively. The magnitude $|\varphi(\xi)|$ gives the distance to the nearest boundary (Fig. 3(a)). We use this field to screen candidate nodes and to correct nodes that move outside the admissible domain during relaxation.

The designer prescribes a target member-length field $h(\xi)$ that controls the local discretization density — the spacing of candidate nodes and the characteristic lengths of candidate members. For uniform discretization, a constant target member length is adopted:

$$h(\xi)=h_0 \tag{15}$$

When higher resolution is required near load application points, support regions, or local geometric features, a spatially varying field is defined instead:

$$h(\xi)=h_0\psi(\xi),\quad \psi(\xi)>0 \tag{16}$$

where h_0 is the reference target length and $\psi(\xi)$ is a spatial scaling function. Smaller values of $h(\xi)$ produce denser nodal distributions and finer member scales (Fig. 3(b)).

We distribute movable candidate nodes within Ω at a spacing controlled by $h(\xi)$. The prescribed nodes — supports, loads, and anchor points — are then added to form the initial node set (Fig. 3(c)).

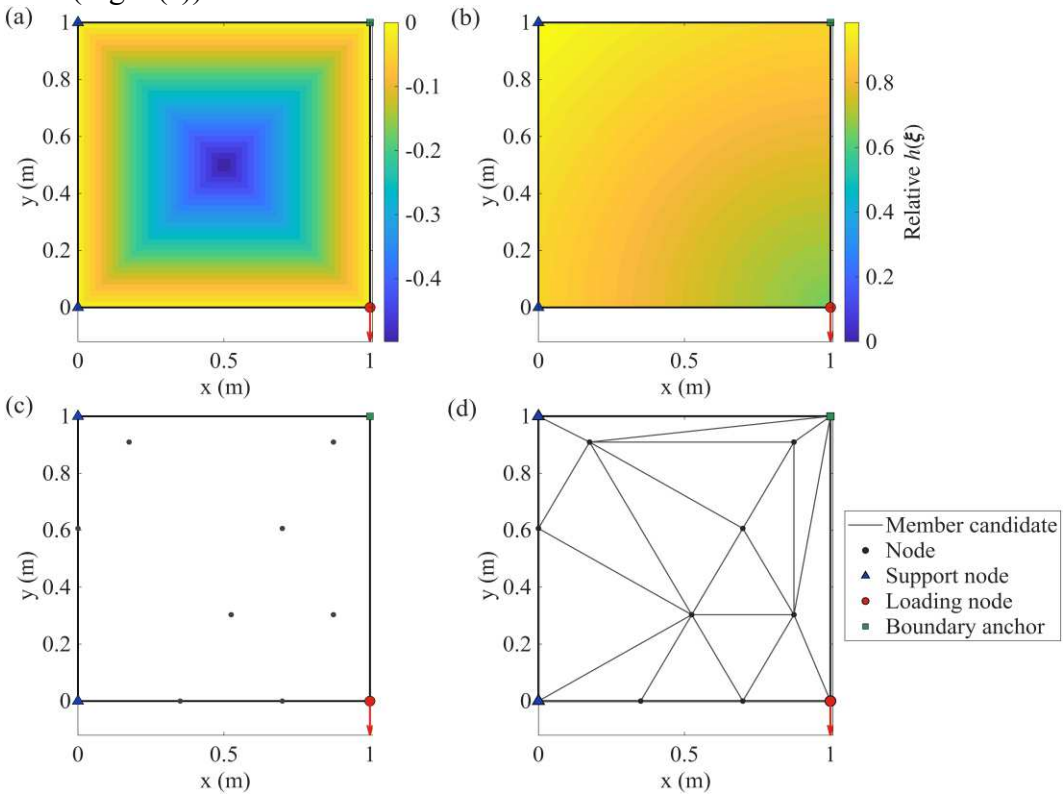


Figure 3: Construction of the initial candidate graph: (a) implicit design domain; (b) target member-length field $h(\xi)$; (c) candidate and prescribed nodes; and (d) initial candidate graph $G_0=(P_0, B_0)$.

Candidate adjacency relationships are constructed from the current nodal positions. At iteration k, the node set $P^{(k)}$ is triangulated by Delaunay tessellation, and the edges of the resulting simplices become candidate members:

$$B^{(k)} = T(\mathrm{Del}(P^{(k)})) \tag{17}$$

where $\mathrm{Del}(P^{(k)})$ denotes the Delaunay tessellation and $T(\cdot)$ is the edge-extraction operator. For two-dimensional problems, members are the edges of Delaunay triangles; for three-dimensional problems, they are the edges of Delaunay tetrahedra. The graph at iteration k is $G^{(k)} = (P^{(k)}, B^{(k)})$, with its initial form G_0 shown in Fig. 3(d).

3.2 One-sided virtual forces and local-scale adjustment

For each candidate member $e_{ij} = (i, j)$, let $\mathbf{p}_i$ and $\mathbf{p}_j$ denote the position vectors of its end nodes. The relative endpoint vector, current member length, and midpoint coordinate are:

$$\mathbf{r}_{ij} = \mathbf{p}_i - \mathbf{p}_j \tag{18}$$

$$L_{ij} = \left\| \mathbf{r}_{ij} \right\| \tag{19}$$

$$x_{ij} = \frac{\mathbf{p}_i + \mathbf{p}_j}{2} \tag{20}$$

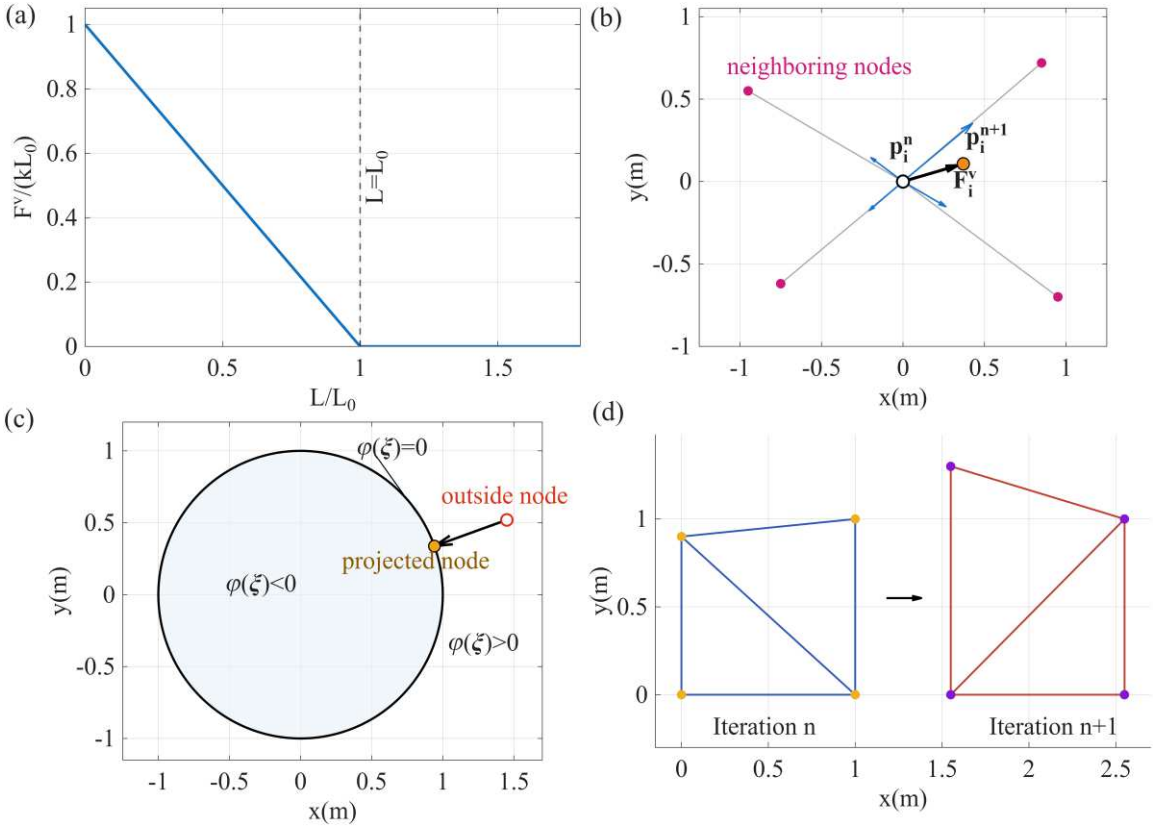


Figure 4: Local mechanism and boundary treatment in virtual-force relaxation: (a) one-sided virtual force-length relationship; (b) nodal update driven by the resultant virtual force; (c) node projection based on the implicit boundary; and (d) Delaunay adjacency reconstruction.

To reconcile the prescribed target member-length field with the overall length scale

of the graph, we introduce a globally scaled target member length:

$$L_{ij}^{0} = \alpha h(\boldsymbol{\xi}_{ij})\left(\frac{\sum_{e\in B^{k}} L_{e}^{d}}{\sum_{e\in B^{k}} h_{e}^{d}}\right)^{\frac{1}{d}} \tag{21}$$

where

$$\alpha = 1 + \frac{0.4}{2^{d-1}} \tag{22}$$

$$h_{e} = h(\boldsymbol{\xi}_{ij}) \tag{23}$$

Here, h_e is the local target length evaluated at the member midpoint, and α is a scale-correction coefficient that depends on the spatial dimension d.

Initial node sampling and the embedding of prescribed nodes can produce local clustering and excessively short candidate edges. We suppress these with a one-sided repulsive interaction: a virtual force acts only when $L_{ij} < L_{ij}^{0}$; when $L_{ij} \geq L_{ij}^{0}$, the force is zero (Fig. 4(a)). The force magnitude is:

$$f_{ij}^{v} = k_{v}\max(L_{ij}^{0} - L_{ij}, 0) \tag{24}$$

The corresponding force vector on node i is:

$$\mathbf{f}_{ij}^{v} = f_{ij}^{v}\frac{\mathbf{r}_{ij}}{L_{ij}} \tag{25}$$

where k_v is the virtual stiffness coefficient. The force $\mathbf{f}_{ij}^{v}$ is directed from node j toward node i, pushing adjacent nodes apart when the member falls below its scaled target length. This one-sided mechanism eliminates nodal clustering and suppresses extremely short members while exerting no active pull on longer ones.

3.3 Nodal updating, boundary projection, and dynamic adjacency reconstruction

At iteration k, the resultant virtual force on node i superposes contributions from all incident candidate members:

$$\mathbf{F}_{i}^{v,k} = \sum_{\substack{j \\ (i,j)\in B^{(k)}}} \boldsymbol{f}_{ij}^{v,k} \tag{26}$$

We introduce an artificial time variable t and formulate the relaxation as a first-order pseudo-time evolution:

$$\frac{d\mathbf{p}_{i}}{dt} = \mathbf{F}_{i}^{v}(\mathbf{p}) \tag{27}$$

This equation drives each movable node along its resultant force direction. As short members are progressively suppressed, the nodal increments diminish.

Discretizing by forward Euler gives the trial position:

$$\tilde{\mathbf{p}}_{i}^{k+1} = \mathbf{p}_{i}^{k} + \Delta t\mathbf{F}_{i}^{v,k} \tag{28}$$

where Δt is the artificial time-step controlling the per-iteration displacement. Prescribed nodes — supports, loads, and anchor points — remain fixed throughout relaxation:

$$\mathbf{p}_{i}^{(k+1)} = \mathbf{p}_{i}^{(k)} \tag{29}$$

The trial update is illustrated in Fig. 4(b).

If a trial-updated node lies outside the design domain, we project it onto the boundary along the negative gradient of the signed distance function (Fig. 4(c)):

$$\mathbf{p}_i^{(k+1)} = \tilde{\mathbf{p}}_i^{(k+1)} - \varphi(\tilde{\mathbf{p}}_i^{(k+1)}) \frac{\nabla \varphi(\tilde{\mathbf{p}}_i^{(k+1)})}{\left\| \nabla \varphi(\tilde{\mathbf{p}}_i^{(k+1)}) \right\|^2} \tag{30}$$

If the trial-updated node remains within the domain, no correction is required:

$$\mathbf{p}_i^{(k+1)} = \tilde{\mathbf{p}}_i^{(k+1)} \tag{31}$$

Eq. (30) performs a normal correction using the zero level set $\varphi(\boldsymbol{\xi}) = 0$. When an analytical gradient $\nabla\varphi(\boldsymbol{\xi})$ is unavailable, we approximate it by finite differences.

After the update and boundary correction, we recompute the Delaunay tessellation from $P^{(k+1)}$ and extract $B^{(k+1)}$. Candidate edges that violate the domain geometry are removed (Fig. 4(d)).

3.4 Stopping criterion and initial truss-graph output

We quantify the geometric change between successive iterations by the normalized maximum nodal displacement:

$$\delta_k = \max \frac{\left\| \mathbf{p}_i^{(k+1)} - \mathbf{p}_i^{(k)} \right\|}{h_0} \tag{32}$$

When $\delta_k < \varepsilon_{\text{tol}}$, the nodal positions satisfy the stopping criterion. The computation also terminates at a prescribed maximum iteration count.

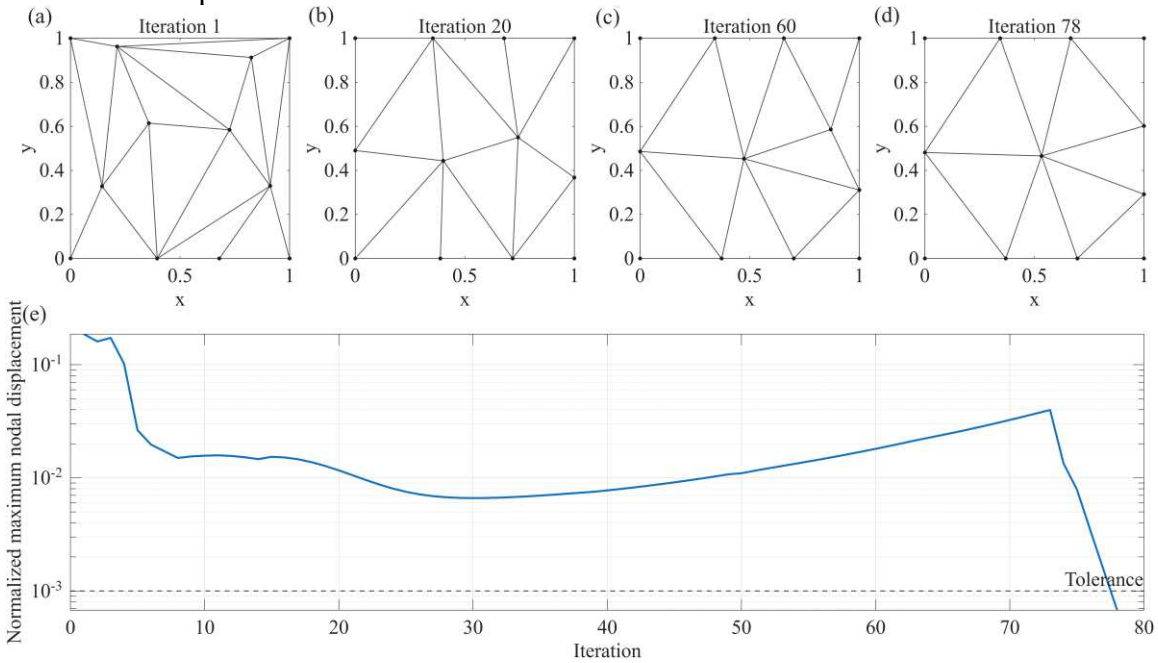


Figure 5: (a-d) Node-member graphs at selected iterations; (e) history of the normalized maximum nodal displacement.

Figure 5 shows the evolution of the node-member graph and the convergence history for a square test domain. During the initial iterations, both node positions and adjacency adjust substantially, yielding a rapid decrease in the convergence measure.

At termination, we perform geometric and graph-validity checks: all nodes must lie

within or on the domain boundary; all prescribed nodes must be retained; and a valid connectivity path must exist between supports and loads. Isolated nodes, invalid connected components, duplicate members, and members shorter than $L_{\min}$ are removed. The initial truss graph satisfying all criteria is denoted by

$$G = (P, B) \tag{33}$$

This graph serves as the initial configuration for the parametric finite-element model and the subsequent shape-size optimization under fixed member connectivity.

4 Parametric Finite-Element-Driven Simultaneous Shape and Size Optimization

We map the verified initial truss graph onto a parametric COMSOL model. With the member connectivity held fixed, we then perform simultaneous shape and size optimization using particle swarm optimization (PSO), treating the free-node coordinates and member cross-sectional areas as continuous design variables.

4.1 Characteristics of the Optimization Problem and PSO-Based Solution Strategy

The movement of free nodes alters the member lengths and orientations as well as the global structural stiffness, whereas variations in the cross-sectional areas affect the axial stiffness, stress state, and structural mass. Consequently, the objective function and constraint responses are nonlinearly coupled with respect to the shape and sizing variables. Moreover, all structural responses must be obtained through parametric finite-element analyses, making it difficult to directly derive reliable analytical sensitivities within a unified framework. In view of these problem characteristics, particle swarm optimization (PSO) is adopted to perform the search without requiring explicit gradient information. PSO encodes the shape and sizing variables within a unified position vector and guides the search using the mass, displacement, and stress responses returned by the finite-element model. This choice is motivated by the compatibility of the algorithm with the form of the design variables and the computational interface, rather than by any presumption that PSO offers a universal advantage over other optimization methods.

Let $\mathbf{x}_r^t$ denote the position vector of the r-th particle at iteration t. Its unified encoding is expressed as

$$\mathbf{x}_r^t = \left[\mathbf{q}_r^t, \mathbf{A}_r^t \right] \tag{34}$$

where $\mathbf{q}_r^t$ is the coordinate vector of the free nodes and $\mathbf{A}_r^t$ is the vector of member cross-sectional areas. Nodes subjected to prescribed boundary conditions are excluded from the shape design variables, and their coordinates remain fixed throughout the optimization process. All design variables are bounded by the lower and upper limits specified in Section 2.3, thereby restricting the candidate geometries and cross-sectional dimensions to the prescribed design space.

For each particle, the position vector is first decoded according to a fixed indexing scheme to assemble the global nodal coordinate matrix and the vector of member cross-sectional areas. The parametric finite-element model is then updated and subjected to a static analysis. The resulting nodal displacements, member axial forces, and structural mass

are extracted to evaluate the constraint violations and fitness value. This procedure can be expressed as

$$\mathbf{x}_r^t \rightarrow \{\mathbf{p}_r^t, \mathbf{A}_r^t\} \rightarrow M(\mathbf{p}_r^t, \mathbf{A}_r^t) \rightarrow \{\mathbf{u}_r^t, \mathbf{N}_r^t, m_r^t\} \rightarrow F(\mathbf{x}_r^t) \tag{35}$$

where $\mathbf{p}_r^t$ denotes the global nodal coordinate matrix assembled from the coordinates of the free nodes and those of the boundary-condition nodes; $M(\mathbf{p}_r^t, \mathbf{A}_r^t)$ represents the corresponding parametric finite-element model; and $\mathbf{u}_r^t$, $\mathbf{N}_r^t$, and m_r^t denote the nodal displacements, member axial forces, and structural mass obtained from the finite-element analysis, respectively. The quantity $F(\mathbf{x}_r^t)$ is the fitness value associated with the particle. The PSO generates candidate designs and updates the search state, while the finite-element model returns the corresponding physical responses. The constraint-handling module subsequently feeds the feasibility information back into the search process, thereby establishing the closed computational loop illustrated in Fig. 6.

4.2 Parametric Finite-Element Mapping of the Initial Truss Graph

The initial truss graph G that passes the solvability check is mapped onto a linear-elastic static truss model under the small-deformation assumption. The members are represented by truss elements that sustain axial tension and compression only, while the joints are idealized as perfectly pinned connections. The material properties, support conditions, and external loads are specified separately for each numerical example.

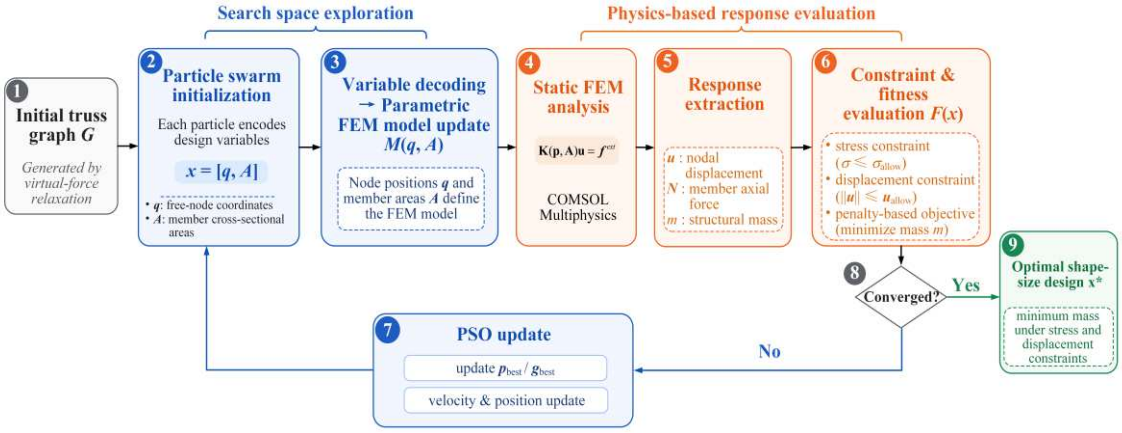


Figure 6: Closed-loop framework for parametric-finite-element-coupled simultaneous shape and size optimization using PSO.

The parametric mapping preserves the node numbering, member-endpoint indices, and boundary-condition selection sets of the initial truss graph. For each candidate design, MATLAB updates the coordinates of the free nodes and the member cross-sectional areas according to the particle position vector and subsequently transfers the corresponding parameters to the parametric truss model established in COMSOL Multiphysics®. COMSOL then updates the member geometry and cross-sectional properties and performs a linear static finite-element analysis under the prescribed material properties, support conditions, and external loads. The resulting structural responses, including the nodal displacements and member axial forces, are returned to MATLAB for constraint evaluation and fitness calculation. Because the member connectivity set B remains fixed throughout the optimization, model updates involve only the geometric and cross-sectional parameters, without

element insertion or removal or any topological reconstruction.

To facilitate the unified management of configuration data, finite-element parameters, and optimization settings, a structured data container is established in MATLAB to store the nodal coordinates, member-endpoint indices, cross-sectional areas, material properties, support and loading definitions, and optimization parameters. During the evaluation of each particle, only the data fields associated with the current shape and size variables are updated, while all other model information remains unchanged. Before optimization is initiated, the initial model must successfully undergo a finite-element analysis. If the stiffness matrix is singular, the numerical solution fails, or any non-finite response is returned, the corresponding initial truss graph is excluded from the optimization stage.

4.3 Design-Variable Mapping and Parametric Model Updating

According to the definition of the design variables in Section 2.3, the particle position vector comprises the shape and size variables arranged in a prescribed order. For the *r*-th particle at iteration *t*, the corresponding variable mapping is expressed as

$$D:\mathbf{x}_r^t \rightarrow (\mathbf{p}_r^t, \mathbf{A}_r^t) \tag{36}$$

where D denotes the variable-mapping operator defined by the prescribed indexing scheme; $\mathbf{p}_r^t$ is the global nodal coordinate matrix assembled from the fixed coordinates of the boundary-condition nodes and the updated coordinates of the free nodes; and $\mathbf{A}_r^t$ is the updated vector of member cross-sectional areas. This mapping preserves the member-endpoint indices, while the boundary-condition nodes remain fixed.

Once the index-based mapping has been completed, the updated global nodal coordinate matrix and cross-sectional-area vector are transferred to the parametric finite-element model:

$$M(\mathbf{p}^{old}, \mathbf{A}^{old}) \rightarrow M(\mathbf{p}^{new}, \mathbf{A}^{new}) \tag{37}$$

4.4 Finite-Element Response Extraction, Constraint Handling, and Fitness Evaluation

A linear static finite-element analysis is performed for each candidate design that successfully passes the parameter-mapping and model-update checks. The resulting structural responses comprise the nodal displacement vector, member axial-force vector, and structural mass, which may be expressed as

$$M(\mathbf{p}, \mathbf{A}) \rightarrow \{\mathbf{u}, \mathbf{N}, m\} \tag{38}$$

where $\mathbf{N}$ denotes the vector of member axial forces and m is the total structural mass.

The axial stress in the *k*-th member is determined from the corresponding finite-element axial force and its current cross-sectional area:

$$\sigma_k = \frac{N_k}{A_k} \quad k = 1, 2, ..., n_b \tag{39}$$

An exterior penalty function is employed to incorporate the displacement and stress constraints into the fitness evaluation. To prevent differences in physical units and numerical scales from exerting disproportionate influences on the penalty terms, the violation of each constraint is normalized with respect to its corresponding allowable value. The violation measure is set to zero when the constraint is satisfied, and only the portion exceeding

the prescribed limit contributes to the penalty.

The normalized violation measure associated with the displacement constraints is defined as

$$G_u(\mathbf{x}) = \max\left(0, \frac{u_{x,\max}}{u_{\text{allow}}} - 1\right)^2 + \max\left(0, \frac{u_{y,\max}}{u_{\text{allow}}} - 1\right)^2 \tag{40}$$

Eq. (40) gives the displacement-constraint violation measure for two-dimensional problems. For three-dimensional problems, the displacement component in the z-direction is incorporated in an analogous manner. The normalized violation measure associated with the stress constraints is defined as

$$G_\sigma(\mathbf{x}) = \sum_{k=1}^{m} \max(0, \frac{|\sigma_k|}{\sigma_{allow}} - 1)^2) \tag{41}$$

Accordingly, the fitness function of a candidate design is formulated as

$$F(\mathbf{x}) = M(\mathbf{x}) + \eta[G_u(\mathbf{x}) + G_\sigma(\mathbf{x})] \tag{42}$$

where η denotes the penalty factor. When a candidate design satisfies all prescribed constraints, its fitness value is equal to the structural mass. Once any constraint is violated, the fitness value increases with the corresponding normalized violation measure, thereby reducing the likelihood that an infeasible candidate will be accepted as either the personal best or the global best.

If the candidate geometry cannot be incorporated into the parametric model, the finite-element solution fails, or an exception occurs during structural-response extraction, the candidate is classified as numerically infeasible and assigned a failure-penalty value:

$$F(\mathbf{x}) = F_{\text{fail}} \tag{43}$$

where F_{fail} is a constant selected to be substantially larger than the characteristic magnitude of the structural mass under normal conditions.

4.5 PSO Search Procedure and Termination Criteria

Within the evaluation loop described above, each particle represents a complete shape-size design with fixed member connectivity. Following initialization, the fitness of every particle is evaluated through the same sequence of design-variable mapping, finite-element analysis, and constraint assessment. The resulting fitness values are subsequently used to update the personal best position of each particle and the global best position of the swarm.

Let $\mathbf{x}_r^t$ and $\mathbf{v}_r^t$ denote, respectively, the position and velocity of the r-th particle at iteration t. Its personal best position and the global best position of the swarm serve as the cognitive and social learning references, respectively. The particle velocity is updated according to

$$\mathbf{v}_r^{t+1} = \omega \mathbf{v}_r^t + c_1 r_1 (\mathbf{pbest}_r^t - \mathbf{x}_r^t) + c_2 r_2 (\mathbf{gbest}^t - \mathbf{x}_r^t) \tag{44}$$

The particle position is advanced using the updated velocity according to

$$\mathbf{x}_r^{t+1} = \mathbf{x}_r^t + \mathbf{v}_r^{t+1} \tag{45}$$

Here, ω denotes the inertia weight; c_1 and c_2 are the cognitive and social learning coefficients, respectively; and r_1 and r_2 are random numbers drawn from the interval [0,1]. If the updated particle position exceeds the prescribed design-variable bounds, it is projected

componentwise onto the admissible range:

$$\mathbf{x}_r^{t+1} = \min[\max(\mathbf{x}_r^{t+1}, \mathbf{x}^L), \mathbf{x}^U] \tag{46}$$

where $\mathbf{x}^L$ and $\mathbf{x}^U$ denote the lower and upper bounds of the design variables, respectively.

The iterative search is terminated when either the prescribed maximum number of iterations is reached or the variation in the best-so-far global fitness remains below a specified tolerance for a predefined number of consecutive iterations. The final reported solution must yield a valid finite-element solution and satisfy all stress, displacement, and design-variable-bound constraints. The overall execution procedure of the PSO algorithm is illustrated in Fig. 7.

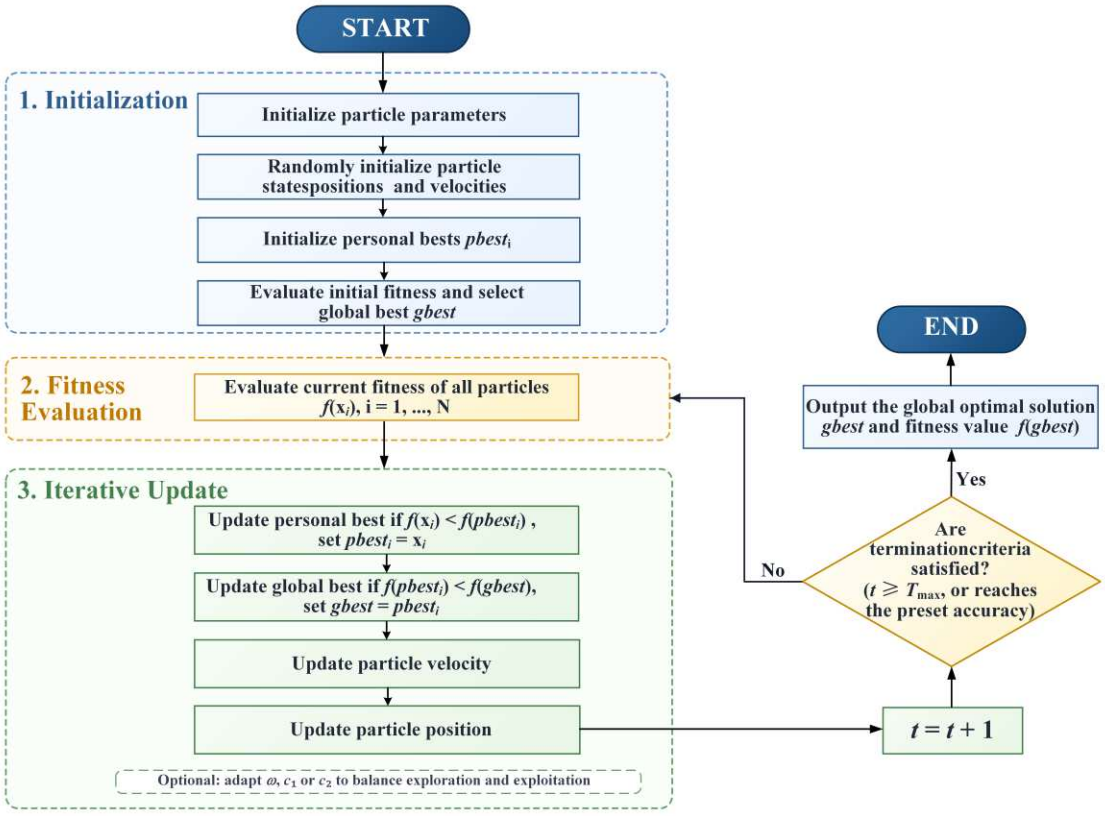


Figure 7: Flowchart of the particle swarm optimization procedure.

5 Numerical Investigation and Validation

We validate the proposed framework at three levels: initial truss-graph generation, the solvability of the resulting parametric finite-element models, and simultaneous shape and size optimization under fixed member connectivity.

5.1 Validation of Initial Truss-Graph Generation

We evaluate the responsiveness of the initial truss-graph generation procedure across four representative configurations: a rectangular domain with uniform and refined target member-length fields, a non-uniform target member-length field, and a design domain containing an internal hole (Fig. 8). Each configuration includes prescribed and movable nodes,

and candidate connections are established through virtual-force relaxation, boundary projection, and dynamic Delaunay adjacency reconstruction, without requiring a pre-enumerated ground structure.

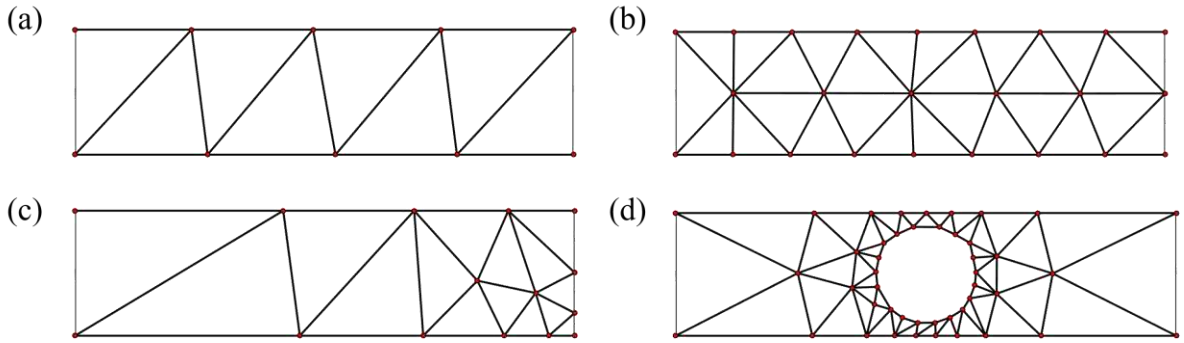


Figure 8: Initial truss graphs generated for different design domains and target member-length fields: (a) coarse uniform discretization; (b) refined uniform discretization; (c) non-uniform target member-length field; and (d) design domain containing an internal hole.

Within the same rectangular design domain, the target member length controls the discretization density: larger values produce sparser candidate networks, smaller values yield denser triangulated patterns (Fig. 8(a)–(b)). A non-uniform field enables locally refined discretization (Fig. 8(c)). For the domain with an internal hole, geometric filtering removes connections that intersect the hole while preserving connectivity across the domain (Fig. 8(d)). These results demonstrate that the target member-length field and the implicit boundary representation jointly regulate the local resolution and spatial distribution of the candidate graph.

All four candidate graphs retain the complete set of prescribed nodes and pass the checks for isolated nodes, invalid subgraphs, duplicate members, and excessively short members. Their structural solvability must still be verified through finite-element analysis under the specified support and loading conditions.

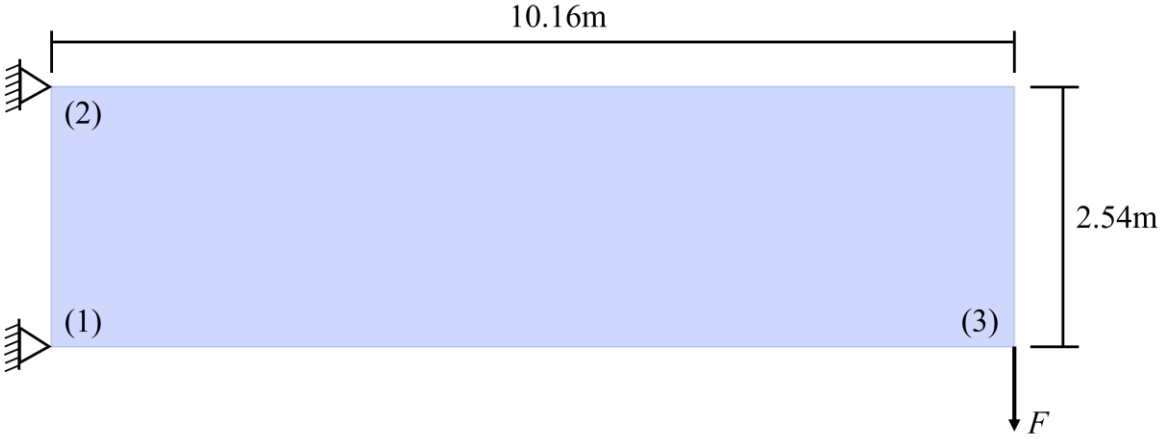


Figure 9: Design domain and boundary conditions of the two-dimensional planar cantilever truss.

Table 1: Coordinates of the boundary-condition nodes for the two-dimensional planar cantilever truss

Node type	**Node coordinates (m)**
Fixed-support nodes	(0.0, 0.0), (0.0, 2.54)
Concentrated-load application point	(10.16, 0.0)

Table 2: Material properties of the two-dimensional planar cantilever truss

Material property	Value
Density	7418.21 kg/m^3
Young's modulus	206.85 GPa
Poisson's ratio	0.3

Table 3: Constraints and design-variable bounds for the two-dimensional planar cantilever truss

Parameter	Value
Allowable axial stress	334.6 MPa
Nodal displacement limit	50.8 mm
Member cross-sectional-area range	0.6452–200 cm^2

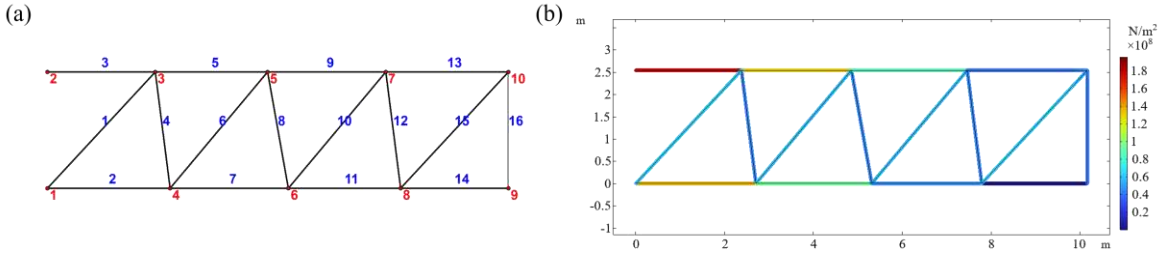


Figure 10: Initial truss graph and finite-element response for the two-dimensional example: (a) initial truss graph; and (b) distribution of member axial stresses in the initial finite-element model.

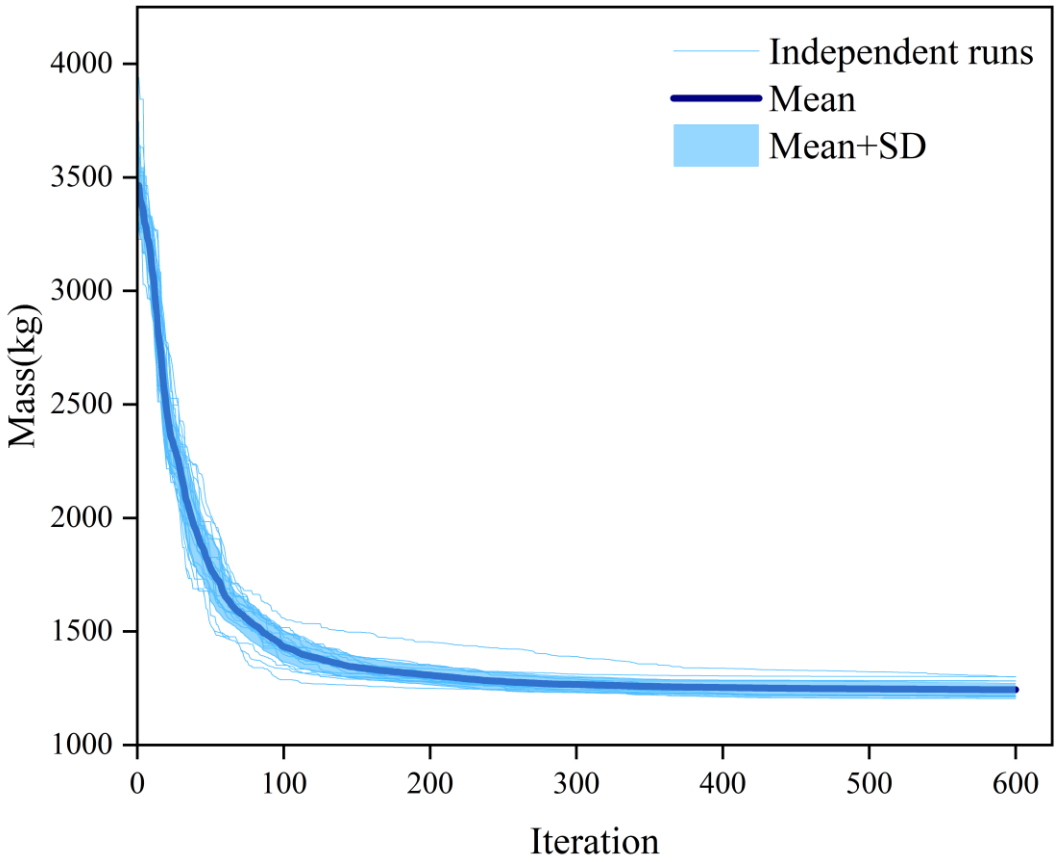


Figure 11: Convergence histories of structural mass over 20 independent optimization runs for the two-dimensional example. The thin light-blue lines represent individual optimization runs, the dark-blue line denotes the mean mass, and the shaded region indicates the mean ± one sample standard deviation.

5.2 Two-Dimensional Planar Cantilever Truss Example

We consider the benchmark planar cantilever truss reported in Refs. [34, 35]. The rectangular design domain measures 10.16 m × 2.54 m. The two nodes on the left boundary are fully constrained, and the node at the lower-right corner is subjected to a vertically downward concentrated load of 444.8 kN (Fig. 9). The boundary-condition coordinates, material properties, and optimization constraints are summarized in Tables 1-3. The initial truss graph generated from this domain retains all prescribed nodes and establishes a continuous candidate member network connecting the supports to the load application point (Fig. 10(a)). After mapping to the parametric finite-element model in COMSOL, the linear static analysis converges successfully, yielding an initial structural mass of 3276.8 kg (Fig. 10(b)).

To evaluate the influence of stochastic initialization on the optimization results, we perform 20 independent PSO runs using identical initial truss graphs, parametric finite-element models, design-variable bounds, and termination criteria. Each run uses a swarm size of 30 particles and a maximum of 600 iterations and differs only in its initial particle population. All runs exhibit a consistent convergence pattern (Fig. 11): a rapid reduction in structural mass during the early iterations, followed by gradual stabilization. The final masses converge within a narrow range across the 20 runs. The best feasible design attains a mass of 1225.2 kg, corresponding to a 62.6% reduction relative to the initial mass.

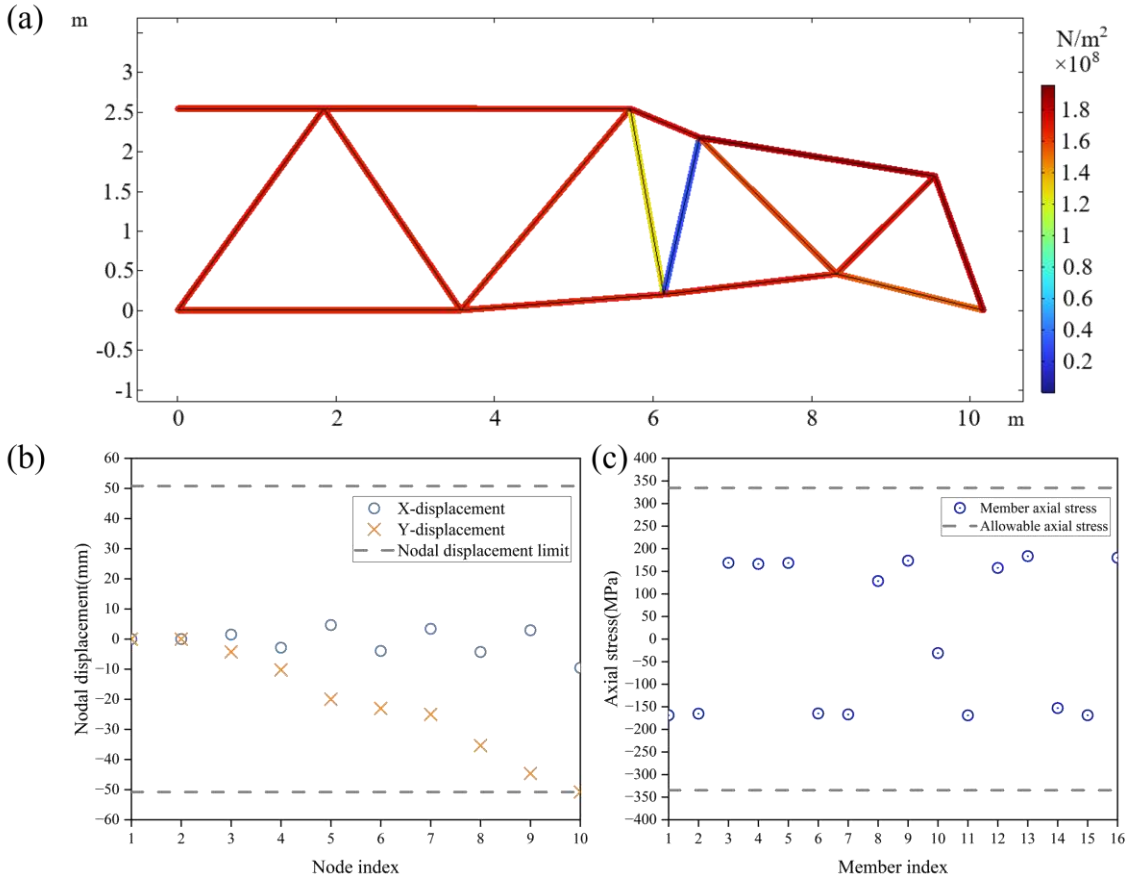


Figure 12: Best feasible design and corresponding finite-element responses for the two-dimensional example: (a) optimized design and distribution of member axial stresses; (b) nodal displacement components; and (c) member axial stresses.

The best feasible design is shown in Fig. 12 together with its finite-element responses. Larger cross-sectional areas are concentrated in members that contribute most to the global stiffness and axial load transfer, while the remaining members tend toward the prescribed lower area bound. Analysis of the constraint-utilization ratios across the 20 runs reveals that the vertical displacement constraint governs the design (Fig. 13). The maximum stress-utilization ratio and the horizontal displacement-utilization ratio remain substantially below unity, whereas the vertical displacement-utilization ratio is concentrated near unity. In the final design, the vertical displacement reaches its allowable limit, while the maximum member axial stress remains below the prescribed allowable stress.

Table 4 provides a contextual comparison between the best feasible result obtained in this study and values reported in related studies. The minimum structural masses reported by KR-UCT [34] and AutoTruss [35] are 1463.44 kg and 1378 kg, respectively, whereas the present method yields a feasible design with a mass of 1225.2 kg. Since the studies differ in design variables, structural scales, and some constraint settings, these results do not constitute a strictly like-for-like comparison and are presented mainly to indicate the mass level achieved in this study. The result demonstrates the potential of the proposed method for generating lightweight truss structures.

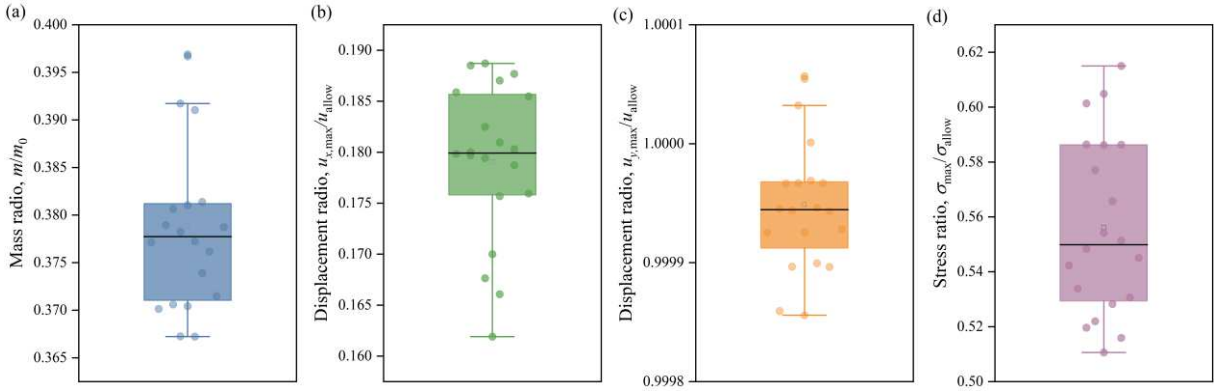


Figure 13: Statistical distributions of the final designs obtained from 20 independent runs: (a) normalized mass, m/m_0; (b) maximum x-direction displacement-utilization ratio, $u_{x,max}/u_{allow}$; (c) maximum y-direction displacement-utilization ratio, $u_{y,max}/u_{allow}$; (d) maximum axial-stress-utilization ratio, $\sigma_{y,max}/\sigma_{allow}$. The center lines and box boundaries indicate the medians and interquartile ranges, respectively; the whiskers extend to the most extreme observations lying within 1.5 times the interquartile range, while the superimposed markers represent the individual runs.

Table 4: Representative results for the two-dimensional planar cantilever truss benchmark

Method	Best design mass (kg)
KR-UCT [34]	1463.44
AutoTruss [35]	1378.00
Present method	1225.20

5.3 Three-Dimensional Spatial Cantilever Truss Example

We evaluate the three-dimensional cantilever space-truss benchmark from Refs. [34, 35]. The three nodes at the left end are fully constrained, and the node at the right end carries a vertically downward concentrated load of 50 N (Fig. 14). The boundary-node coordinates, material properties, and optimization constraints are summarized in Tables 5-7. Both the allowable tensile stress and allowable compressive stress are set to 0.6 times the yield strength. The initial truss graph retains the four prescribed nodes and establishes

a spatial candidate member network through three-dimensional Delaunay-based adjacency reconstruction (Fig. 15(a)). After mapping to COMSOL, the linear static analysis converges, yielding an initial mass of 11.2135 kg (Fig. 15(b)).

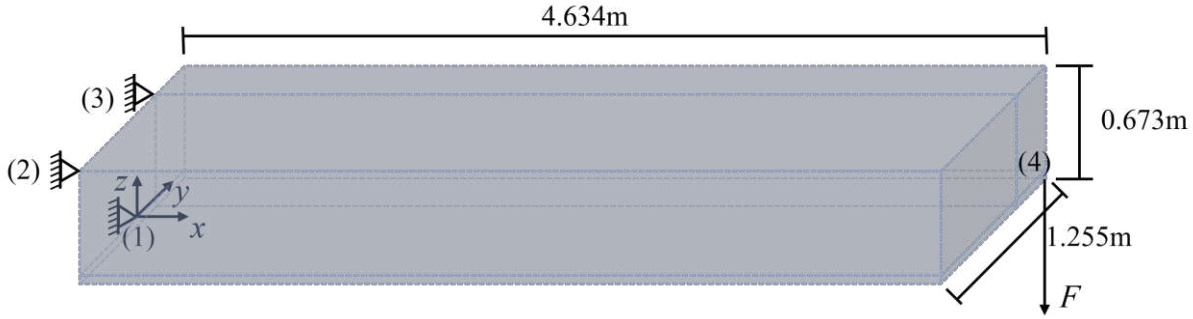


Figure 14: Design domain and boundary conditions of the three-dimensional spatial cantilever truss.

Table 5: Coordinates of the boundary-condition nodes for the three-dimensional spatial cantilever truss

Node type	Node coordinates (m)
Fixed-support nodes	(0.0, 0.0, 0.0), (0.0, -0.483, 0.595), (0.0, 0.483, 0.595)
Concentrated-load application point	(4.634, 0.772, -0.078)

Table 6: Material properties of the three-dimensional spatial cantilever truss

Material property	Value
Density	8000 kg/m^3
Young's modulus	193 GPa
Poisson's ratio	0.3
Yield strength	213 MPa

Table 7: Constraints and design-variable bounds for the three-dimensional spatial cantilever truss

Parameter	Value
Allowable axial stress	127.8 MPa
Nodal displacement limit	2 mm
Member cross-sectional-area range	1.11–30.28 cm^2

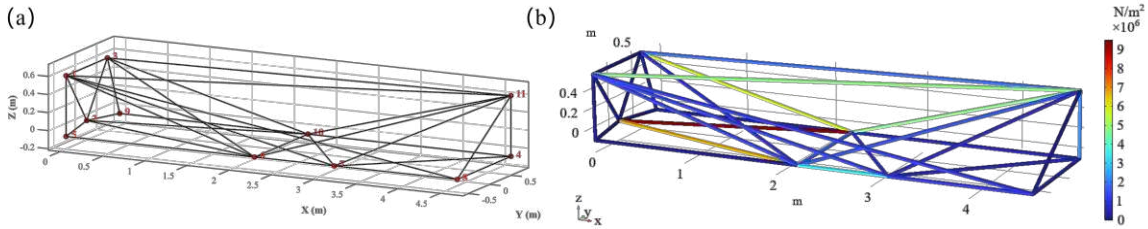


Figure 15: Initial spatial truss graph and finite-element response for the three-dimensional example: (a) initial spatial truss graph; and (b) distribution of member axial stresses in the initial finite-element model.

Consistent with the two-dimensional example, we perform 20 independent PSO runs using identical initial truss graphs, parametric finite-element models, design-variable bounds, and optimization parameters. Fig. 16 presents the mass-convergence histories. All runs show a rapid reduction in mass during the initial iterations, followed by progressively slower improvement. Among the 20 runs, the best feasible design achieves a mass of

2.5995 kg, corresponding to a 76.8% reduction relative to the initial mass.

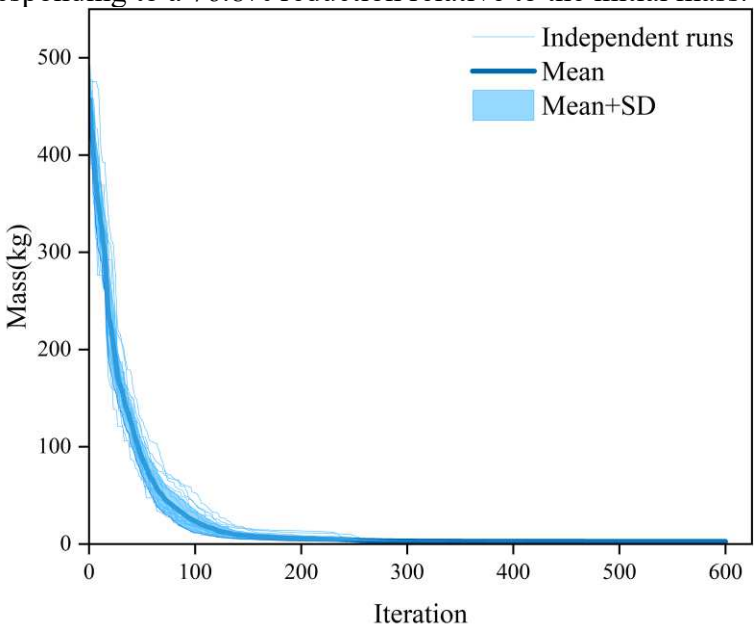


Figure 16: Convergence histories of the structural mass over 20 independent optimization runs for the three-dimensional example. The thin light-blue curves represent the individual optimization runs, the dark-blue curve denotes the mean mass, and the shaded band indicates the mean ± one sample standard deviation.

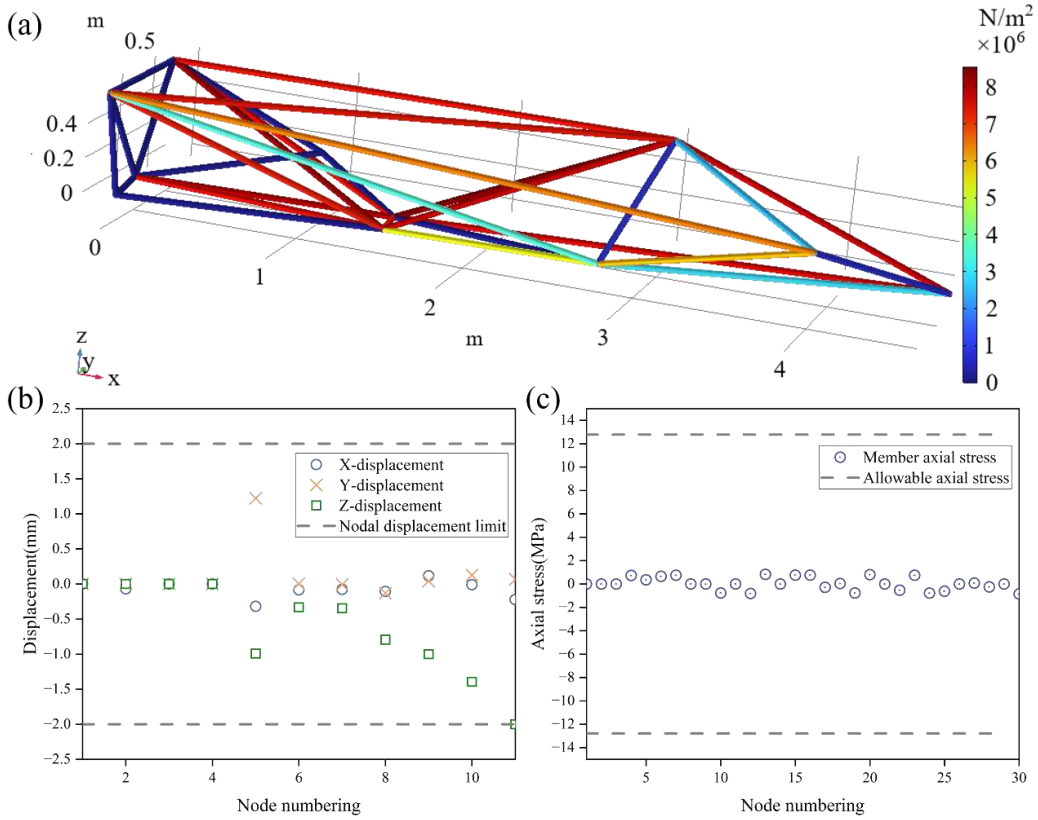


Figure 17: Best feasible design and corresponding finite-element responses for the three-dimensional example: (a) distribution of member axial stresses in the optimized design; (b) nodal displacements; and (c) member axial stresses.

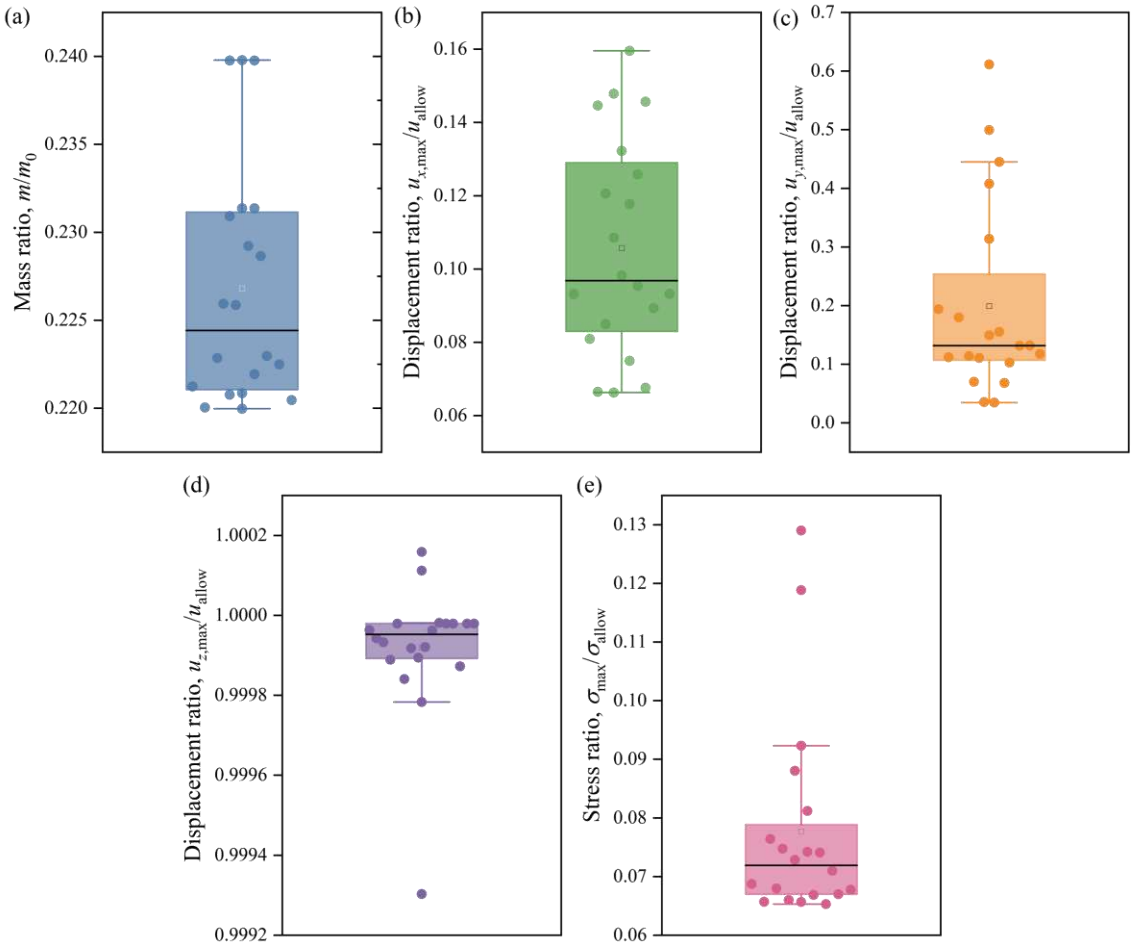


Figure 18: Statistical distributions of the normalized mass and constraint-utilization ratios obtained from 20 independent runs of the three-dimensional example: (a) mass ratio, m/m_0; (b) maximum x-direction displacement-utilization ratio, $u_{x,\max}/u_{\text{allow}}$; (c) maximum y-direction displacement-utilization ratio, $u_{y,\max}/u_{\text{allow}}$; (d) maximum z-direction displacement-utilization ratio, $u_{z,\max}/u_{\text{allow}}$;(e) maximum axial-stress-utilization ratio, $\sigma_{y,\max}/\sigma_{\text{allow}}$. The center lines and box boundaries indicate the medians and interquartile ranges, respectively; the whiskers extend to the most extreme observations lying within 1.5 times the interquartile range, while the superimposed markers represent the individual runs.

Fig. 17 shows the best spatial design (2.5995 kg) together with its finite-element responses. Larger cross-sectional areas are assigned to members that carry the principal internal forces, while the remaining members approach the prescribed lower bound (Fig. 17(a)). Analysis of the constraint-utilization ratios reveals that the z-direction displacement constraint governs the 3D design (Fig. 18). The stress-utilization and in-plane displacement ratios remain substantially below unity, whereas the z-direction ratio is concentrated near unity.

To obtain a clearer final layout, low-force members are identified as candidates for removal. A candidate member is removed only if its elimination does not compromise the three-dimensional bracing pattern or introduce a potential kinematic mechanism. The reconstructed finite-element model is subsequently reanalyzed to verify solvability and constraint compliance. The simplified design passes reanalysis with a mass of 2.5974 kg (Fig. 19). Ref. [34] reported 37.2 kg using KR-UCT, and Ref. [35] obtained 28.8 kg using AutoTruss, both with discrete circular hollow sections and additional constraints.

Using the KR-UCT method, Ref. [34] reported a minimum structural mass of 37.2 kg across the considered nodal configurations, whereas Ref. [35] obtained a minimum mass of 28.8 kg using AutoTruss. However, these studies employed discrete circular hollow sections and incorporated constraints on structural self-weight, member slenderness, member length, and the angles between adjacent members. In contrast, the present study redefined the design-domain boundaries, cross-sectional parameterization, and constraint set in accordance with the developed parametric finite-element and shape-size optimization framework. Accordingly, the reported results are used solely as a reference for the mass levels achieved under different modelling assumptions and optimization settings.

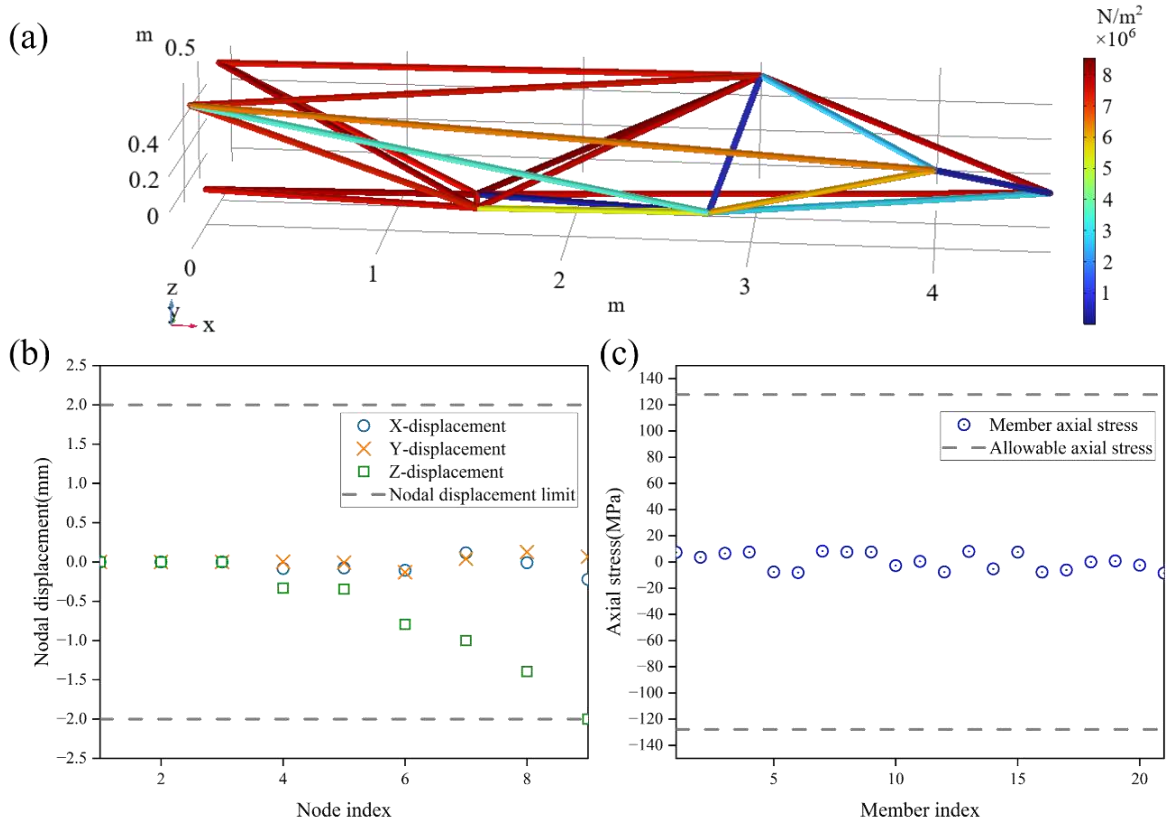


Figure 19: Spatial design after member-removal post-processing and the corresponding reanalysis results: (a) distribution of member axial stresses in the simplified design; (b) nodal displacements; and (c) member axial stresses.

Table 8: Summary of the principal optimization results for the two- and three-dimensional examples

Example	Number of independent runs	Initial mass(kg)	Best feasible mass (kg)	Mass reduction(%)	Governing constraint
Two-dimensional planar cantilever truss	20	3276.8	1225.2	62.6	Vertical displacement
Three-dimensional spatial cantilever truss	20	11.2135	2.5995	76.8	*z*-direction displacement

5.4 Summary of Results

Table 8 summarizes the key results. In both examples, the generated initial truss graphs were successfully mapped onto solvable parametric FE models, and substantial mass reductions were achieved. The best masses were reduced by 62.6% (2D) and 76.8% (3D). The vertical displacement was the governing constraint in both cases.

6 Conclusions

This study presents a domain-driven method for generating initial truss graphs without requiring a fully predefined node-member topology. The method integrates virtual-force relaxation, a target member-length field, boundary projection, and dynamic Delaunay adjacency reconstruction to transform a geometric design domain and a set of prescribed mechanical nodes into a valid truss graph. The generated graph is automatically mapped onto a parametric finite-element model in COMSOL Multiphysics, where particle swarm optimization simultaneously optimizes the free-node coordinates and member cross-sectional areas while preserving the generated connectivity.

The numerical results demonstrate that the generated graphs conform to the prescribed boundaries and local discretization scales of uniform, non-uniform, and perforated design domains, and successfully pass geometric-validity, connectivity, and parametric finite-element solvability checks. For the two- and three-dimensional examples considered herein, the best feasible masses obtained over 20 independent PSO runs were reduced by 62.6% (from 3276.8 kg to 1225.2 kg) and 76.8% (from 11.2135 kg to 2.5995 kg), respectively, while all prescribed stress and displacement constraints were satisfied. In both cases, the governing displacement constraint reached its allowable limit, whereas the member axial stresses retained discernible margins below the allowable values. Following member-removal post-processing and FEM reanalysis, the three-dimensional mass was further reduced from 2.5995 kg to 2.5974 kg. Comparison with the literature shows that the best feasible design obtained for the two-dimensional benchmark (1225.2 kg) improves upon the values reported for KR-UCT (1463.44 kg) and AutoTruss (1378 kg), confirming that the generated initial graph provides an effective foundation for subsequent shape-size optimization.

By updating the adjacency relationships during the generation stage while holding the edge set fixed during optimization, the proposed method balances configurational adaptability against FE interface stability. However, the attainable final design remains constrained by the connectivity of the initial graph. The current FE formulation is limited to linear elasticity, small-deformation kinematics, and ideal pin-jointed connections, with constraints restricted to stress, displacement, and design-variable bounds. Future work may incorporate compression-member buckling, slenderness-ratio restrictions, structural self-weight, multiple load cases, discrete standard cross-sections, and manufacturability constraints to enhance the engineering applicability of both the generated graphs and the resulting optimized designs.

CRediT authorship contribution statement

Shuai Yang: Writing – original draft, Software, Methodology, Formal analysis. **Yu-Xin Lin:** Software, Data curation. **Yi-Bo Song:** Validation, Investigation. **Shi-Ke Zhang:**

Validation, Supervision. **Meng-Meng Lian:** Writing – review & editing, Formal analysis, Conceptualization. **Xiao-Liang Zhou:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

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

Acknowledgements

The work was supported by National Natural Science Foundation of China (12502190), Joint Fund Project of Science and Technology Research and Development Program of Henan Province of China (252103810109), Key Scientifc and Technological Projects of Anyang of China (2025C01GX016).

Data availability

Data will be made available on request.

References

1. Michell AGM (1904) LVIII. *The limits of economy of material in frame-structures*. Lond Edinb Dublin Philos Mag J Sci 8:589–597. https://doi.org/10.1080/14786440409463229
2. Dorn WC, Gomory RE, Greenberg HJ (1964) Automatic design of optimal structures. J Mécanique 3:25–52
3. Bendsøe MP, Sigmund O (2003) Topology Optimization: Theory, Methods and Applications. Springer, Berlin, Heidelberg
4. Bendsøe MP (1989) Optimal shape design as a material distribution problem. Struct Optim 1:193–202. https://doi.org/10.1007/BF01650949
5. Bendsøe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. Comput Methods Appl Mech Eng 71:197–224. https://doi.org/10.1016/0045-7825(88)90086-2
6. Deaton JD, Grandhi RV (2014) A survey of structural and multidisciplinary continuum topology optimization: post 2000. Struct Multidiscip Optim 49:1–38. https://doi.org/10.1007/s00158-013-0956-z
7. Manguri A, Hassan H, Saeed N, Jankowski R (2025) Topology, Size, and Shape Optimization in Civil Engineering Structures: A Review. Comput Model Eng Sci 142:933–971. https://doi.org/10.32604/cmes.2025.059249
8. Sigmund O, Maute K (2013) Topology optimization approaches: A comparative review. Struct Multidiscip Optim 48:1031–1055. https://doi.org/10.1007/s00158-013-0978-6
9. Allaire G, Jouve F, Toader A-M (2004) Structural optimization using sensitivity analysis and a level-set method. J Comput Phys 194:363–393. https://doi.org/10.1016/j.jcp.2003.09.032
10. Huang X, Xie YM (2007) Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method. Finite Elem Anal Des 43:1039–1049. https://doi.org/10.1016/j.finel.2007.06.006
11. Querin OM, Steven GP, Xie YM (1998) Evolutionary structural optimisation (ESO) using a bidirectional algorithm. Engineering Computations 15:1031–1048.

https://doi.org/10.1108/02644409810244129
12. Sethian JA, Wiegmann A (2000) Structural Boundary Design via Level Set and Immersed Interface Methods. J Comput Phys 163:489–528. https://doi.org/10.1006/jcph.2000.6581
13. Sigmund O (2001) A 99 line topology optimization code written in Matlab. Struct Multidiscip Optim 21:120–127. https://doi.org/10.1007/s001580050176
14. Svanberg K (1987) The method of moving asymptotes—a new method for structural optimization. Int J Numer Methods Eng 24:359–373. https://doi.org/10.1002/nme.1620240207
15. Wang MY, Wang X, Guo D (2003) A level set method for structural topology optimization. Comput Methods Appl Mech Eng 192:227–246. https://doi.org/10.1016/S0045-7825(02)00559-5
16. Xie YM, Steven GP (1993) A simple evolutionary procedure for structural optimization. Comput Struct 49:885–896. https://doi.org/10.1016/0045-7949(93)90035-C
17. Yago D, Cante J, Lloberas-Valls O, Oliver J (2022) Topology Optimization Methods for 3D Structural Problems: A Comparative Study. Arch Comput Methods Eng 29:1525–1567. https://doi.org/10.1007/s11831-021-09626-2
18. Hu X, Li Z, Bao R, et al (2022) An adaptive method of moving asymptotes for topology optimization based on the trust region. Comput Methods Appl Mech Eng 393:114202. https://doi.org/10.1016/j.cma.2021.114202
19. Gilbert M, Tyas A (2003) Layout optimization of large-scale pin-jointed frames. Engineering Computations 20:1044–1064. https://doi.org/10.1108/02644400310503017
20. He L, Gilbert M (2015) Rationalization of trusses generated via layout optimization. Struct Multidiscip Optim 52:677–694. https://doi.org/10.1007/s00158-015-1260-x
21. Sokół T (2011) A 99 line code for discretized Michell truss optimization written in Mathematica. Struct Multidiscip Optim 43:181–190. https://doi.org/10.1007/s00158-010-0557-z
22. Zegard T, Paulino GH (2015) GRAND3 — Ground structure based topology optimization for arbitrary 3D domains using MATLAB. Struct Multidiscip Optim 52:1161–1184. https://doi.org/10.1007/s00158-015-1284-2
23. Zegard T, Paulino GH (2014) GRAND — Ground structure based topology optimization for arbitrary 2D domains using MATLAB. Struct Multidiscip Optim 50:861–882. https://doi.org/10.1007/s00158-014-1085-z
24. Cai Q, Xie YM, Li Y, et al (2026) Bi-material layout optimization of trusses with simultaneous topology–geometry design under stress constraints. Compos Struct 389:120397. https://doi.org/10.1016/j.compstruct.2026.120397
25. Canelas A, Carrasco M, López J (2024) Topology optimization of truss structures under failure probability using the Bernstein approximation. Comput Struct 296:107295. https://doi.org/10.1016/j.compstruc.2024.107295
26. Cui H, An H, Huang H (2018) Truss topology optimization considering local buckling constraints and restrictions on intersection and overlap of bar members. Struct Multidiscip Optim 58:575–594. https://doi.org/10.1007/s00158-018-1910-x
27. Lai Y, Cai Q, Li Y, et al (2025) A new evolutionary topology optimization method for truss structures towards practical design applications. Eng Struct 324:119326. https://doi.org/10.1016/j.engstruct.2024.119326

28. Tang X, Wen X, Zhang G, et al (2026) Augmented Lagrangian-based truss topology optimization with stability, stress, and geometric feasibility constraints. Comput Struct 326:108212. https://doi.org/10.1016/j.compstruc.2026.108212
29. Marzok A (2026) Simultaneous layout and shape optimization of thin-walled beams with buckling considerations. Thin-Walled Struct 219:114206. https://doi.org/10.1016/j.tws.2025.114206
30. Li Z, Hu X, Chen W (2025) 3D Stress-constrained topology optimization via stabilized time-series moving morphable components approach. Struct Multidiscip Optim 68:211. https://doi.org/10.1007/s00158-025-04123-7
31. Li Z, Hu X, Chen W (2023) Moving morphable curved components framework of topology optimization based on the concept of time series. Struct Multidiscip Optim 66:19. https://doi.org/10.1007/s00158-022-03472-x
32. Singh H, Sanders ED (2026) Automated lattice embedding and transitioning with application to multi-lattice topology optimization. Eng Comput 42:208. https://doi.org/10.1007/s00366-026-02441-y
33. Fenton M, McNally C, Byrne J, et al (2016) Discrete Planar Truss Optimization by Node Position Variation Using Grammatical Evolution. IEEE Trans Evol Comput 20:577–589. https://doi.org/10.1109/TEVC.2015.2502841
34. Luo R, Wang Y, Liu Z, et al (2022) A Reinforcement Learning Method for Layout Design of Planar and Spatial Trusses using Kernel Regression. Appl Sci 12:8227. https://doi.org/10.3390/app12168227
35. Du W, Zhao J, Yu C, et al (2023) Automatic Truss Design with Reinforcement Learning. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence. International Joint Conferences on Artificial Intelligence Organization, Macau, SAR China, pp 3659–3667. https://doi.org/10.24963/ijcai.2023/407
36. Ariyasinghe N, Wickremasinghe T, Weeratunge H, et al (2026) Topology and size optimization of trusses using graph neural networks: Towards efficient surrogate modeling. Structures 88:111934. https://doi.org/10.1016/j.istruc.2026.111934
37. Kupwiwat C, Hayashi K, Ohsaki M (2024) Multi-objective optimization of truss structure using multi-agent reinforcement learning and graph representation. Eng Appl Artif Intell 129:107594. https://doi.org/10.1016/j.engappai.2023.107594
38. Kupwiwat C, Ohsaki M (2026) A unified evaluation framework for reinforcement learning paradigms in bi-objective truss optimization. Adv Eng Inform 74:104750. https://doi.org/10.1016/j.aei.2026.104750
39. Liu J, Xia Y (2022) A hybrid intelligent genetic algorithm for truss optimization based on deep neutral network. Swarm Evol Comput 73:101120. https://doi.org/10.1016/j.swevo.2022.101120
40. Xia Y, Liu J, Qi H (2024) A population-based DNN-augmented optimization method for designing truss structures. Swarm Evol Comput 89:101613. https://doi.org/10.1016/j.swevo.2024.101613
41. Zheng S, Qiu L, Lan F (2023) TSO-GCN: A Graph Convolutional Network approach for real-time and generalizable truss structural optimization. Appl Soft Comput 134:110015. https://doi.org/10.1016/j.asoc.2023.110015
42. Zhu S, Ohsaki M, Hayashi K, Guo X (2021) Machine-specified ground structures for topology optimization of binary trusses using graph embedding policy network. Adv Eng Softw 159:103032. https://doi.org/10.1016/j.advengsoft.2021.103032

43. Mai HT, Lieu QX, Kang J, Lee J (2023) A novel deep unsupervised learning-based framework for optimization of truss structures. Eng Comput 39:2585–2608. https://doi.org/10.1007/s00366-022-01636-3
44. Chen H, Liu X, Wang L, Kara LB (2025) Multi-scale topology optimization using neural networks. Eng Comput 41:3313–3327. https://doi.org/10.1007/s00366-025-02156-6
45. Zhu G, Hu X, Bao R, Chen W (2025) Continuous high-throughput characterization of mechanical properties via deep learning. Int J Mech Sci 291–292:110137. https://doi.org/10.1016/j.ijmecsci.2025.110137
46. Wang Y, Chu J, Yuan J, et al (2025) Accurate prediction of contact stiffnesses of irregular interfaces. Int J Mech Sci 308:110969. https://doi.org/10.1016/j.ijmecsci.2025.110969
47. An J, Park JS, Park HS (2025) Integrated truss optimization using a ternary-quantized gradient method with implicit topology control. Comput-Aided Civ Infrastruct Eng 40:6437–6455. https://doi.org/10.1111/mice.70158
48. Cao H, Sun W, Chen Y, et al (2023) Sizing and shape optimization of truss employing a hybrid constraint-handling technique and manta ray foraging optimization. Expert Syst Appl 213:118999. https://doi.org/10.1016/j.eswa.2022.118999
49. He S-X, Cui Y-T (2023) Medalist learning algorithm for configuration optimization of trusses. Appl Soft Comput 148:110889. https://doi.org/10.1016/j.asoc.2023.110889
50. Kaveh A, Raissi Dehkordi M, Shaghaghian AH (2025) A neuro-gradient evolution method for simultaneous size and layout optimization of truss structures in the combined domain of variables. Structures 79:109469. https://doi.org/10.1016/j.istruc.2025.109469
51. Lai Y, Liu G, Xie YM (2025) A new nodal shift-based evolutionary structural optimization method for truss design. Structures 81:110202. https://doi.org/10.1016/j.istruc.2025.110202
52. Park JS, Hong T, Lee DE, Park HS (2024) Constraint-aware optimization model for plane truss structures via single-agent gradient descent. Comput-Aided Civ Infrastruct Eng 39:2737–2759. https://doi.org/10.1111/mice.13226
53. Dong Y, Yuan X, Wang X, et al (2025) Simultaneous topology, configuration, and prestress optimization for lightweight design of modular tensegrity chain structures. Thin-Walled Struct 212:113184. https://doi.org/10.1016/j.tws.2025.113184
54. Yang H, Hu J, Gao R, Liu S (2026) A co-optimization method for stiffener layout and skin shape of complex curved thin-walled structures via multi-patch isogeometric analysis. Thin-Walled Struct 225:114724. https://doi.org/10.1016/j.tws.2026.114724
55. Clerc M, Kennedy J (2002) The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput 6:58–73. https://doi.org/10.1109/4235.985692
56. Fourie PC, Groenwold AA (2002) The particle swarm optimization algorithm in size and shape optimization. Struct Multidiscip Optim 23:259–267. https://doi.org/10.1007/s00158-002-0188-0
57. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks, vol 4. IEEE, Perth, WA, Australia, pp 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
58. Poli R, Kennedy J, Blackwell T (2007) Particle swarm optimization: An overview. Swarm Intell 1:33–57. https://doi.org/10.1007/s11721-007-0002-0
59. Schutte JF, Groenwold AA (2003) Sizing design of truss structures using particle swarms. Struct Multidiscip Optim 25:261–269. https://doi.org/10.1007/s00158-003-0316-

60. Venter G, Sobieszczanski-Sobieski J (2003) Particle Swarm Optimization. AIAA J 41:1583–1589. https://doi.org/10.2514/2.2111
61. Azizi M, Aickelin U, Khorshidi HA, Shishehgarkhaneh MB (2022) Shape and size optimization of truss structures by Chaos game optimization considering frequency constraints. J Adv Res 41:89–100. https://doi.org/10.1016/j.jare.2022.01.002
62. Contreras-Bejarano O, Villalba-Morales JD (2024) On the use of the differential evolution algorithm for truss-type structures optimization. Appl Soft Comput 161:111372. https://doi.org/10.1016/j.asoc.2024.111372
63. Salajegheh F, Salajegheh E, Shojaee S (2023) Optimum design of truss structures with frequency constraints by an enhanced particle swarm optimization method with gradient directions based on emigration philosophy. Eng Optim 55:434–456. https://doi.org/10.1080/0305215X.2021.2011259
64. Hu H, Song W, Song G, Li Y (2023) Optimization design and parametric research of belt conveyor deck truss structure. Structures 57:105155. https://doi.org/10.1016/j.istruc.2023.105155
65. Ren X, Xi R, Kang Z, et al (2026) Optimization study on the parameter design of double-layer steel truss bridges based on the response surface method. KSCE J Civ Eng 30:100436. https://doi.org/10.1016/j.kscej.2025.100436
66. Li Y, Zhang Z, Chen H, et al (2026) Sizing optimization of spatial pipe trusses based on an enhanced superb Fairy-wren optimization algorithm. Structures 90:112486. https://doi.org/10.1016/j.istruc.2026.112486