

ARTICLE

Spline-Based Shape Optimization of Laminated Composite Shells Using a Surrogate-Assisted BCMO

Hoang-Anh Pham^{1,2,*}, Van-Binh Tran³, Tien-Chuong Vu¹ and Minh-Tu Tran^{1,2}

¹Faculty of Building and Industrial Construction, Hanoi University of Civil Engineering, Hanoi, Vietnam

²Frontier Research Group of Mechanics of Advanced Materials and Structures (MAMS), Hanoi University of Civil Engineering, Hanoi, Vietnam

³Department of Technology and Engineering, Ha Tinh University, Ha Tinh, Vietnam

*Corresponding Author: Hoang-Anh Pham. Email: anhph2@huce.edu.vn

Received: 05 July 2026; Accepted: 17 September 2026; Published: 21 October 2026

ABSTRACT: Composite shell shape optimization increasingly relies on the seamless convergence of composite mechanics, advanced geometric modeling, and robust optimization algorithms. Approaches based on Isogeometric Analysis (IGA) incur high implementation complexity for non-conforming multi-patch configurations, whereas traditional Finite Element Method (FEM) using nodal coordinates as design variables leads to excessive dimensionality and geometrically unrealistic shapes. Furthermore, gradient-based optimization techniques struggle with the highly non-linear, multimodal landscapes inherent to shell shape optimization, while standard metaheuristic approaches demand prohibitive computational costs. To address these challenges, this paper proposes an efficient and integrated Python-based workflow for the shape optimization of composite shells. The shell geometry is parameterized using spline curves, where control point coordinates serve as design variables to significantly reduce the design space dimension compared to nodal-based FEM. The optimization objective focuses on minimizing total strain energy under external loads. For the structural analysis stage, a 3-node triangular flat shell element (CS-DSG3-A) is deployed based on the first-order shear deformation theory, which effectively integrates the smoothed discrete shear gap method with Allman membrane strains including rotational degrees of freedom. Geometric modeling and mesh generation are dynamically coupled via the Gmsh API in Python. To accelerate the optimization process, the Balancing Composite Motion Optimization (BCMO) algorithm is integrated with a k-Nearest Neighbor Comparison (k-NNC) classification model, which filters out unpromising candidates and drastically reduces the computational burden. Numerical examples encompassing both isotropic and laminated composite shells demonstrate that the proposed framework effectively achieves optimal shell shapes dominated by membrane action. The integration of BCMO and k-NNC delivers a highly unified pipeline that significantly reduces the computational cost, offering a viable efficiency-accuracy trade-off where substantial computational savings are achieved with a manageable compromise on the global optimum in complex design landscapes.

KEYWORDS: Laminated composite shells; shape optimization; CS-DSG3-A element; Balancing Composite Motion Optimization (BCMO); k-nearest neighbor comparison

1 Introduction

Laminated composite shell structures are increasingly being applied in industries requiring high stiffness-to-weight ratios, including aerospace, wind energy, shipbuilding, and civil engineering. Compared to traditional homogeneous materials, laminated composites allow engineers to simultaneously adjust

geometric and mechanical parameters, including the surface shape of the shell, layer thickness, and fiber angle, to achieve desired structural performance. Among these design problems, shape optimization plays a particularly important role, as the shell shape directly affects the load transfer mechanism: an optimized shape will shift most internal forces to the membrane-dominated state, minimizing local bending and maximizing the superior mechanical properties of the composite. This problem, however, presents simultaneous challenges in three aspects: structural analysis, geometric modeling, and optimization algorithms, which need to be addressed.

In terms of structural analysis, the finite element method (FEM) based on first-order shear deformation theory (FSDT) and higher-order shear deformation theory (HSDT) is the main foundation for composite shells [1,2]. Among the shell element classes, the 3-node triangular flat shell element is favored in shape optimization problems because of its ability to create flexible meshes on complex geometries and continuously changing in the optimization loop. However, this type of element faces two prominent problems: shear locking when the thickness/span ratio is small, and lack of rotational degrees of freedom (DOF) in the element plane (drilling DOF), leading to a potentially degenerate stiffness matrix when assembling flat elements into a general curved shell structure. The Discrete Shear Gap (DSG) technique [3] and the cell-based smoothing variant (CS-DSG3) [4] have proven to be superior in eliminating shear locking for this element. From the original formulation for isotropic materials [4], CS-DSG3 has been extended to several more complex structures, including laminated composite shells [5], functionally graded material (FGM) shells [6]. However, in these studies, the rotational degrees of freedom treatments were performed artificially and the membrane deformation still relied on the constant strain field of the linear triangle. This limits the membrane's performance on deformed meshes and reduced accuracy in problems dependent on membrane or complex shell shapes. To improve the membrane performance of 3-node triangular elements, a key strategy is to incorporate rotational degrees of freedom, i.e., rotation around the element's normal axis, at each node. Allman [7] and later Cook [8,9] pioneered the proposal to add drilling DOF to the membrane displacement components to eliminate the stiffness matrix degeneracy of the triangular flat shell element. The incorporation of rotational DOF into CS-DSG3 (e.g., integration with Allman membrane strain) has also been performed for several plate and shell structures, namely: stiffened isotropic fold plates [10], laminated composite fold plates [11], and stiffened flat shells [12]. Recent developments have pursued the integration of improved Allman membrane strain with true rotational DOF into advanced shell elements. Chau-Dinh [13] proposed the CS-MITC18+ element, in which the Allman membrane strain and the bending strain are smoothed in the manner of CS-DSG3 technique. The CS-MITC18+ element has been extended for laminated composite plate and shell analysis [14]. Le-Phuong and Chau-Dinh developed the CS-MITC3+ flat shell element with Allman membrane strain for GPLRC-reinforced FGM shells [15]. In addition, Nguyen [16] proposed the κ MITC3+ element combining MITC3+ theory for bending/shearing and the improved Allman-type triangular element for membrane behavior. However, to date, CS-DSG3 as well as the aforementioned triangular shell elements have not been implemented in the shape optimization problem, where the element must operate stably and efficiently on thousands of meshes with continuously changing geometry in the optimization loop.

Besides traditional FEM, Isogeometric Analysis (IGA) [17] has emerged as a unified approach between geometric modeling and structural analysis through the direct use of B-spline/NURBS (Non-Uniform Rational B-Spline) functions as approximation functions [18]. IGA proves particularly attractive for shell shape optimization problems because NURBS control points are also design variables, creating a natural combination of geometric and analytical modeling [19]. However, with incompatible multi-patch configurations, common in practical engineering shell structures, IGA requires complex connection processing such as the Nitsche method [20,21], Interpolation for Nonconforming Decompositions (INTERNODES) [22], or

free-form deformation (FFD) [23], significantly increasing implementation complexity. Furthermore, the integration of IGA with open-source mesh generation tools remains limited compared to the traditional FEM ecosystem, impacting the reproducibility and scalability of optimal frameworks.

The second challenge, closely related to the analysis, is the selection of geometric parameterization, which directly determines the dimension of the design variable space and the geometric practicality of the optimal solution. The simplest approach is to directly use FEM node coordinates as design variables (e.g., [24]) which results in extremely large variable spaces and often generates non-smooth shapes due to a lack of continuity constraints. To overcome this, smooth basis function-based parameterization methods have been developed, including B-spline and NURBS [25–27], radial basis functions (RBF) [28,29]. Among these, spline-based parameterization with control points as design variables allows for a significant reduction in the design dimension space while ensuring the necessary geometric continuity for the shell. Notably, the approach of combining spline parameterization with traditional FEM, i.e., completely separating geometric modeling and analysis, linked via an automatic mesh generation step, is less exploited than IGA in studies on laminated composite shell optimization [30]. This approach allows simultaneous exploitation of the advantages of dimensionality reduction and smoothness from splines, while still inheriting the entire rich FEM element library (e.g., combining the software ABAQUS as done in [26]).

The third challenge, and often the decisive computational bottleneck, is the selection and acceleration of the optimal algorithm. The composite shell shape optimization problem is often a multi-modal problem with an objective function that is not differentiable with respect to the geometry variable, so gradient-based methods are prone to getting stuck at local extrema and require complex sensitivity analysis as the geometry changes continuously [31]. Metaheuristic algorithms (e.g., Genetic Algorithm (GA) [32], Particle Swarm Optimization (PSO) [33], and Differential Evolution (DE) [34]) do not require gradient information and have better global search capabilities, but have high computational costs due to the requirement of many iterations with thousands of full FEM analyses. Among metaheuristic algorithms, Balancing Composite Motion Optimization (BCMO) [35] is a relatively new algorithm that effectively solves complex optimization problems. As shown in [35], BCMO outperforms several metaheuristic algorithms by utilizing a parameter-free design that eliminates the need for manual tuning. Grounded in the laws of Newtonian mechanics, BCMO balances global exploration and local exploitation through a unique probabilistic equilibrium strategy. This simplified mathematical structure enables significantly faster convergence speeds and lowers computational overhead, especially in high-dimensional spaces. Ultimately, it also offers a more predictable, efficient, and user-friendly optimization process than several nature-inspired counterparts. BCMO has shown competitive performance on many engineering problems [36–39], including composite plate optimization [40,41]. However, BCMO still faces high computational costs when the number of objective function evaluations, each corresponding to a full FEM problem, reaches thousands of times during the optimization process.

To reduce computational costs when optimizing with metaheuristics, two main approaches have been pursued in research. The first approach involves building surrogate models that approximate the objective function, such as Kriging [42], Radial Basis Function Network [43], and machine learning models such as Artificial Neural Network (ANN) [40,44] or Support Vector Machine (SVM) [45]; however, surrogates require expensive training processes and accuracy can be reduced when the design space has a complex landscape as in the shell shape optimization problem. The second approach is simpler in terms of implementation and uses a binary classifier to filter out unpromising candidates before calling the full FEM, instead of approximating the entire objective function. Among the classifiers, k-Nearest Neighbor Comparison (k-NNC) [46] has the advantage of not requiring explicit training, not requiring complex hyper parameter adjustments, and instantaneous updating when new data is available in the optimization loop. In Pham et al. [46] and Nguyen et al. [47], k-NNC has been successfully integrated into metaheuristic algorithms for

truss structure optimization problems, showing the ability to significantly reduce the number of objective function evaluations without significantly affecting the quality of the solution. However, extending k-NNC to composite shell shape optimization problems, with continuous geometric design space and objective function landscape more complex than truss size optimization, and specific integration with BCMO has not been done in the existing literature.

From the above overview, it can be seen that although each component: shell element models with rotational degrees of freedom (e.g., CS-DSG3, CS-MITC18+, or κ MITC3+), spline parameterization, BCMO, and k-NNC, has been studied, their synchronous integration into a unified composite shell shape optimization framework remains an unfilled gap. Specifically, three gaps need to be addressed: (i) CS-DSG3 has not been implemented in the shape optimization loop; (ii) k-NNC has not been integrated with BCMO and has not been verified in the continuous geometric design space of the shell optimization problem; and (iii) a complete automated Python workflow integrating automatic mesh generation with analysis and optimization does not exist in the literature. To fill these gaps, this paper proposes an efficient and integrated framework for the shape optimization problem of laminated composite shells with the following main contributions:

- The CS-DSG3 element incorporating Allman membrane deformation (named CS-DSG3-A) is implemented in the shape optimization workflow of laminated composite shells.
- The k-NNC model [47] is integrated with BCMO, forming a BCMO/k-NNC algorithm capable of filtering out unpromising candidates before calling FEM analysis, thereby significantly reducing overall computational costs.
- The k-NNC model is applied to the continuous geometric design space of the shape optimization problem, extending its application scope from optimizing the size of truss structures.
- The shell shape optimization workflow integrated with the Gmsh API [48] for automatic mesh generation is implemented synchronously in Python, ensuring reproducibility and scalability for the research community.

The effectiveness of the framework is verified through numerical examples with isotropic shells and laminated composite shells, focusing on three aspects: the accuracy of CS-DSG3-A in the context of continuously changing geometry, the quantitative acceleration efficiency of k-NNC when integrated with BCMO, and the optimal shape quality achieved. The remainder of the paper is organized as follows: [Section 2](#) presents the CS-DSG3-A element theory for laminated composite shells within the FSDT framework. [Section 3](#) describes the spline geometry parameterization method and the automatic mesh generation process via the Gmsh API. [Section 4](#) presents the BCMO algorithm and the k-NNC integration mechanism. [Section 5](#) proposes the overall workflow of the framework. [Section 6](#) presents and discusses the numerical results. [Section 7](#) provides conclusions and directions for further research.

2 CS-DSG3-A Element Formulation for Laminated Composite Shells

This section presents the theoretical basis of the CS-DSG3-A 3-node triangular flat shell element within the framework of first-order shear deformation theory (FSDT). The element is constructed by integrating two independent components: (i) the CS-DSG3 plate element to describe bending and shear behaviors; and (ii) the Allman membrane element with in-plane rotational degrees of freedom (drilling DOF) to describe membrane behavior. This combination allows the element to fully describe the stress-strain state of a generalized curved laminated composite shell, while eliminating two degenerative phenomena: shear locking and stiffness matrix degeneracy due to the lack of drilling DOF.

2.1 Element Model

In the flat shell element model, the curved shell structure is approximated by a set of flat elements. Each 3-node triangular element is placed in the element's local xyz coordinate system, with the xy plane coinciding with the element plane and the z-axis perpendicular to the element (Fig. 1). This shell element is a combination of a 9-degree-of-freedom membrane element (Fig. 1a) and a 9-degree-of-freedom plate element (Fig. 1b). At each node, there are six DOFs: three linear displacements u , v , w and three rotational displacements β_x , β_y , β_z . Therefore, the element's DOF vector takes the form:

$$\mathbf{u}_e = \begin{bmatrix} \mathbf{u}_i & \mathbf{v}_i & \beta_{zi} & \mathbf{w}_i & \beta_{xi} & \beta_{yi} \end{bmatrix} \quad (1)$$

where $\mathbf{u}_i = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}$, $\mathbf{v}_i = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}$ are membrane displacements; $\mathbf{w}_i = \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix}$, $\beta_{xi} = \begin{bmatrix} \beta_{x1} & \beta_{x2} & \beta_{x3} \end{bmatrix}$, $\beta_{yi} = \begin{bmatrix} \beta_{y1} & \beta_{y2} & \beta_{y3} \end{bmatrix}$ are the bending and rotational displacements according to Reissner-Mindlin theory; and $\beta_{zi} = \begin{bmatrix} \beta_{z1} & \beta_{z2} & \beta_{z3} \end{bmatrix}$ are the drilling DOFs—degrees of freedom of rotation around the z-axis of the element, triggered by Allman membrane strain [7].

The overall element stiffness matrix in the local coordinate system has an 18×18 structure reflecting the interaction between membrane behavior and bending behavior:

$$\mathbf{K}_e = \begin{bmatrix} \mathbf{K}_m & \mathbf{K}_{mb} \\ \mathbf{K}_{bm} & \mathbf{K}_b \end{bmatrix} \quad (2)$$

where $\mathbf{K}_m$ is the 9×9 membrane stiffness matrix; $\mathbf{K}_b$ is the 9×9 bending-shear stiffness matrix, and $\mathbf{K}_{mb} = \mathbf{K}_{bm}^T$ is the 9×9 membrane-bending interaction stiffness matrix. This interaction matrix is only non-zero when the material layers are arranged asymmetrically across the normal plane (asymmetric laminate). Each matrix component in Eq. (2) is constructed by its own method presented in the following sections.

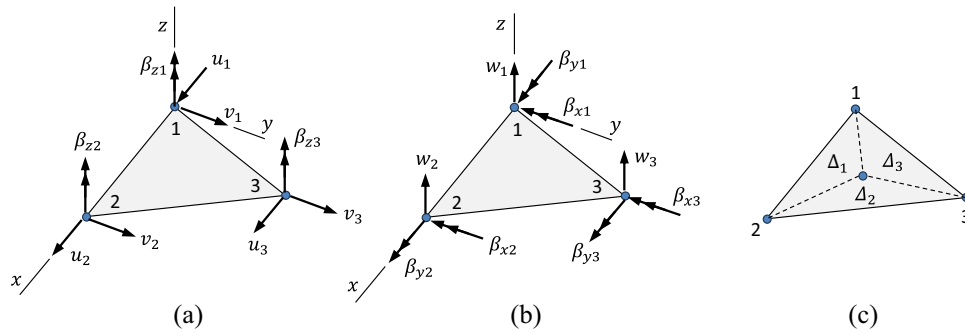


Figure 1: 3-node triangular element in local coordinate system: (a) Membrane DOFs; (b) Bending DOFs; (c) Three sub-triangles.

2.2 CS-DSG3 Plate Element for Bending Behavior

The CS-DSG3 (Cell-based Smoothed Discrete Shear Gap) plate element proposed by Nguyen-Thoi et al. [49] is based on a cell-based smoothing technique combined with a discrete shear gap (DSG) method. Each triangular element is divided into three sub-triangles Δ_1 , Δ_2 , Δ_3 sharing a common node at the centroid, as illustrated in Fig. 1c. The smoothing technique is applied by averaging the gradient matrices of the three sub-triangles by area, eliminating shear locking without the need for a Gaussian integral point.

The bending-shear stiffness matrix of the element is constructed as follows:

$$\mathbf{K}_b = \Delta \left(\bar{\mathbf{B}}_b^T \mathbf{D}_b \bar{\mathbf{B}}_b + \bar{\mathbf{B}}_s^T \mathbf{D}_s \bar{\mathbf{B}}_s \right) \quad (3a)$$

$$\bar{\mathbf{B}}_b = \frac{1}{\Delta} \sum_{i=1}^3 \Delta_i \mathbf{B}_{bi} \quad (3b)$$

$$\bar{\mathbf{B}}_s = \frac{1}{\Delta} \sum_{i=1}^3 \Delta_i \mathbf{B}_{si} \quad (3c)$$

where Δ is the total area of the element; Δ_i ($i = 1, 2, 3$) is the area of the i -th sub-triangle; $\mathbf{B}_{bi}$ and $\mathbf{B}_{si}$ are the bending gradient matrix and the shear gradient matrix of the i -th sub-triangle, respectively, determined by the DSG3 method [49]; $\mathbf{D}_b$ and $\mathbf{D}_s$ are the bending and shear elastic matrix of the composite material, as presented in Section 2.3.

2.3 Elastic Matrix of Laminated Composite Materials

Consider a composite shell formed from n layers of orthotropic material (Fig. 2), where the j -th layer has normal coordinates in the range $[z_{j-1}, z_j]$ and the fiber angle θ_j with respect to the x -axis of the element. In the FSDT framework, the bending $\mathbf{D}_b$ and shear $\mathbf{D}_s$ elastic matrices are determined by integrating over the thickness:

$$\mathbf{D}_b = \sum_{j=1}^n \int_{z_{j-1}}^{z_j} \bar{\mathbf{Q}}_m z^2 dz; \mathbf{D}_s = k \frac{h^2}{(h^2 + \alpha l_e^2)} \sum_{j=1}^n \int_{z_{j-1}}^{z_j} \bar{\mathbf{Q}}_s dz \quad (4)$$

where $k = 5/6$ is the Reissner shear correction factor; h is the total thickness of the shell; l_e is the characteristic edge length of the element. The term $\frac{h^2}{(h^2 + \alpha l_e^2)}$ is included to improve the accuracy and shear stability when the h/l_e ratio is small, and $\alpha = 0.1$ is the stability constant [49]. The stabilization parameter α is fixed at 0.1 following the mathematically validated shell element formulations by Bischoff and Bletzinger [50]. Within the automated optimization loop, the potential dependency of this stabilization term on local mesh dimensions l_e during geometry updates is suppressed by enforcing a uniform mesh discretization. This is achieved by keeping the characteristic element length parameter lc strictly constant within the Gmsh API across all iterations. Consequently, the local element sizes are kept consistently with the boundary transformations, minimizing the influence of mesh density variations.

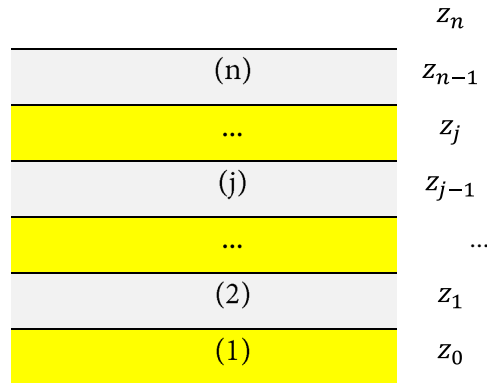


Figure 2: Layered structure of the laminated composite shell.

The transformed matrices $\bar{\mathbf{Q}}_m$ and $\bar{\mathbf{Q}}_s$, representations of material properties of the orthotropic composite layer in the element coordinate system, are determined by [1]:

$$\bar{\mathbf{Q}}_m = \mathbf{T}_m \mathbf{Q}_m \mathbf{T}_m^T; \bar{\mathbf{Q}}_s = \mathbf{T}_s \mathbf{Q}_s \mathbf{T}_s^T \quad (5a)$$

where $\mathbf{Q}_m$ and $[\mathbf{Q}_s]$ are the material elastic matrices in the principal material coordinate system of the layer:

$$\mathbf{Q}_m = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{33} \end{bmatrix}; \mathbf{Q}_s = \begin{bmatrix} Q_{44} & 0 \\ 0 & Q_{55} \end{bmatrix} \quad (5b)$$

with elastic components:

$$Q_{11} = \frac{E_1}{(1 - \nu_{21}\nu_{12})}, Q_{12} = \frac{\nu_{12}E_2}{(1 - \nu_{21}\nu_{12})}, Q_{22} = \frac{E_2}{(1 - \nu_{21}\nu_{12})}$$

$$Q_{33} = G_{12}, Q_{44} = G_{23}, Q_{55} = G_{13}$$

where E_1, E_2 are the elastic moduli in the fiber direction and perpendicular to the fiber; G_{12}, G_{13}, G_{23} are the shear moduli; ν_{12}, ν_{21} are the Poisson's ratios with the symmetry relationship $\nu_{12}E_2 = \nu_{21}E_1$. The transformation matrix $\mathbf{T}_m$ and $\mathbf{T}_s$ are given by:

$$\mathbf{T}_m = \begin{bmatrix} c^2 & s^2 & -2sc \\ s^2 & c^2 & 2sc \\ sc & -sc & c^2 - s^2 \end{bmatrix}; \mathbf{T}_s = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \quad (6)$$

with $c = \cos\theta, s = \sin\theta$, where θ is the fiber orientation angle of the material layer with respect to the x -axis of the element.

2.4 Allman Membrane Element with Drilling DOF

Allman membrane [7] adds rotational degrees of freedom β_z in the plane at each node of the triangular element, creating a membrane element with 9 DOFs (3 nodes $\times$ 3 DOFs/node: u, v, β_z). Drilling DOF β_z is activated through an extended linear strain displacement field, where the form function for β_z includes quadratic terms at the mid-edge. This addition not only eliminates stiffness matrix degeneracy when fitting a planar element into a curved shell structure, but also improves the accuracy of membrane behavior for angled shells and generalized curved shells. The element's membrane stiffness matrix is constructed in the form:

$$\mathbf{K}_m = \int_{\Delta} \mathbf{B}_m^T \mathbf{D}_m \mathbf{B}_m d\Delta \quad (7a)$$

The Allman membrane gradient matrix $\mathbf{B}_m$ relating membrane strain to the extended nodal displacement vectors including drilling DOFs has the form:

$$\mathbf{B}_m = \mathbf{B}_0 \mathbf{C} \quad (7b)$$

where $\mathbf{B}_0$ is explicitly expressed as

$$\mathbf{B}_0 = \begin{bmatrix} 1 & y & 0 & 0 & 0 \\ 0 & 0 & 1 & x & 0 \\ 0 & -x & 0 & -y & 1 \end{bmatrix} \quad (7c)$$

and matrix $\mathbf{C}$, which is dependent on the element's nodal coordinates, is given in detail in the [Appendix A](#). The elastic matrix $\mathbf{D}_m$ of the laminated composite shell is determined by integrating it through the thickness:

$$\mathbf{D}_m = \sum_{j=1}^n \int_{z_{j-1}}^{z_j} \bar{\mathbf{Q}}_m dz \quad (8)$$

2.5 Membrane-Bending Interactions in Asymmetric Laminated Composites

For non-symmetric layer arrangements across the mid-plane (e.g., $[0^\circ/90^\circ]$ laminates, $[\theta/-\theta/\theta/-\theta]$), membrane and bending behavior interact through matrices $\mathbf{D}_{mb}$. This interaction causes pure membrane loading to induce bending deformation, and *vice versa*. The interacting stiffness matrices are defined by:

$$\mathbf{K}_{mb} = \int_{\Delta} \mathbf{B}_m^T \mathbf{D}_{mb} \bar{\mathbf{B}}_b d\Delta \quad (9a)$$

$$\mathbf{K}_{bm} = \int_{\Delta} \bar{\mathbf{B}}_b^T \mathbf{D}_{mb} \mathbf{B}_m d\Delta \quad (9b)$$

$$\mathbf{D}_{mb} = \sum_{j=1}^n \int_{z_{j-1}}^{z_j} \bar{\mathbf{Q}}_m z dz \quad (9c)$$

The membrane-bending interaction is eliminated, i.e., $\mathbf{K}_{mb} = \mathbf{0}$, when the laminate is symmetrical across the mid-plane (symmetric laminate).

2.6 Coordinate System Transformation and Global Matrix Assembly

Since each triangular flat element has its own local coordinate system, the element stiffness matrix $\mathbf{K}_e$ (18×18 in the local coordinate system) must be converted to the global coordinate system before assembly. The global-local axis transformation matrix $\mathbf{T}_e$ (18×18) is constructed from the direction cosines of the local coordinate system to the global coordinate system. The element stiffness matrix converted to the global coordinate system has the form:

$$\mathbf{K}_e^g = \mathbf{T}_e^T \mathbf{K}_e \mathbf{T}_e \quad (10)$$

The global stiffness matrix $\mathbf{K}$ and nodal load vector $\mathbf{f}$ are assembled from the contributions of all elements, and the system equilibrium equations ([Eq. \(11\)](#)) are solved to find the global nodal displacement vector $\mathbf{u}$:

$$\mathbf{K}\mathbf{u} = \mathbf{f} \quad (11)$$

where $\mathbf{f}$ and $\mathbf{u}$ are the global nodal load vector and the global nodal displacement vector, respectively. After solving [Eq. \(11\)](#) with the imposed boundary conditions, the stress field in each element and each composite layer is recovered from the nodal displacement field through the strain-displacement relationship and the stress-strain relationship of each material layer.

3 Spline-Based Shape Parameterization and Automated Meshing

3.1 Overview of Geometric Parameterization Strategy

In the structural shell shape optimization problem, the choice of geometric parameterization method directly affects three decisive factors: (i) the dimension of the design variable space and therefore the search efficiency of the optimization algorithm; (ii) the continuity and smoothness of the generated shape, which affects the accuracy of FEM analysis and the engineering feasibility of the solution; and (iii) the ability to automate the process within the optimization loop. The approach of directly using FEM node coordinates as design variables, although simple in implementation, results in an extremely large variable space and often produces non-smooth or technically impractical shapes due to a lack of geometric continuity constraints. Conversely, spline-based parameterization with control points as design variables allows describing a rich class of shapes with a much smaller number of parameters, while ensuring continuity across the entire geometric domain—a necessary condition for accurate and stable solutions to the shell analysis problem.

In the proposed framework, the geometric parameterization strategy is built on the principle of complete separation between geometric modeling and analytical modeling, connected via an automatic mesh generation step. Specifically, the shell shape is described by spline curves/surfaces through control points—those coordinates are direct design variables of the optimization problem. For each set of control point values generated by the optimization algorithm, the Gmsh software [48] is used to automatically generate the corresponding triangular mesh; this mesh is then fed into the CS-DSG3-A finite element analyzer to calculate the objective function (strain energy). The entire process—from updating control points, creating geometry, generating mesh to structural analysis—is fully automated in a Python pipeline, ensuring no manual intervention in the optimization loop.

3.2 Introduction to Gmsh

Gmsh [48] is open-source finite element mesh generation software with a full Python API, allowing the construction and analysis of geometry, mesh generation, and mesh file export entirely from Python without a graphical interface. Gmsh provides fast, lightweight, and user-friendly mesh generation tools with flexible parameter input and visualization capabilities. Gmsh is well-supported for creating geometric lines and surfaces through Splines/B-Spline/NURBS. In particular, Gmsh allows for complex geometry processing, mesh quality control, and automatic numbering of nodes and elements.

Gmsh acts as a bridge between spline geometry modeling and FEM analysis modeling: with each new set of control points proposed by the BCMO optimization algorithm, the Gmsh API automatically (i) constructs the new geometry, (ii) generates a suitable triangular mesh, and (iii) returns node-element data to the CS-DSG3-A solver. This entire process takes place in memory without the need for intermediate file writes/reads, reducing I/O overhead in the optimization loop. The benefits of integrating Gmsh into the optimization process lie in two main points. First, mesh quality is automatically guaranteed—Gmsh allows for element angle optimization, avoiding the distorted elements that manual mesh generation easily creates when the curvature geometry is complex. Gmsh supports options to automatically fine-tune elements in areas of high curvature, which is particularly useful when the shell shape changes significantly between optimization generations. Secondly, through the Physical Group (PG), boundary conditions and loads can be precisely applied to nodes even if the mesh structure changes after each iteration. In Gmsh, PGs are a set of geometric entities (points, curves, and surfaces) assigned fixed integer labels. When Gmsh recreates the mesh after each update of the design variables, the PG labels remain the same for the geometric entities, not for the nodes or elements. These labels are used to determine the degrees of freedom at the nodes and elements belonging to the PG. Therefore, regardless of any complete change in the mesh structure—including changes in the

number of nodes and elements—the applied boundary conditions and loads are still accurately mapped to their intended geometric positions.

3.3 Automated Mesh Generation Process via Gmsh API

3.3.1 Parameterization of Shell Geometry Using Splines

The mid-surface of the shell is described on the (x, y, z) coordinate system by points, edges, and surfaces. The points (called control points) serve as the basis for constructing the spline curves and surfaces. The geometric structure of the shell can be changed by altering the coordinates of the control points. Geometric boundary conditions, including support positions, tangent angles at the boundary, or symmetry constraints, are imposed by fixing or linearly constraining the coordinates of the control points, ensuring that the shell shape always satisfies the geometric boundary conditions throughout the optimization process. In the proposed framework, only the z -coordinate (controlling the shell's curvature) of the control points is allowed to vary, while the x and y coordinates of all control points are kept fixed to respect the boundary conditions and avoid geometric degradation. The set of z -coordinate of N_{cp} control point is described in the form:

$$\mathbf{d} = \{z_1, z_2, \dots, z_{N_{cp}}\}^T \in \mathbb{R}^{N_{cp}} \quad (12)$$

Reducing the dimensionality from the entire FEM node coordinates (typically hundreds to thousands of nodes) to a few control point coordinates (3 to 10 in the numerical examples of this paper) is a key advantage of the spline-based shell geometry parameterization method, allowing the metaheuristic algorithm to search efficiently in low-dimensional design spaces. Design constraints are directly imposed on z_i as box constraints: $z_{i,\min} \leq z_i \leq z_{i,\max}$, where the upper/lower limits are determined based on technical requirements (e.g., maximum allowable height, installation space constraints). Additionally, volume constraints or slenderness constraints can be added, depending on the specific problem requirements.

3.3.2 Creating Geometric Shapes and Generating Meshes

For each set of $\mathbf{d} = \{z_1, z_2, \dots, z_{N_{cp}}\}^T$, the geometry construction and mesh generation process in the Gmsh API is performed according to the steps presented below. Internally, Gmsh uses the OpenCASCADE kernel (OCC) to handle accurate NURBS/B-spline geometry, ensuring that curve continuity is maintained in the geometric representation.

- Step 0: Initialize the Gmsh API: The Gmsh API is initialized using the command `gmsh.initialize()`.
- Step 1: Create control points and build boundaries: Control points (a collection of movable and fixed points) are created in Gmsh using the command: `gmsh.model.occ.addPoint()`. Spline boundaries are created in Gmsh using a command with input being a list of point tags and spline degree:
 - + `gmsh.model.occ.addBSpline()`: Creates a B-spline curve based on control points. This curve typically only passes through the start and end points, while the points in between act as “magnets” attracting the curve towards them (the curve only passes near them, not cuts through them).
 - + `gmsh.model.occ.addSpline()`: Creates a C^2 -continuous (smooth) curve that must pass through exactly all control points.
- Step 2: Create the shell surface: The shell surface is constructed directly from a set of boundaries and internal control points depending on the geometry of the problem. The Gmsh API commands for this purpose are as follows:
 - + `gmsh.model.occ.addBSplineSurface()`: creates a surface from a grid of control points arranged in a 2D matrix (rows and columns). The generated surface typically only passes through the points

- at the edges, while the points in the middle act as “attractions” and shape the surface’s convexity and concavity.
- + `gmsh.model.occ.addBSplineFilling()`: fills a tight-fitting B-Spline surface onto given closed boundaries.
- + `gmsh.model.occ.addSurfaceFilling()`: uses an energy minimization algorithm to create a surface that correctly traverses the boundaries and can also bend to precisely pass the control points located in the middle of the surface. This is the most “powerful and versatile” function and will be used in the illustrative examples in [Section 6](#).
- Step 3: Assign geometric boundary conditions: Boundaries and boundary points corresponding to support, symmetry, or load conditions are marked by physical groups in the Gmsh API using the command `gmsh.model.addPhysicalGroup()`. These physical groups are used in post-processing to automatically and consistently assign mechanical boundary conditions and loads to the CS-DSG3-A solver, regardless of the number of nodes or mesh structure changing between iterations.
- Step 4: Generate triangular mesh: The command `gmsh.model.mesh.generate(2)` activates 2D surface mesh generation. The default Delaunay mesh generation algorithm in Gmsh produces a high-quality triangular mesh with a reasonable edge ratio, suitable for the CS-DSG3-A element.
- Step 5: Extract mesh data: After generating the mesh, node coordinates and element connections are extracted directly from Gmsh’s memory via `gmsh.model.mesh.getNodes()` and `gmsh.model.mesh.getElements()`. This data is converted into NumPy array data structures and passed directly to the CS-DSG3-A solver without intermediate files, minimizing I/O overhead.

3.3.3 Mesh Quality Control

During the optimization loop, the quality of the triangular mesh can vary significantly as the shell shape changes, especially as design variables approach the boundaries of the allowable domain, creating shapes with large curvatures or extreme span/height ratios. To ensure stable mesh quality throughout the optimization process, two measures are applied:

- Firstly, the element size is controlled by the parameter lc (characteristic length)—determined based on the overall size of the structure and accuracy requirements—via `gmsh.model.mesh.setSize()`. This ensures that while the mesh is automatically regenerated to conform to the evolving spline geometry, the overall mesh density and local element sizes remain highly uniform and scale proportionally with the geometric bounds.
- Secondly, the Laplacian smoothing algorithm is activated after mesh generation using the command `gmsh.model.mesh.optimize('Laplace2D')`, which improves the edge ratio of triangular elements and reduces the number of distorted elements in areas with high curvature.

In this study, the value of lc was chosen based on a preliminary investigation, ensuring both accuracy and computational cost. For example, $lc = 1.0$ m is chosen for Example 1. The mesh convergence results for Example 1 are included in [Section 6.2.1](#). For the other examples, the mesh convergence study is not shown to save the paper space.

3.4 Integration into the Overall Python Framework

The entire process from geometric parameterization to structural analysis is implemented in a single *fitness* function that takes the z-coordinate of control points as input and returns a strain energy value. This function is called by the optimization algorithm and performs five sequential steps: (1) reconstructs the full control points; (2) checks the geometrical feasibility; (3) builds the geometry and generates the mesh

via the Gmsh API; (4) assigns mechanical boundary conditions and loads; (5) calls the solver CS-DSG3-A and calculates the strain energy value. This clear layered architecture ensures that any changes to the geometric parameterization method, mesh generation engine, or FEM analyzer can be made independently without affecting the remaining components, a crucial characteristic for the scalability and maintainability of the framework.

In terms of computational cost of fitness, the Gmsh mesh generation time typically accounts for about 5%–15% of the total time for isotropic materials, and less than 2% for laminated materials, while the CS-DSG3-A analysis accounts for the remainder. Therefore, the acceleration strategy is primarily focused on reducing the number of *fitness* calls through the k-NNC filtering mechanism, rather than individually optimizing each substep in the pipeline. This strategy and the k-NNC integration mechanism with BCMO are detailed in [Section 4](#) below.

4 BCMO Algorithm and k-NNC Filter

4.1 Balancing Composite Motion Optimization (BCMO)

In general, consider the optimization problem with D design variables, $\mathbf{x} = \{x_1, x_2, \dots, x_D\}^T$, described by [Eq. \(13\)](#),

$$\begin{aligned} &\text{Find } \mathbf{x} \text{ to minimize } f(\mathbf{x}) \\ &\text{subjected to } g_m(\mathbf{x}) \leq 0, m = 1..M \end{aligned} \quad (13)$$

where $f(\mathbf{x})$ is the objective function (total strain energy of the shell in this study); $g_m(\mathbf{x})$ is the m -th constraint.

4.1.1 Population Initialization

The initial population consists of NP individuals corresponding to NP design options, all randomly initialized in the design space:

$$\mathbf{x}_i = \mathbf{LB} + \text{rand}(1, D) \times (\mathbf{UB} - \mathbf{LB}) \quad (14)$$

where $\mathbf{LB}$ and $\mathbf{UB}$ are the lower and upper bounds of the design space, respectively; $\text{rand}(1, D)$ is a D -dimensional vector with evenly distributed components in $[0, 1]$. After initialization, individuals are evaluated and ranked in ascending order according to the objective function value $f(\mathbf{x})$.

4.1.2 Determining the Instantaneous Global Optimal Point

BCMO does not require prior knowledge of the global optimal point O . Instead, the algorithm defines an instantaneous global optimal point, O_{in} , which is updated with each generation. At generation t , a test individual $\mathbf{x}_{\text{trial}}^t$ is created based on the information of the previous generation:

$$\mathbf{x}_{\text{trial}}^t = \mathbf{x}_c + \mathbf{v}_{k_1/k_2}^t + \mathbf{v}_{k_2/1}^t \quad (15)$$

where $\mathbf{x}_c = (\mathbf{LB} + \mathbf{UB})/2$ is the center of the design space; k_1 is randomly chosen in $[2, NP]$, k_2 is also randomly chosen ensuring $k_2 < k_1$; $\mathbf{v}_{k_1/k_2}^t$ and $\mathbf{v}_{k_2/1}^t$ are respectively the relative movements defined with $L_{\text{GS}} = 1$ (see [Section 4.1.3](#)). The instantaneous global optimal point is determined by the selection mechanism:

$$\mathbf{x}_{O_{\text{in}}}^t = \mathbf{x}_{\text{trial}}^t \text{ if } f(\mathbf{x}_{\text{trial}}^t) < f(\mathbf{x}_1^{t-1}), \text{ otherwise } \mathbf{x}_{O_{\text{in}}}^t = \mathbf{x}_1^{t-1} \quad (16)$$

The best individual of generation t is assigned directly: $\mathbf{x}_1^t = \mathbf{x}_{O_{in}}^t$. This design ensures that O_{in} is continuously updated without the need for an auxiliary variable to store the history.

4.1.3 The Mechanism of Individual Composite Motion

The updated position of individual i ($i > 1$) in generation t is determined by the composite motion principle:

$$\mathbf{x}_i^t = \mathbf{x}_i^{t-1} + \mathbf{v}_{i/j} + \mathbf{v}_j \quad (17)$$

where the $\mathbf{v}_{i/j}$ is local search component (the relative movement of individual i with respect to the randomly selected individual j with better rank, $j < i$), and the $\mathbf{v}_j$ is global search component (the movement of individual j towards O_{in}).

Global search—Movement $\mathbf{v}_j$ is calculated:

$$\mathbf{v}_j = \alpha_j \cdot (\mathbf{x}_{O_{in}}^t - \mathbf{x}_j^{t-1}) \quad (18)$$

with $\alpha_j = L_{GS} \times \mathbf{d}\mathbf{v}_j$. The global step coefficient L_{GS} and the direction vector $\mathbf{d}\mathbf{v}_j$ are determined through the random value $TV_j \sim U(0, 1)$:

$$L_{GS} = \exp(-(1/D) \cdot (j/NP) \cdot r_j^2) \text{ if } TV_j > 0.5 \quad (19a)$$

$$L_{GS} = \exp(-(1/D) \cdot (1 - j/NP) \cdot r_j^2) \text{ if } TV_j \leq 0.5 \quad (19b)$$

$$\mathbf{d}\mathbf{v}_j = \text{rand}(1, D) \text{ if } TV_j > 0.5; \mathbf{d}\mathbf{v}_j = -\text{rand}(1, D) \text{ if } TV_j \leq 0.5 \quad (20)$$

where $r_j = \|\mathbf{x}_j^{t-1} - \mathbf{x}_{O_{in}}^t\|$ is the Euclidean distance from individual j to O_{in} . When j is close to O_{in} (r_j is small), $L_{GS} \rightarrow 1$ increases the attractive force; when j is far from O_{in} , L_{GS} decreases sharply to avoid premature convergence.

Local search—Relative motion $\mathbf{v}_{i/j}$ is computed by:

$$\mathbf{v}_{i/j} = \alpha_{ij} \cdot (\mathbf{x}_j^{t-1} - \mathbf{x}_i^{t-1}) \quad (21)$$

with $\alpha_{ij} = L_{LS} \times \mathbf{d}\mathbf{v}_{ij}$, where $L_{LS} = 1$ to balance local exploration and exploitation capabilities; $\mathbf{d}\mathbf{v}_{ij}$ determined similarly to Eq. (20).

Probability balance—With the rule of equal probability of 0.5 for the signs of $\mathbf{d}\mathbf{v}_j$ and $\mathbf{d}\mathbf{v}_{ij}$, each individual has 4 combinations of movement with equal probability:

$$P(v_{ik}) = P(v_{i/j}) \times P(v_j) = 0.5 \times 0.5 = 0.25, k = 1, 2, 3, 4 \quad (22)$$

This ensures a strict balance between global/local search and exploitation/exploration for each individual throughout the optimization process.

4.1.4 Constraint Handling

The optimization problem of laminated composite shell structures in this study may involve inequality constraints. Linear additive values are used to determine constraint violations:

$$c(\mathbf{x}) = \sum_m^M \max(0, g_m(\mathbf{x})) \quad (23)$$

The comparison between two individuals is performed according to Deb's rule [51] as follows:

- (i) If both individuals are feasible ($c \leq 0$) or have the same level of violation: the individual with the smaller f wins;
- (ii) If only one individual is feasible: the feasible individual wins;
- (iii) If neither is feasible: the individual with the smaller violation wins.

4.2 K-Nearest Neighbor Comparison (k-NNC) Mechanism

4.2.1 Motivation and Overview

In the structural optimization problem of a laminated composite shell, each evaluation of the objective function requires solving the finite element problem on a mesh automatically reconstructed via Gmsh, resulting in significant computational cost. To reduce the number of evaluations actually required, the k-nearest neighbor comparison (k-NNC) mechanism is integrated into the BCMO loop as a lightweight surrogate filter. The core idea: before calling the expensive FEM solver, the k-NNC predicts whether the new individual $\mathbf{x}_i^t$ is actually better than the current individual $\mathbf{x}_i^{t-1}$, based on information already available in the archive of evaluated points.

4.2.2 Data Archive

k-NNC maintains two datasets: the current population $P = \{(\mathbf{x}_i, f(\mathbf{x}_i))\}$ and an additional archive $P_w = \{(\mathbf{x}_i^w, f(\mathbf{x}_i^w))\}$ containing individuals that have been replaced through the optimization process. The complete reference dataset is:

$$X_{\text{ref}} = P \cup P_w, |P_w| \leq NP \quad (24)$$

The archive size of P_w is limited to NP to control memory and keep a low computational cost for k-NNC operation. When the archive is full, a random instance is removed to make room for a new one.

4.2.3 Finding k Neighbors and Estimating Their Weights

Given a test individual $\mathbf{x}_i^t$ and a reference set X_{ref} (with the i -th individual removed to avoid self-comparison), the algorithm determines the k nearest neighbors based on Euclidean distance:

$$\{id_1, \dots, id_k\} = \arg \min_k \|\mathbf{x}_i^t - \mathbf{x}_{\text{ref},j}\|_2, j \in \{1, \dots, |X_{\text{ref}}|\} \quad (25)$$

The objective function value and the constraint violation level of $\mathbf{x}_i^t$ are estimated by inverse distance weighting (IDW) interpolation:

$$\hat{f}(\mathbf{x}_i^t) = \frac{\sum_{l=1}^k f(\mathbf{x}_{id_l})/D_l}{\sum_{l=1}^k 1/D_l} \quad (26)$$

$$\hat{c}(\mathbf{x}_i^t) = \frac{\sum_{l=1}^k c(\mathbf{x}_{id_l})/D_l}{\sum_{l=1}^k 1/D_l} \quad (27)$$

where $D_l = \|\mathbf{x}_i^t - \mathbf{x}_{id_l}\| + \varepsilon_0$ ($\varepsilon_0 = 10^{-16}$ to avoid dividing by 0); $c(\mathbf{x}_{id_l})$ is the level of violation of neighbor l , determined according to Eq. (23).

4.2.4 Weighted Evaluation Index and Decision Rules

Each neighbor $\mathbf{x}_{id_l}$ is compared to the current individual $\mathbf{x}_i^{t-1}$ according to the feasibility rule (Section 4.1.4). The comparison result is a binary variable:

$$e_l = \text{ebetter}(f(\mathbf{x}_{id_l}), c(\mathbf{x}_{id_l}), f(\mathbf{x}_i^{t-1}), c(\mathbf{x}_i^{t-1})) \in \{0, 1\} \quad (28)$$

The function *ebetter* gives a value of 1 if $\mathbf{x}_{id_l}$ is better than $\mathbf{x}_i^{t-1}$, otherwise it gives a value of 0. The weighted composite rating index is calculated as follows:

$$\text{eval} = \frac{\sum_{l: e_l=1}^k e_l / D_l}{\sum_{l=1}^k 1 / D_l} - \frac{\sum_{l: e_l=0}^k 1 / D_l}{\sum_{l=1}^k 1 / D_l} \quad (29)$$

Essentially, *eval* measures the weight difference between the number of “better $\mathbf{x}_i^{t-1}$ ” and “worse $\mathbf{x}_i^{t-1}$ ” neighbors, where each neighbor is weighted inversely proportional to its distance from the neighbor $\mathbf{x}_i^t$. Hence, closer neighbors have a greater influence.

Decision rule:

Evaluate $f(\mathbf{x}_i^t), c(\mathbf{x}_i^t) \iff \text{eval} \geq 0$

Skip $\iff \text{eval} < 0$ (30)

Additionally, an override rule is applied to avoid missing good solutions: if the estimation shows $\mathbf{x}_i^t$ feasibility (i.e., $\hat{c}(\mathbf{x}_i^t) \leq 0$) and it is better than the current individual (i.e., $\hat{f}(\mathbf{x}_i^t) < f(\mathbf{x}_i^{t-1})$), it is forced to the actual evaluation regardless of the sign of *eval*:

$$\text{If } \hat{c}(\mathbf{x}_i^t) \leq 0 \wedge \hat{f}(\mathbf{x}_i^t) < f(\mathbf{x}_i^{t-1}) \wedge \text{eval} < 0, \text{ then } \text{eval} = 1 \quad (31)$$

The effectiveness of this strategy depends on the quality of the archive X_{ref} : the richer the archive (more evaluated points, well-distributed in the design space), the higher the accuracy of the k-NNC filter. In the initial optimization phase when the archive is sparse, k-NNC tends to return $\text{eval} \geq 0$ more frequently, ensuring complete exploration. As the archive becomes denser, the skip rate increases, which reduces the computational burden but, as detailed in Section 6.3, may lead to a trade-off in the solution quality for certain complex design spaces. The only parameter of k-NNC is k —the number of neighbors. In this study, $k = 3$ is used, which is the minimum choice sufficient to distinguish the improvement trend while keeping the balance between the accuracy of k-NNC and the skip rate (detailed investigation can be found in Refs. [46,47])

4.3 Integrating k-NNC into BCMO

In the proposed framework, k-NNC does not completely replace the FEM solver but acts as an intelligent filter before each costly evaluation. At each i -th instance update, the decision is made along two ways:

- True evaluation ($\text{eval} \geq 0$): The CS-DSG3-A solver is called to calculate the exact objective function. The result is used to update the individual and add the old individual to archive P_w .
- Skip ($\text{eval} < 0$): The FEM solver is not called, saving all computational costs for that individual. The current individual is retained for the next generation.

The integration of BCMO and the k-NNC filter relies on a structural synergy where BCMO’s distance-driven exploration aligns with the k-NNC’s inverse distance weighting, ensuring consistent screening across the design space. Furthermore, BCMO’s balanced search strategies generate a high-quality, evenly distributed

archive, enhancing the k-NNC's interpolation accuracy and providing resilience against misclassification at high skip rates.

Overall, the combination of BCMO and k-NNC forms a computationally efficient optimization framework, highly suitable for the structural shape optimization problem of laminated composite shells where each evaluation requires automated mesh construction and the CS-DSG3-A finite element solution.

5 Overall Workflow of the Proposed Framework

5.1 Formulation of the Shape Optimization Problem

The shell shape optimization problem is stated in terms of the strain energy minimization problem, which is equivalent to the problem of maximizing global stiffness under a given external load:

$$\text{Minimize: } U(\mathbf{x}) = \frac{1}{2} \mathbf{u}^T \mathbf{K}(\mathbf{x}) \mathbf{u} = \frac{1}{2} \mathbf{f}^T \mathbf{u}$$

$$\text{Subject to: } \mathbf{K}(\mathbf{x}) \mathbf{u} = \mathbf{f}$$

$$V(\mathbf{x}) \leq V_0$$

$$x_{i,min} \leq x_i \leq x_{i,max}, i = 1, \dots, D$$

where $\mathbf{x} = \{x_1, x_2, \dots, x_D\}^T$ is vector of design variables; $U(\mathbf{x})$ is total strain energy; $\mathbf{f}$ is the nodal load vector; $\mathbf{u}$ is the nodal displacement vector; $\mathbf{K}(\mathbf{x})$ is the global stiffness matrix dependent on shell shape through the design variable vector $\mathbf{x}$; $V(\mathbf{x})$ is the material volume corresponding to $\mathbf{x}$, and V_0 is the allowable volume limit. Matrix $\mathbf{K}(\mathbf{x})$ is assembled from the CS-DSG3-A element stiffness matrix presented in [Section 2](#).

5.2 Framework Overview

The proposed framework integrates three technical components presented in the previous sections: (1) the CS-DSG3-A shell element within the framework of first-order shear deformation theory (FSDT) for composite shells ([Section 2](#)); (2) the spline geometric parameterization technique combined with automatic mesh generation via the Gmsh API ([Section 3](#)); and (3) the BCMO optimization algorithm incorporating the k-NNC mechanism ([Section 4](#)). This combination forms a closed-loop process that allows for the optimization of the geometric shape of the composite shell without requiring manual intervention between stages. The core of the framework is the explicit mapping from design variable vector $\mathbf{x}$ to shell geometry via B-Spline control point coordinates $\mathbf{d}$, from which the Gmsh API automatically generates geometric objects and finite element mesh. For each design generated by BCMO, the CS-DSG3-A solver performs static analysis and returns the objective function value $U(\mathbf{x})$ along with $V(\mathbf{x})$. The k-NNC mechanism is activated to screen design candidates before calling the solver, thereby significantly reducing the number of actual objective function evaluations (NFEs).

To maximize computational resource preservation, the execution sequence within the optimization loop is structured such that geometric parameterization and automated mesh generation via the Gmsh API are strictly executed after the k-NNC filtering decision. When a candidate configuration is predicted as unpromising by k-NNC, the framework immediately bypasses the entire structural evaluation pipeline. Consequently, the computational system avoids not only the expensive CS-DSG3-A finite element solution but also the associated geometric boundary checks and meshing costs. Unpromising individuals incur only a trivial numerical cost related to neighbor distance calculations within the reference archive X_{ref} , ensuring that the percentage reduction in total computational time closely aligns with the reduction rate of NFEs.

5.3 Detailed Procedure

The workflow is organized into two phases: the preprocessing phase (Steps 1–2, performed once before the optimization loop) and the iterative optimization phase (Steps 3–6 in the BCMO loop). Specifically, the process consists of six steps as follows:

- Step 1: Declare geometric parameters. The user defines the geometric topology of the shell, including a set of points, curves, and surfaces. Control points are classified as fixed points, defining geometric boundaries that remain invariant during optimization, and design points whose coordinates are either design variables or linearly dependent on design variables via a connector mapping matrix. Markers (physical groups) are assigned to points, edges, and surfaces to associate with boundary conditions and loads in Step 2.
- Step 2: Declare the FEM model data. Material parameters (elastic modulus, Poisson's ratio, density for each layer), composite layer configuration (orientation angle, thickness of each layer), boundary conditions (degrees of freedom connections at point and edge markers), and loads (forces on the surface, point, or edge) are declared independently of the geometry. This data structure remains unchanged throughout the optimization process; only the FEM mesh is updated in each iteration.
- Step 3: Generate new design. BCMO generates a set of individuals (design variable vectors) for the current generation.
- Step 4: Mesh generation, FEM analysis and objective function calculation. For each individual $\mathbf{x}$, the design variable values are mapped to the coordinates of the control point $\mathbf{d}$ via a connector function, then the Gmsh API is called to: (i) build the B-Spline geometry object according to the topology declared in Step 1; (ii) generate an unstructured triangular mesh with element size controlled by the scaling factor lc ; (iii) extract node coordinates, element connectivity, and map marker-DOF to provide to the solver. The volume of the shell is calculated (in cases where the problem considers volume constraints). The mesh is fed into the CS-DSG3-A solver along with the FEM model data declared in Step 2 to solve the static problem. From the obtained displacement and stress fields, the objective function (strain energy) and constraint vectors (stress, displacement, etc.) are calculated. The results are returned to BCMO as scalar values or vectors depending on the problem.
Before calling the solver at this step, the k-NNC mechanism is activated to predict the individual's potential for improvement based on a database of previous evaluations; individuals predicted to have no prospects are eliminated without actual evaluation, thereby reducing overall computational cost.
- Step 5: Update the individual. BCMO evaluates the objective function value and feasibility of the new individual and updates it into the population to replace the old individual. The old individual, if it is worse than the new one, is put into the archive P_w .
- Step 6: Check the stopping condition and return the results. The loop terminates when the maximum number of generations N_g is reached or when the improvement in the objective function between successive generations is less than the convergence threshold ϵ . The returned results include: the optimal design variable vector $\mathbf{x}^*$, the corresponding objective function value $f(\mathbf{x}^*)$, the convergence history, and the number of actual function evaluations (NFEs). The geometry and internal force fields of the optimal design are output for post-processing.

The flowchart of the optimization framework is shown in [Fig. 3](#).

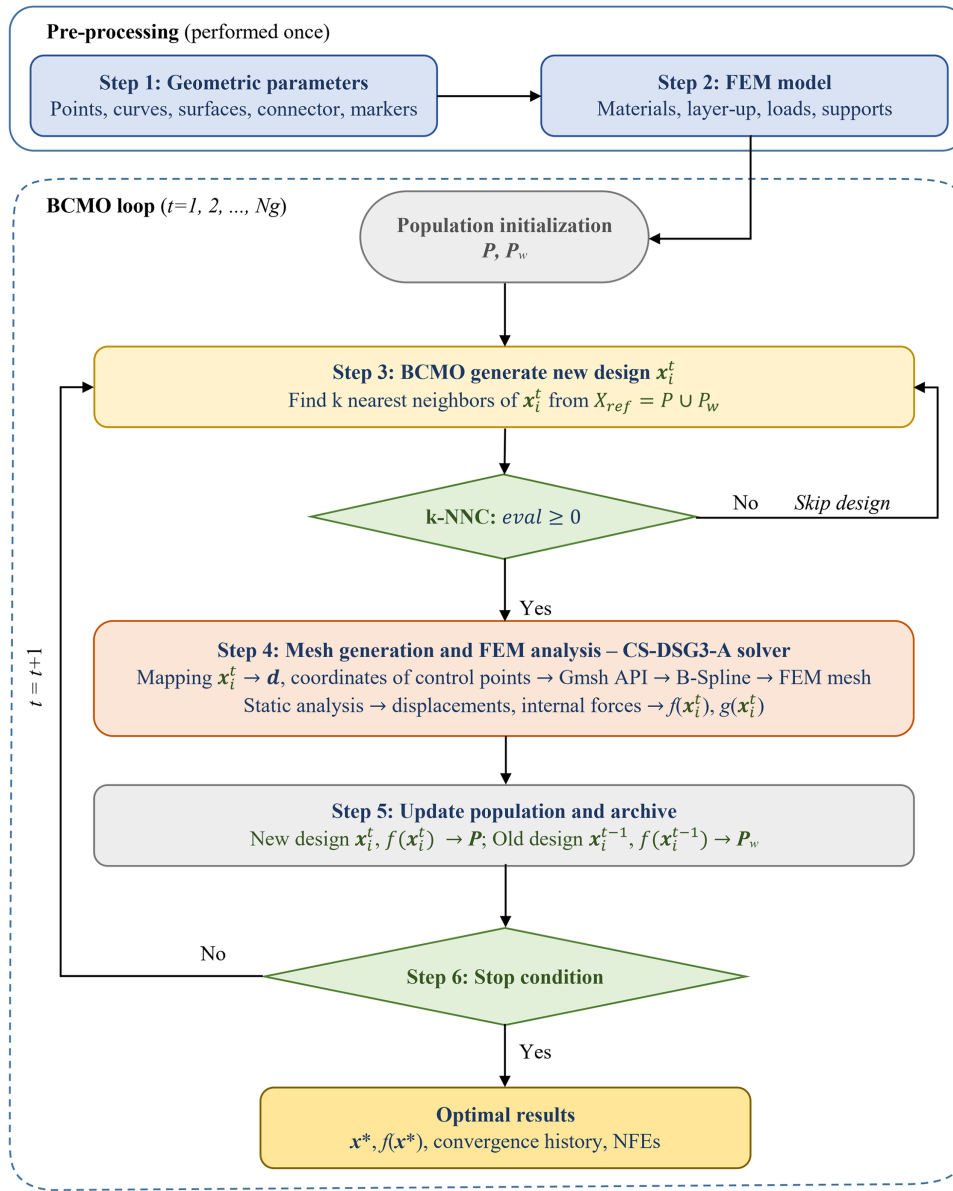


Figure 3: Flowchart of the BCMO/k-NNC+Gmsh+CS-DSG3-A integrated optimization process.

5.4 Interface between Modules

The modularity of the framework allows for the replacement of individual components without affecting the rest. The interface between BCMO and the solver is standardized through the $fitness(\mathbf{x})$ function: it takes a vector of design variables as input and returns the objective function and constraint values. The interface between the Gmsh API and the CS-DSG3-A solver is implemented through a standard mesh data structure consisting of a node coordinate matrix, a nodal degrees of freedom matrix, an element connectivity matrix (by degrees of freedom), and a marker mapping dictionary-DOF/marker-element ID.

In the current implementation, the entire framework is implemented using Python. BCMO and k-NNC are implemented in the *BCMO* module; geometry and mesh generation are handled by the *shell.geometry*

module via the Gmsh API; FEM analysis is performed by the *shell.system* module with the CS-DSG3-A solver; and results are visualized via the *shell.postprocessing* module.

6 Illustrative Examples and Discussion

This section presents several numerical examples to verify the reliability and performance of the proposed composite shell shape optimization framework. The examples include different geometries (positive and negative Gaussian curvature), boundary conditions, and load systems (uniformly distributed load, concentrated load, complex load system), aiming for a comprehensive assessment of the framework's applicability.

All examples were modeled using Python 3.12.7 and Gmsh API 4.15.1. The unstructured triangular meshing relies on the standard Frontal-Delaunay algorithm with Laplacian smoothing activated. Hardware for the execution includes: (1) Laptop Thinkpad T14s with CPU 11th Gen Intel Core i5-1135G7 2.40 GHz and 16 GB RAM; (2) Computer with CPU Intel Core i7-14700K 5.60 GHz and 64 GB RAM. Moreover, a well-known FEM solver, CalculiX using second-order 6-node triangular shell element (S6), is adopted to verify the accuracy of the implemented CS-DSG3-A solver.

To comprehensively evaluate the benefits of the k-NNC classifier, optimization problems were conducted using both the original BCMO algorithm and the BCMO algorithm incorporating k-NNC. For each test case and algorithm, 20 independent runs were performed to gather sufficient statistical data for evaluation.

6.1 CS-DSG3-A Element Verification

To verify the performance of the proposed CS-DSG3-A element, two classic structural benchmarks are evaluated prior to the optimization studies. Elementary tests such as membrane and bending patch tests are omitted here, as their compliance and mesh-distortion tolerance have been mathematically guaranteed in the original formulations [4,7,49].

6.1.1 Isotropic Thin-Shell Benchmark: The Scordelis-Lo Roof

The Scordelis-Lo roof [52], a standard benchmark for evaluating thin-shell kinematics under gravity load, is analyzed to assess shear locking resistance. The problem simulates a thin cylindrical vault, clamped at two ends. The geometric and material parameters are set as: length $L = 50$, radius $R = 25$, $\varphi = 40^\circ$, thickness $h = 0.25$ ($h/R = 1/100$), elastic modulus $E = 4.32 \times 10^8$, Poisson's ratio 0.0, and material density 36.7347. The shell is discretized using two distinct topologies: a regular triangular mesh ($N \times N$) and an irregular mesh generated by Gmsh with equivalent element size ($l_c \cong 0.5L/N$). To isolate the direct kinematical enhancement yielded by integrating the Allman-type drilling DOF, the structural response is solved using two alternative formulations: the original CS-DSG3 without drilling DOF (the element stiffness components corresponding to the drilling DOFs are set as 10^{-3} times the maximum diagonal value in the element stiffness matrix [4]), and the formulated CS-DSG3-A. The vertical displacement at the midpoint of the free edge is the evaluation target, with reference value 0.3024 [52]. The results corresponding to different mesh densities are shown in Table 1.

As shown in Table 1, the numerical responses expose two prominent characteristics of the formulated CS-DSG3-A element. First, the element provides a remarkably accelerated convergence trajectory; on a coarse mesh of $N = 8$, CS-DSG3-A delivers a deflection of 0.2991, nearly matching the theoretical target of 0.3024. The original CS-DSG3 element exhibits a slower convergence path, necessitating a refined $N = 24$ resolution to approximate nearly the same accuracy. Second, the CS-DSG3-A formulation demonstrates

insensitivity toward unstructured mesh irregularity. While the element CS-DSG3 experiences noticeable solution shifts when moving to irregular coordinates, the proposed CS-DSG3-A remains completely stable. This explicitly confirms that the drilling DOF integration achieves its primary motivation of safeguarding the numerical loop against geometric mesh degradation during automated spline boundary transformations.

Table 1: Central vertical displacement under regular and irregular mesh configurations ($h/R = 1/100$).

Mesh Type	Element	N = 8	N = 12	N = 16	N = 20	N = 24	N = 40
Regular	CS-DSG3	0.2662	0.2840	0.2916	0.2953	0.2975	0.3008
	CS-DSG3-A	0.2991	0.3000	0.3005	0.3008	0.3010	0.3021
Irregular		$lc = 3$	$lc = 2$	$lc = 1.5$	$lc = 1.2$	$lc = 1$	$lc = 0.6$
	CS-DSG3	0.2828	0.2909	0.2956	0.2978	0.2991	0.3013
	CS-DSG3-A	0.2992	0.2999	0.3005	0.3008	0.3010	0.3023

6.1.2 Unsymmetric Laminated Shell Benchmark: Spherical Shell Subjected to Concentrated Load

To validate the implementation of the membrane-bending coupling described in Section 2.5, a spherical shell adopted from Ref. [53] with an unsymmetric $[0^\circ/90^\circ]$ cross-ply laminate stacking sequence subjected to a central point load is analyzed. Because of the unsymmetric layup, the structural responses of displacement and bending moments are highly coupled. The shell has a square plan projection ($a = b$), curvature radius R , and thickness $h = a/10$. Single layers are assumed to be orthotropic with properties: $E_1/E_2 = 25$, $G_{12} = G_{13} = 0.5E_2$, $G_{23} = 0.2E_2$, $\nu_{12} = \nu_{21} = 0.25$. The dimensionless center deflection, defined as $w = 10^2 wh^3 E_2 / Pa^2$, is monitored across varying curvature parameters ranging from deep shells ($R/a = 1$) to shallow shells ($R/a = 10$). Two mesh types (generated by Gmsh) are employed: (i) a Delaunay uniform mesh with element edge size $a/20$ (Fig. 4a); (ii) a Delaunay mesh refined toward the center with element edge size $a/20$ on the boundary down to $a/80$ at the midpoint (Fig. 4b).

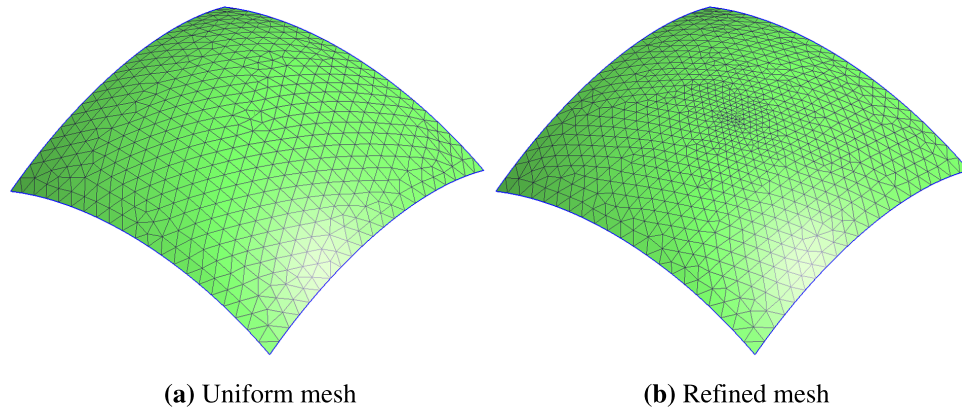


Figure 4: Finite element meshes for the cross-ply laminated spherical shell: (a) Delaunay uniform mesh; (b) Delaunay refined mesh toward load point.

The comparative results presented in Table 2 provide strong evidence of the present solver's capability. For all evaluated shell curvatures, the refined mesh utilizing the proposed CS-DSG3-A element closely approximates the high-order analytical solutions from Ref. [53]. The maximum relative error between the refined mesh results and the exact solution stays strictly below 1.32% across the entire curvature range. Since the unsymmetric cross-ply stacking sequence generates pronounced coupling effects between membrane

and bending behavior, these highly precise metrics verify that the integration of the laminate constitutive laws and the coupling matrix $\mathbf{D}_{mb}$ is executed. This confirms that the CS-DSG3-A element is fully dependable for laminated composite shell shape optimization.

Table 2: Dimensionless center deflection ($w = 10^2 wh^3 E_2 / Pa^2$) of cross-ply spherical shell under concentrated load with $a/h = 10$.

Mesh Type	Method	R/a = 1	R/a = 2	R/a = 3	R/a = 4	R/a = 5	R/a = 10
Uniform	CalculiX	3.8362	5.5170	6.3371	6.7089	6.9002	7.1855
	CS-DSG3-A	3.4371	5.0955	5.8944	6.2566	6.4469	6.7171
Refined	CalculiX	4.3614	6.0922	6.8431	7.1890	7.3870	7.7163
	CS-DSG3-A	4.0696	5.7710	6.5060	6.8451	7.0267	7.3510
	Exact [53]	4.0164	5.7620	6.5444	6.9098	7.1015	7.3836
	Error (%)	1.32	0.16	-0.59	-0.94	-1.05	-0.44

Additionally, simulation results obtained using the software CalculiX, employing 6-node quadratic triangular element (S6) on the same mesh, is also compared in Table 2. For the uniform mesh, the deflection results from CalculiX are consistently higher than those of the CS-DSG3-A element across the entire range of curvature radii (R/a). With the refined mesh, CalculiX (S6) tended to overestimate deflection, yielding errors larger than the exact solution (e.g., reaching 8.6% at (R/a = 1) and fluctuating between 4% and 6% at other curvature radii). This demonstrates that the proposed CS-DSG3-A element offers not only high reliability but also accuracy compared to the CalculiX quadratic S6 element in the analysis of laminated composite shells subjected to concentrated loads.

6.2 Shape Optimization of Isotropic Material Shells

6.2.1 Example 1: Free-Form Shell with Positive Gaussian Curvature

The first example investigates the optimization problem of a free-form shell with positive Gaussian curvature from Ref. [54] and a square plan view with sides of 24 m, and a initial convex height at the surface center of 3 m. The shell has a uniform thickness $h = 0.1$ m, isotropic material with elastic modulus $E = 30$ GPa and Poisson's ratio $\nu = 0.3$. The four edges of the shell are completely hinged, and the entire surface is subjected to a uniformly distributed vertical load with intensity $q = 8$ kN/m². The optimization model is built on a geometric grid of 9 control points. The only optimization variable is the z -coordinate of control point number 7 located at the center of the surface, taken in the range $[0, 6]$ m. The objective function is to minimize the total strain energy of the structure.

A mesh convergence study for the initial shape has been conducted by varying the element size parameter lc from 3 m down to 0.5 m under Gmsh control. The results (given in Table 3) demonstrate that as the mesh density increases, both the maximum deflection and strain energy exhibit excellent convergence, with the response stabilizing significantly at $lc = 1$ m (1.4119 mm and 2112.8235 J) and showing small variation when further refined to $lc = 0.5$ m. This implies that with $lc = 1$ m, the objective function is not significantly affected by the mesh density, while it ensures a robust balance between high accuracy and computational efficiency.

The BCMO algorithm was used with a population size of 20 and 50 iterations (corresponding to 1020 structural analyses and objective function evaluations). The statistical results from 20 independent runs are summarized in Table 4. The structural output from the best design is compared with those of initial design

in Table 5. Fig. 5a depicts the initial shape of the shell (with nine control points) and the element mesh created by the Gmsh API (with $lc = 1$ m), and Fig. 5b is the shape of the shell after optimization, which is similar to the optimal shape given in Ref. [54].

Table 3: Convergence of CS-DSG3-A solution with different element sizes for initial shape.

Size Parameter	$lc = 3$ m	$lc = 2$ m	$lc = 1$ m	$lc = 0.5$ m
No. nodes	102	206	769	2941
No. elements	170	362	1440	5688
Max. deflection (mm)	1.2839	1.3940	1.4119	1.4166
Strain energy (J)	2078.0832	2089.2836	2112.8235	2120.4653
CPU Time (s)	0.270432	0.6052	2.3497	10.8406

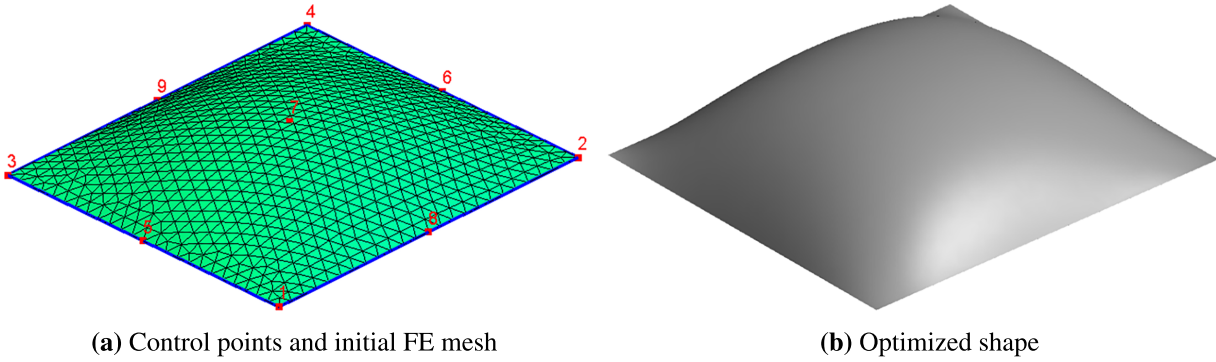


Figure 5: Initial and optimized shape of positive Gaussian curvature shell.

Table 4: Statistical results of optimal strain energy for Example 1.

Method	BCMO	BCMO/k-NNC	Difference (%)
Best (J)	658.1661	658.1661	
Mean (J)	658.5507	660.1260	
Worst (J)	662.0118	670.0737	
Standard deviation (J)	1.1537	3.6463	
Avg. NFEs	1020	306	-70.00
Avg. CPU time (s)	2756.29	955.59	-65.33

Table 5: Comparison of the best optimization design with the initial design for Example 1.

Parameter	Initial	Optimized	Reduction (%)
Strain energy (J)	2112.8235	658.1661	68.85
Max. deflection (mm)	1.4119	0.8000	43.34
Max. bending moment (kNm/m)	1.0393	0.2466	76.27

In the pre-optimization state, the CS-DSG3-A model yielded a strain energy of 2112.82 J. This value approximated the 2116.50 J result obtained from the isogeometric analysis (IGA) method in the Ref. [54], confirming the accuracy of the proposed shell element model in static analysis. After applying the proposed optimization framework, the strain energy decreased from 2112.82 to 658.1661 J, equivalent to a 68.85% reduction; the maximum deflection decreased from 1.4119 to 0.8000 mm (a 43.34% reduction); and the maximum bending moment decreased from 1.0393 kNm to only 0.2466 kNm (a 76.27% reduction). The obtained results are competitive to that of the IGA-PSO workflow reported in Ref. [54] (the optimal strain energy is 765.34 J). The significant reduction in strain energy and bending moment demonstrates that the BCMO algorithm effectively utilizes the design space, largely transforming the load state from local bending to a compressive membrane stress mechanism. This is further illustrated in Fig. 6, which depicts the distribution of bending moment M_{yy} in the shell before and after optimization.

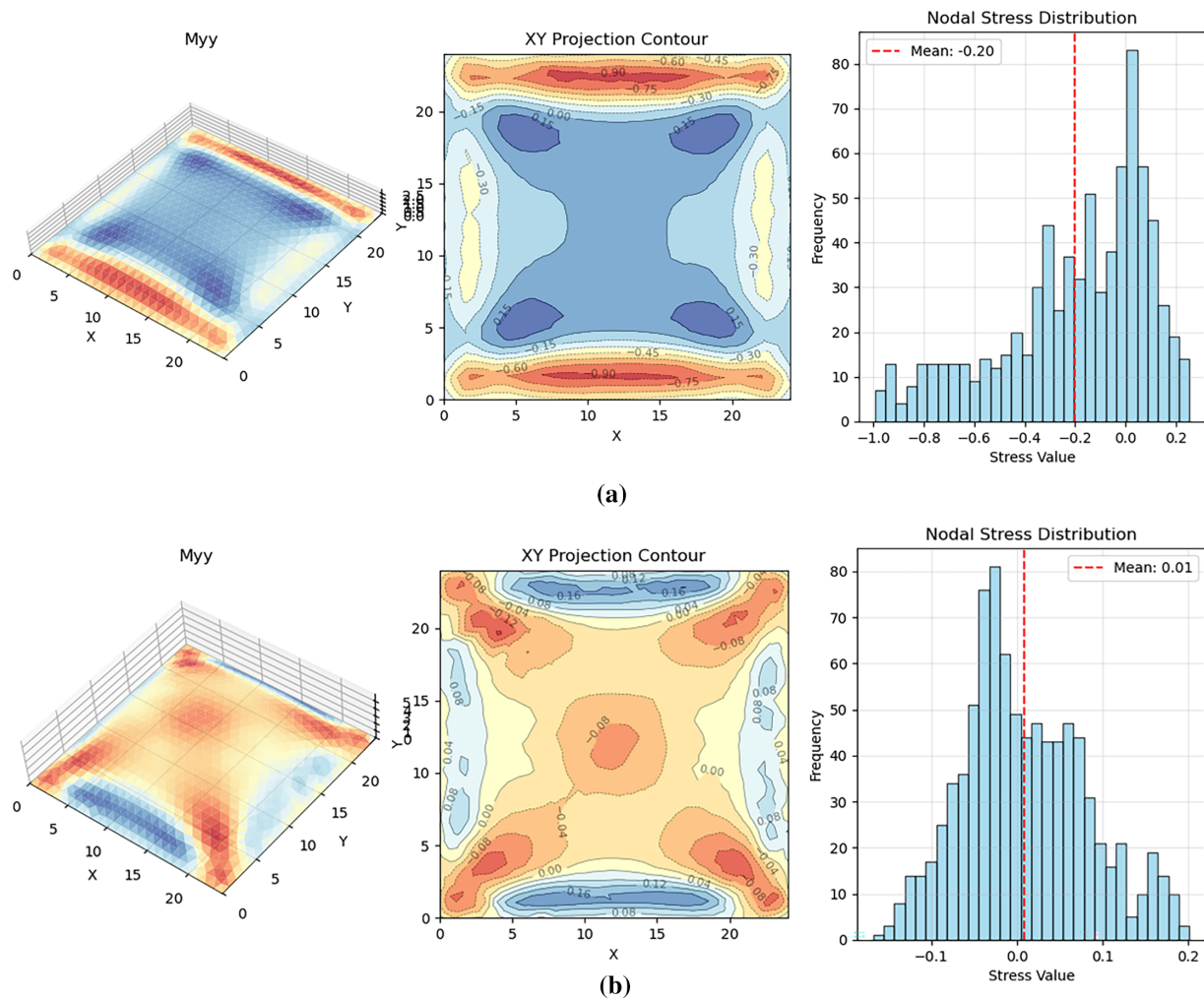


Figure 6: Moment M_{yy} in a shell with positive Gaussian curvature: (a) Before optimization; (b) After optimization.

Statistically, the outcome of BCMO/k-NNC is slightly worse than those of BCMO in terms of the mean, the worst, and the standard deviation of optimal strain energy, as shown in Table 4. Nevertheless, when using the k-NNC filter, the number of structural analyses and objective function evaluations (NFEs) is reduced to average of 306 evaluations, representing a reduction of over 65% in computational workload compared

to the case without using the k-NNC filter. The average convergence-history plots (strain energy vs. NFE) comparing BCMO and BCMO/k-NNC are depicted in Fig. 7, showing that BCMO reaches a higher strain energy value on the same budget as BCMO/k-NNC.

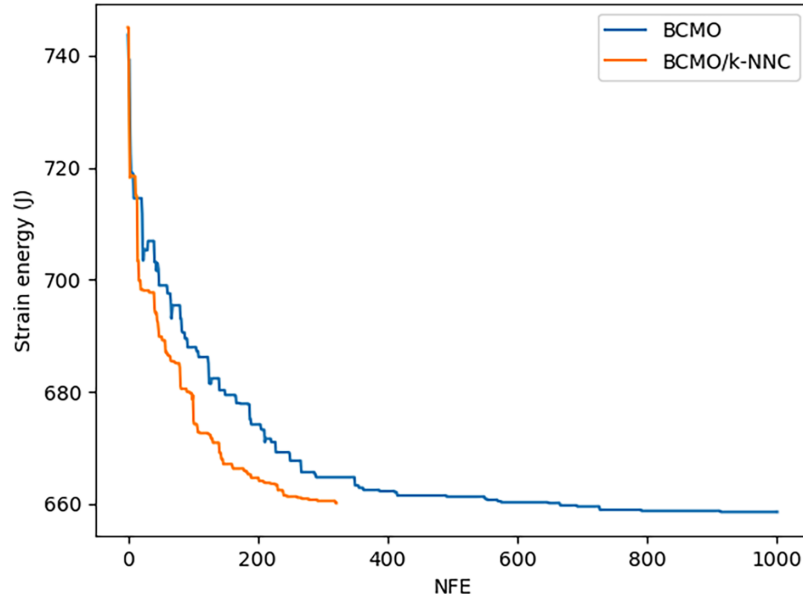


Figure 7: Example 1: averaged convergence-history of the strain energy due to BCMO and BCMO/k-NNC.

To further strengthen the confidence in the accuracy of the proposed CS-DSG3-A shell element, we have provided verification using CalculiX (with second-order 6-node triangular shell element S6) on the optimized shape. The comparison is shown in Table 6. With the same mesh density ($lc = 1$ m), the results demonstrate excellent agreement in both maximum deflection (0.8000 vs. 0.8093 mm) and strain energy (658.1661 vs. 659.5642 J). Notably, the proposed CS-DSG3-A shell element achieves this high accuracy with significantly fewer resources, requiring only 889 nodes and 1680 elements compared to the 3457 nodes and 1680 elements needed by CalculiX. The refined mesh with $lc = 0.5$ m (3385 nodes) results in structural responses that match CalculiX very well. These findings successfully validate the reliability, precision, and high computational efficiency of the proposed formulation against standard FEM software.

Table 6: Comparison of CS-DSG3-A solutions with those of CalculiX for optimal shape of Example 1.

Model	CS-DSG3-A($lc = 1$ m)	CS-DSG3-A($lc = 0.5$ m)	CalculiX (S6)($lc = 1$ m)
No. nodes	889	3385	3457
No. elements	1680	6576	1680
Max. deflection (mm)	0.8000	0.8068	0.8093
Strain energy (J)	658.1661	659.2085	659.5642

6.2.2 Example 2: Free-Form Shell with Negative Gaussian Curvature

The second example considers the optimization problem of a free-form shell with negative Gaussian curvature and an asymmetrical initial geometry, still referenced from document [54]. The shell has a rectangular plan view with dimensions $12 \text{ m} \times 8 \text{ m}$, thickness 0.1 m , isotropic material with elastic modulus $E = 30 \text{ GPa}$ and Poisson's ratio $\nu = 0.2$. Boundary conditions: two long sides are hinged; two short sides are

free. The downward vertical load is uniformly distributed with intensity $q = 10 \text{ kN/m}^2$. The design space is parameterized by a grid of 15 control points (see Fig. 8a); the optimization variables are the z coordinates of the five control points in the central region (5, 6, 7, 10, 11), varying in the range $[0, 10] \text{ m}$. The initial geometry and the FE mesh created by Gmsh are shown in Fig. 8a, with the chosen element size parameter $lc = 0.5 \text{ m}$.

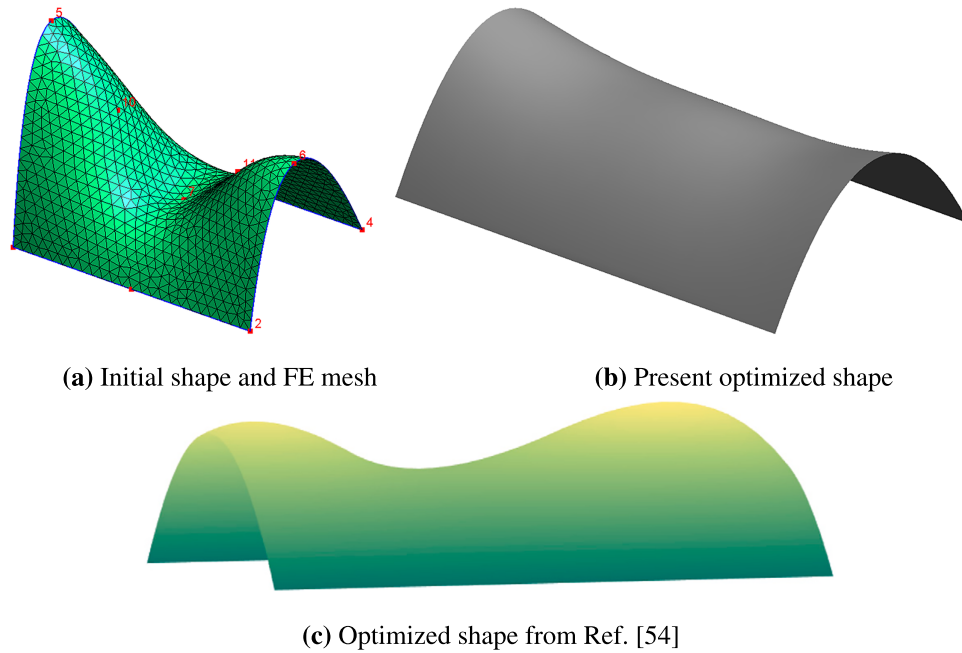


Figure 8: Initial and optimized shapes of negative Gaussian curvature shell [54].

Before optimization, the CS-DSG3-A model yielded a strain energy of 146.2196 J, closely matching the 147.29 J value of the IGA method in Ref. [54]. In the best optimization, the strain energy was reduced to 40.4150 J (a 72.36% reduction). Compared to the reference result (47.16 J), the proposed framework achieved a larger energy reduction. In particular, the maximum bending moment was controlled at 0.3485 kNm (an 89.93% reduction) and the maximum deflection was only 0.1436 mm (an 86.19% reduction). The results are summarized in Table 7. A CalculiX S6 model established for the optimal shell shape with the same mesh density gives the maximum deflection of 0.1457 mm and strain energy of 40.2628 J, which agree well with the proposed model.

Table 7: Comparison of the best optimization design with the initial design for Example 2.

Parameter	Initial	Optimized	Reduction (%)
Strain energy (J)	146.2196	40.4150	72.36
Max. deflection (mm)	1.0398	0.1436	86.19
Max. bending moment (kNm/m)	3.4623	0.3485	89.93

Notably, the optimal shape obtained in this study (Fig. 8b) has geometric differences compared to the asymmetrical wavy shell in the original paper [54] (Fig. 8c). The distinct geometrical profile captured in Fig. 8b indicates that the proposed workflow converged toward an alternative, highly efficient local optimal region compared to the reference solution in Ref. [54]. It is worth noting that in the original study, the authors explicitly reported that the optimization path for this specific structure exhibited noticeable instability, with

results varying widely from one run to the next even under a small set of five design variables. This fluctuation strongly implies that traditional metaheuristics like PSO face significant challenges with severe local extrema entrapment within this specific design landscape. In contrast, the current optimization framework arrives at a highly stable, smooth, and symmetrical optimal shape. While this divergence is partially influenced by differences in boundary interpolation and mesh generation pipelines, the stability and lower energy achieved here suggest that the integrated k-NNC filtering and BCMO provide a robust drive to navigate away from the highly unstable local valleys encountered by standard algorithms.

In this example, the computational efficiency of k-NNC is also impressive. With a population size of 20 and a number of iterations of 100, BCMO/k-NNC uses 864 structural analyses and function evaluations in average, which is only 42.47% of the 2020 function evaluations of the original BCMO. Consequently, the total computational time is reduced by 52.06% (see Table 8). Although at the final stage, BCMO, statistically, derives slightly better outcome (Table 8), BCMO/k-NNC exhibits faster convergence at the same NFEs, as demonstrated by the convergence-history plotted in Fig. 9. Specifically, the best optimized strain energy of BCMO/k-NNC is even better than that of BCMO (40.4150 vs. 40.4189 J).

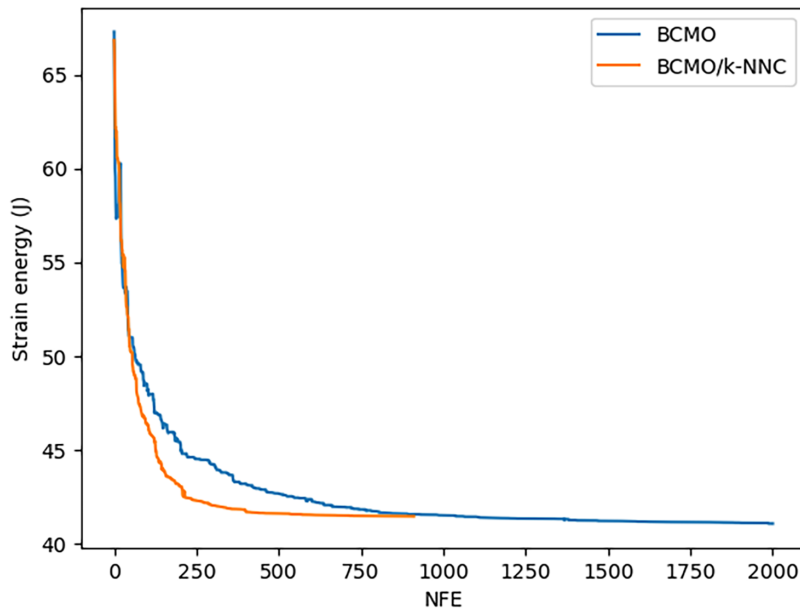


Figure 9: Example 2: averaged convergence-history of the strain energy due to BCMO and BCMO/k-NNC.

Table 8: Statistical results of optimal strain energy for Example 2.

Method	BCMO	BCMO/k-NNC	Difference (%)
Best (J)	40.4189	40.4150	
Mean (J)	41.0662	41.4518	
Worst (J)	43.1080	43.8484	
Standard deviation (J)	0.9398	1.2763	
Avg. NFEs	2020	864	-57.23
Avg. CPU time (s)	3820.23	1831.45	-52.06

6.2.3 Example 3: Simply Supported Shell Subjected to Concentrated Load with Volume Constraint

The third example considers a shell with four simple support corners, plan dimensions $5 \text{ m} \times 5 \text{ m}$, thickness $h = 25 \text{ cm}$ (refer to document [55]). The material is isotropic with $E = 200 \text{ GPa}$ and $\nu = 0.3$. A vertically concentrated load $P = 40 \text{ kN}$ is placed at the center of the shell. In this example, a 5×5 control point grid is used to design the geometric shape of the shell (see Fig. 10a), with the z coordinates of the points as the design variables. The optimization process is performed under strict volume constraints: $V \leq 1.001 \times V_0$, with $V_0 = 6.70 \text{ m}^3$.

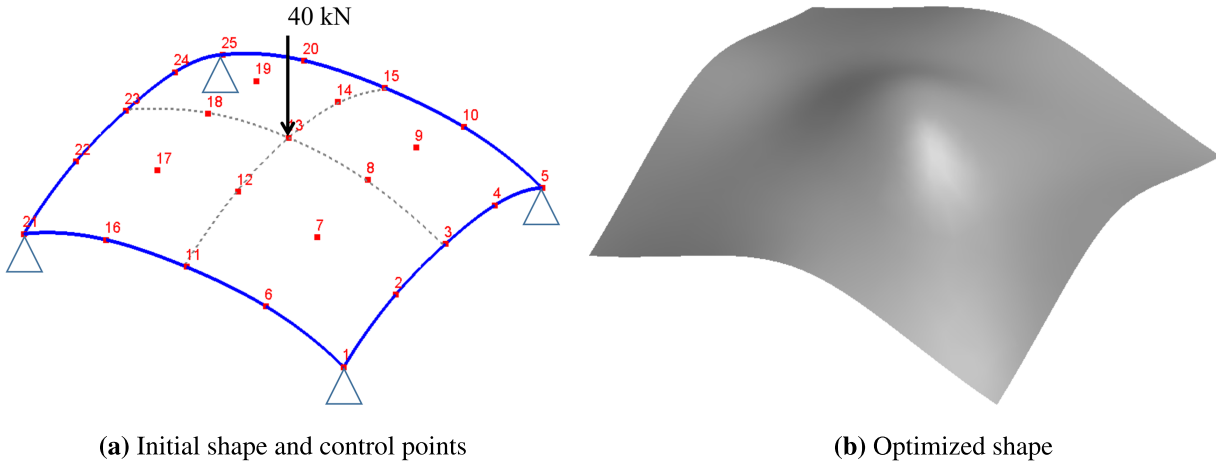


Figure 10: Initial and optimized shapes of simply supported shell subjected to a concentrated load.

Prior to optimization, the proposed model with element size parameter $lc = 0.5 \text{ m}$ recorded: strain energy 0.1909 J (structural compliance 0.3818 Nm), maximum bending moment 4.3463 kNm , and maximum deflection 0.0095 mm . With a population size of 50 and 100 iterations, the optimal results by BCMO/k-NNC are summarized in Table 9. After optimization, the shell volume was controlled at 6.7067 m^3 , fully satisfying the volume constraint. The total strain energy decreased by 40.96% to 0.1127 J , resulting structural compliance decreased to 0.2254 Nm . Compared to the optimal results based on IGA analysis with a 13×13 control point grid in reference [55] (structural compliance after optimization was 0.2141 Nm), the results of the proposed model are 5.28% higher. In particular, the maximum bending moment decreased from 4.3463 to 1.6955 kNm (a reduction of 70.00%), and the maximum deflection decreased to 0.0056 mm (a reduction of 41.05%). The optimal shape tends to raise the central region relative to the edges (Fig. 10b). This geometric transformation is consistently verified with the solution obtained from the IGA model in the reference [55]. Raising the central region helps the structure suppress local bending, efficiently converting energy to a membrane stress mechanism, enhancing the load-bearing capacity at the point of concentrated force. It is noted that for this shell subjected to point support and concentrated load, CalculiX (same mesh with S6 element) produces higher values for the optimal results (strain energy of 0.1516 J ; maximum deflection of 0.0072 mm).

Table 9: Comparison of the best optimization design with the initial design for Example 3.

Parameter	Initial	Optimized	Reduction (%)
Volume (m^3)	6.6940	6.7043 ($< 1.001 \times V_0$)	–
Strain energy (J)	0.1909	0.1127	40.96
Max. bending moment (kNm/m)	4.3463	1.6955	60.99
Max. deflection (mm)	0.0095	0.0056	41.05

Statistical results (Table 10) show high stability of the proposed framework in this example. In terms of computational load, k-NNC allows for the elimination of more than 3100 function evaluations (FEM analyses) in average, corresponding to a computational load reduction of 56.23%. The convergence-history of strain energy are depicted in Fig. 11, clearly demonstrating the efficiency of BCMO/k-NNC, compared with BCMO. Moreover, BCMO/k-NNC even with much lower computational cost, exhibits superiority to BCMO in all statistical results (the best, the mean, the worst, and the standard deviation) of the optimal strain energy as shown in Table 10.

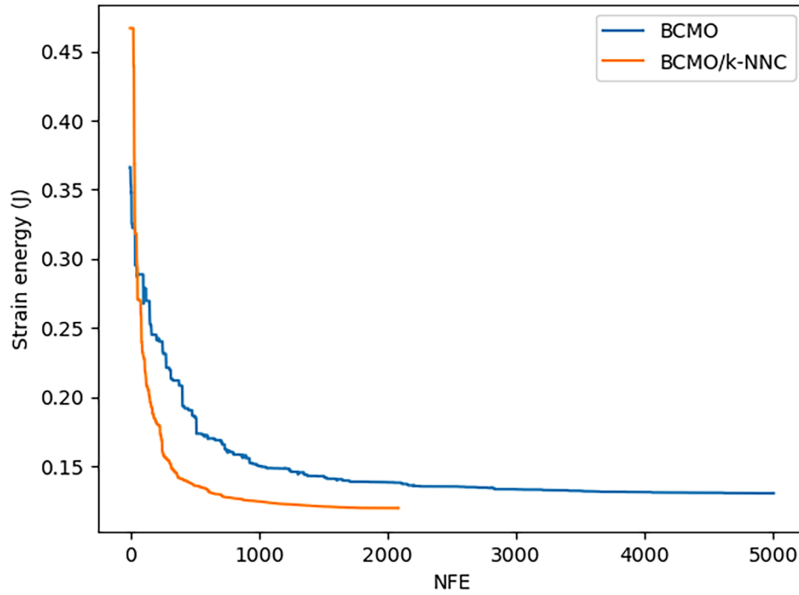


Figure 11: Example 3: averaged convergence-history of the strain energy due to BCMO and BCMO/k-NNC.

Table 10: Statistical results of optimal strain energy for Example 3.

Method	BCMO	BCMO/k-NNC	Difference (%)
Best (J)	0.1131	0.1127	
Mean (J)	0.1259	0.1198	
Worst (J)	0.1433	0.1315	
Standard deviation (J)	0.0096	0.0058	
Avg. NFEs	5050	1882.7	-62.72
Avg. CPU time (s)	2544.2	1113.5	-56.23

6.2.4 Example 4: Shell under a Complex Loading System

This example extends the scope of verification to complex boundary conditions and multi-point load systems. The shell has the same structural configuration as in Example 3, however the boundary conditions and loads are different (Fig. 12a).

With a controlled volume $V < 1.001 \times V_0$, the proposed framework yields an optimized strain energy of 0.4963 J (structural compliance of 0.9927 Nm) (a 59.00% reduction); the maximum bending moment is maintained at 3.3409 kNm (a 36.17% reduction), and the maximum deflection is controlled at 0.0195 mm (a 62.28% reduction). The results are summarized in Table 11. The structural outputs of CS-DSG3-A agree with

those obtained by CalculiX (strain energy of 0.5498 J, maximum deflection of 0.0206 mm). The optimal shape (Fig. 12b) shows that the central area of the shell surface is concave while the four surrounding corners are raised. This geometric transformation redistributes the stress field from local bending to a widely distributed membrane stress, while increasing the geometric moment of inertia of the system, effectively exploiting the load-bearing capacity under complex force combinations.

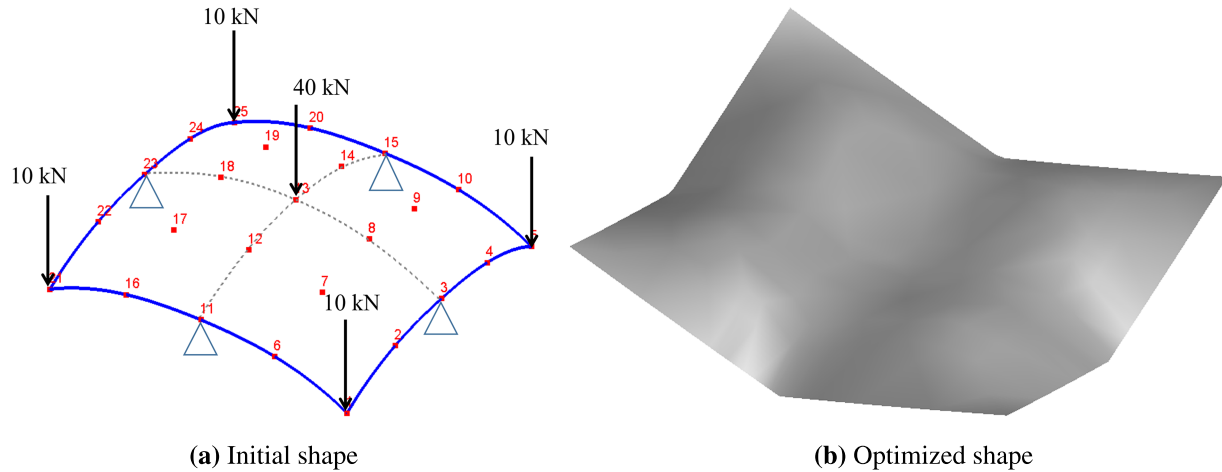


Figure 12: Initial and optimized shapes of simply supported shell subjected to multi-point concentrated load.

Table 11: Comparison of best optimization results for Example 4.

Parameter	Initial	Optimized	Reduction (%)
Volume (m ³)	6.6940	6.7065 ($<1.001 \times V_0$)	–
Strain energy (J)	1.2106	0.4963	59.00
Max. bending moment (kNm/m)	5.2337	3.3409	36.17
Max. deflection (mm)	0.0517	0.0195	62.28

The statistical and visual results (Table 12 and Fig. 13) demonstrate a clear efficiency of the proposed method. BCMO/k-NNC (with a population size of 50 and 100 iterations) exhibits a substantial computational advantage, reducing averaged NFEs by 63.35% and CPU time by 57.85%. This acceleration is verified by Fig. 13, where the BCMO/k-NNC curve shows a much faster convergence. In terms of solution quality, BCMO/k-NNC identifies a superior optimum with a lower “Best” value of 0.4963 J (vs. 0.4977 J of BCMO). However, standard BCMO offers slightly better robustness, maintaining a lower standard deviation (0.0012 vs. 0.0044) over its longer run up to 5000 function evaluations.

Based on the four examples above, some observations can be drawn about the performance of the proposed framework:

- The CS-DSG3-A solution closely matches those of other solver (CalculiX with S6 element) in the investigated examples, confirming that the improved shell element achieves high accuracy and reliability in free-form shell static analysis.
- The optimal configuration obtained reflects the transformation from local bending to membrane stress. This accurately reflects the nature of the shell shape optimization problem: effectively exploiting geometric stiffness to distribute the load in the most reasonable way.

- The results of Examples 1–4 show that the BCMO with k-NNC can produce considerable reduction of computational cost compared with BCMO, while it remains high accuracy.
- Examples 3 and 4 demonstrate the ability to handle multiple constraints simultaneously (volume, boundary conditions, multi-point load system). The framework maintains optimal performance in both cases, demonstrating its generality and scalability for real-world applications.

Table 12: Statistical results of optimal strain energy for Example 4.

Method	BCMO	BCMO/k-NNC	Difference (%)
Best (J)	0.4977	0.4963	
Mean (J)	0.4993	0.5000	
Worst (J)	0.5013	0.5170	
Standard deviation (J)	0.0012	0.0044	
Avg. NFEs	5050	1850.9	−63.35
Avg. CPU time (s)	2612.3	1101.2	−57.85

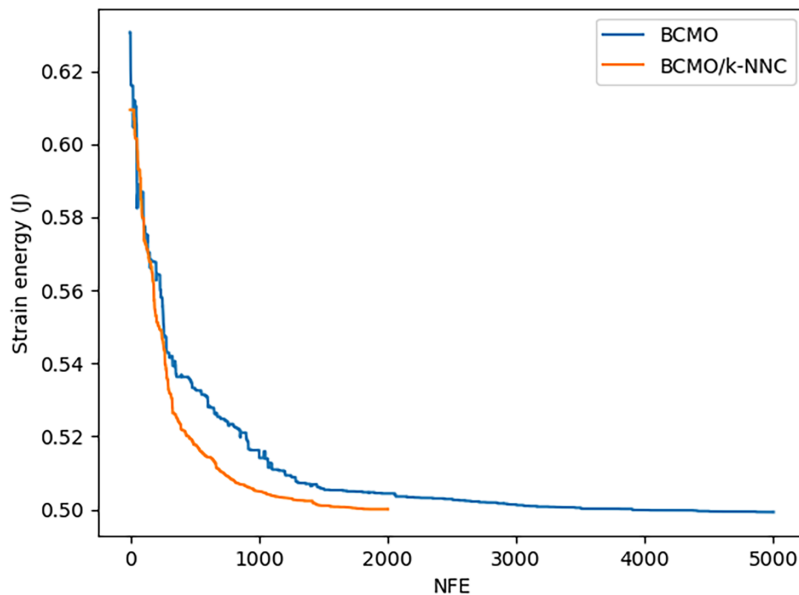


Figure 13: Example 4: averaged convergence-history of the strain energy due to BCMO and BCMO/k-NNC.

6.3 Laminated Composite Shells

Consider a shell made from laminated composite material. The shell model is subjected to immovable pin-supported conditions at its four corners ($u_x = 0$, $u_y = 0$, $u_z = 0$), has a square plan of 5×5 m and a control-point layout as shown in Fig. 14. The total shell thickness is 25 mm and is composed of four orthotropic layers with the following material characteristics: $E_1 = 155.0$ GPa, $E_2 = 10.3$ GPa, $G_{12} = 6.9$ GPa, $G_{13} = 6.9$ GPa, $G_{23} = 0.69$ GPa, and $\nu_{12} = 0.3$. The applied load is uniformly distributed on the shell surface with an intensity of 6.9 kN/m^2 . The proposed framework is used to find the optimal shape with different laminate configurations, including: $[0^\circ/90^\circ/90^\circ/0^\circ]$, $[0^\circ/90^\circ/0^\circ/90^\circ]$, $[45^\circ/-45^\circ/-45^\circ/45^\circ]$ and $[45^\circ/-45^\circ/45^\circ/-45^\circ]$. The fiber angles are oriented with respect to the global coordinate system. The z-coordinate of the control points is constrained in range from 0 to 1.2 m. For this example, BCMO uses 100 iterations and a population

size of 50. The statistical optimization results are summarized in Table 13. The averaged convergence curves of the optimal energy during the optimization process are plotted in Fig. 15. The optimization results were obtained based on a mesh density with $l_c = 0.5$ m, which requires a reasonable computational cost. A brief mesh-sensitivity analysis was conducted for the $[45^\circ/-45^\circ/45^\circ/-45^\circ]$ optimized configuration. The details, provided in Appendix B, confirm that the baseline mesh ($l_c = 0.5$ m) yields well-converged and stable solutions.

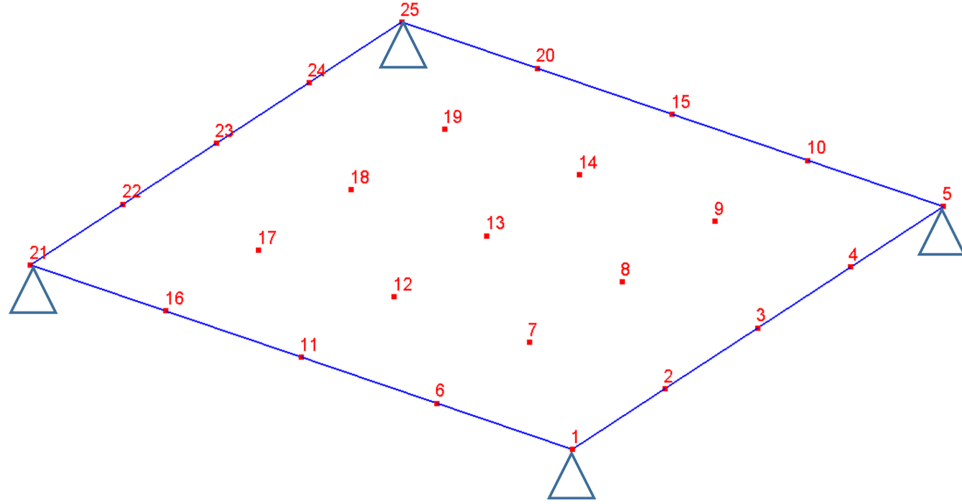


Figure 14: Initial shape and control point layout of the laminated shell subjected to uniformly distributed load.

Table 13: Statistical results of optimal strain energy (J) for laminated shell.

Configuration	Method	Best	Mean	Worst	Standard Deviation	NFEs
$[0^\circ/90^\circ/90^\circ/0^\circ]$	BCMO	40.6484	53.3869	65.8977	8.2190	5050
	BCMO/k-NNC	41.8010	52.4252	65.1798	7.5633	2166.6
$[0^\circ/90^\circ/0^\circ/90^\circ]$	BCMO	40.6811	47.2701	68.4863	6.4627	5050
	BCMO/k-NNC	41.2676	56.6691	69.1979	7.2612	2147.8
$[45^\circ/-45^\circ/-45^\circ/45^\circ]$	BCMO	29.4669	48.8888	70.3982	10.5806	5050
	BCMO/k-NNC	39.3222	50.2765	74.3453	10.2630	2147.5
$[45^\circ/-45^\circ/45^\circ/-45^\circ]$	BCMO	29.8979	46.4786	75.6225	12.6276	5050.0
	BCMO/k-NNC	38.2606	52.1843	73.2173	9.5809	2185.6

Table 13 clearly demonstrates the computational efficiency gained by integrating k-NNC into BCMO. It is shown that k-NNC reduces the number of objective function evaluations (NFEs) from 5050 to approximately 2150–2185 across all four configurations. Conversely, the quality of the best solution found by BCMO/k-NNC and BCMO exhibits distinct behaviors depending on the stacking sequence. The performance gap ranges from relatively small for cross-ply laminates (approx. 1.1%–1.4%) to more significant for angle-ply laminates (approx. 25%–33%; e.g., 39.3222 vs. 29.4669 J for the $[45^\circ/-45^\circ/45^\circ/-45^\circ]$ layup). This suggests that k-NNC may overlook certain promising individuals within the more complex design spaces of angle-ply laminates, where the fitness landscape exhibits more pronounced multi-modality compared to cross-ply configurations. The differences in the mean values are also appreciable for some configurations

($[0^\circ/90^\circ/0^\circ/90^\circ]$ and $[45^\circ/-45^\circ/45^\circ/-45^\circ]$ layups). Nevertheless, BCMO/k-NNC stands out as an efficient optimizer when computational budgets are strictly limited, saving up to 58% of finite element evaluations at the cost of a localized performance loss in highly non-linear layups.

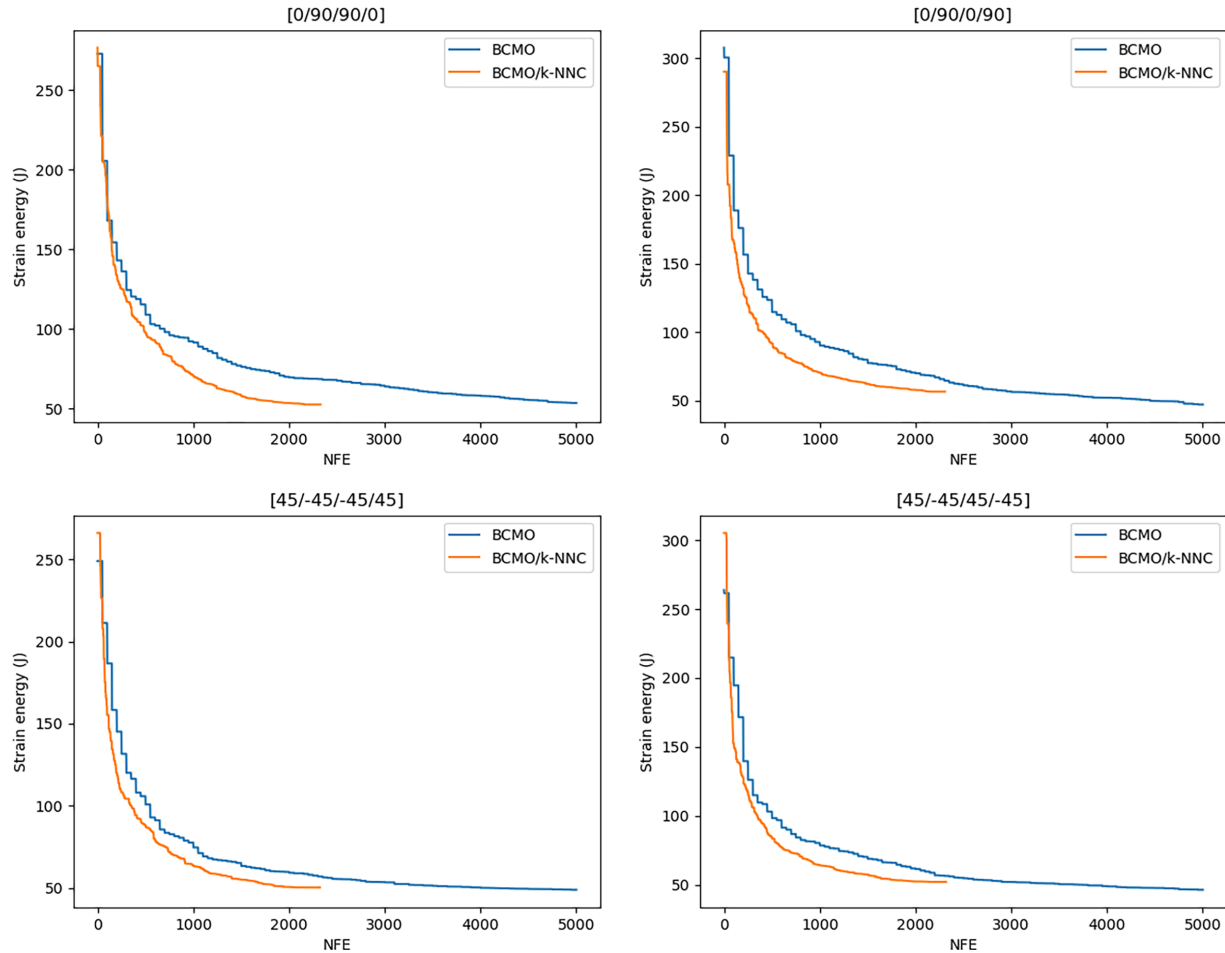


Figure 15: Laminated shell: averaged convergence history of the strain energy due to BCMO and BCMO/k-NNC.

The convergence histories (strain energy vs. NFE) in Fig. 15 further support the computational efficiency of BCMO/k-NNC. Across all four laminate configurations, the BCMO/k-NNC curve consistently lies below the BCMO curve throughout the optimization process, particularly in the early stage ($NFE < 1000$), indicating that the k-NNC filtering accelerates convergence toward promising regions of the design space, not merely truncates the search. Notably, the two averaged curves converge to a similar strain energy level by the point where BCMO/k-NNC terminates (~ 2100 – 2300 NFE). This confirms that the early termination enabled by k-NNC captures the majority of achievable energy reduction at a fraction of the computational cost. However, it is critical to distinguish between this averaged convergence behavior and the best solution obtained across independent runs as reported in Table 13, showing that BCMO manages to find markedly better best-case solutions for the angle-ply laminates when given the full computational budget.

The results in Table 14 illustrate the optimized shell forms corresponding to the best optimization result provided by BCMO/k-NNC. Starting from an initial flat plate subjected to pure bending, all four laminate configurations converged into an arch-like form curved in both the x-z and y-z planes, with rise values varying according to the stacking sequence. This represents the core mechanism of the problem: by allowing

curvature in both directions via spline control points, the structure transitions from a bending-dominated state to a membrane-dominated state. This mirrors the mechanism of arches or thin shells, which transfer lateral loads into axial forces along a curved surface, a far more efficient approach than that of a flat plate resisting loads through bending. It should be noted that these shell forms represent the converged solutions obtained via BCMO/k-NNC within a limit of 100 generations and a population of 50 individuals. Due to the stochastic nature of BCMO, these are near-optimal shapes rather than guaranteed global minima; however, the trend of shifting the load-bearing mechanism from bending to membrane action remains consistent and reliable across multiple runs.

Table 14: Optimized shape of composite shells with different laminated configurations.

Configuration	Optimized Shape—XZ View	Optimized Shape—YZ View	Strain Energy (J)
$[0^\circ/90^\circ/90^\circ/0^\circ]$			41.8010
$[0^\circ/90^\circ/0^\circ/90^\circ]$			41.2676
$[45^\circ/-45^\circ/-45^\circ/45^\circ]$			39.3222
$[45^\circ/-45^\circ/45^\circ/-45^\circ]$			38.2606

To comprehensively evaluate the mechanical efficiency of the proposed shape optimization framework, the maximum internal forces of the initial (flat surface) and optimal composite shells are summarized in Table 15. The numerical results demonstrate a fundamental shift in the structural load-carrying mechanism, transitioning from a bending-dominated state to a highly efficient membrane-dominated behavior across all laminate configurations. For the initial flat shape, the shells exhibit critically high bending moments (M_{xx} and M_{yy}) while the membrane forces (N_{xx} and N_{yy}) are either completely zero (in the $[0^\circ/90^\circ/90^\circ/0^\circ]$ and $[45^\circ/-45^\circ/-45^\circ/45^\circ]$ layups) or excessively high (in the $[0^\circ/90^\circ/0^\circ/90^\circ]$ and $[45^\circ/-45^\circ/45^\circ/-45^\circ]$ layups). For instance, the initial $[0^\circ/90^\circ/0^\circ/90^\circ]$ shell experiences severe bending actions ($M_{xx} = 41.52$ kNm/m, $M_{yy} = 42.48$ kNm/m). However, after undergoing shape optimization by the BCMO/k-NNC algorithm, the optimal geometry effectively introduces structural curvature. This curvature activates the arching effect, causing the bending moments to drastically decrease by up to 99% (e.g., M_{xx} drops to 0.27 kNm/m and 0.23 kNm/m for cross-ply laminates). Concurrently, the vertical external loads are successfully converted into in-plane membrane forces, as evidenced by the substantial increase in N_{xx} and N_{yy} for the optimized profiles.

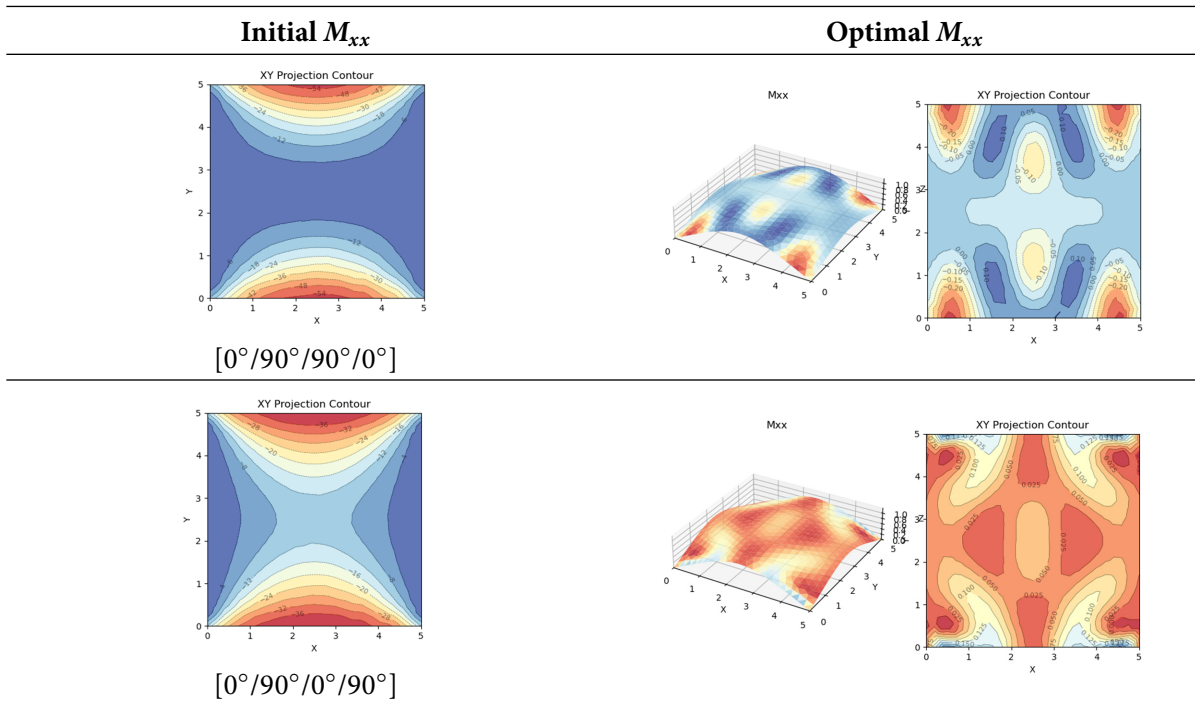
Table 15: Maximum internal force of initial and optimal composite shells.

Configuration	N_{xx} (kN/m)		N_{yy} (kN/m)		M_{xx} (kNm/m)		M_{yy} (kNm/m)	
	Initial	Optimal	Initial	Optimal	Initial	Optimal	Initial	Optimal
$[0^\circ/90^\circ/90^\circ/0^\circ]$	0	237.57	0	305.30	57.83	0.27	26.56	0.07
$[0^\circ/90^\circ/0^\circ/90^\circ]$	2190.77	242.96	2135.47	275.65	41.52	0.23	42.48	0.44
$[45^\circ/-45^\circ/-45^\circ/45^\circ]$	0	196.36	0	232.01	27.71	0.42	27.59	0.29
$[45^\circ/-45^\circ/45^\circ/-45^\circ]$	1965.21	144.96	1974.95	212.53	27.11	0.44	27.05	0.44

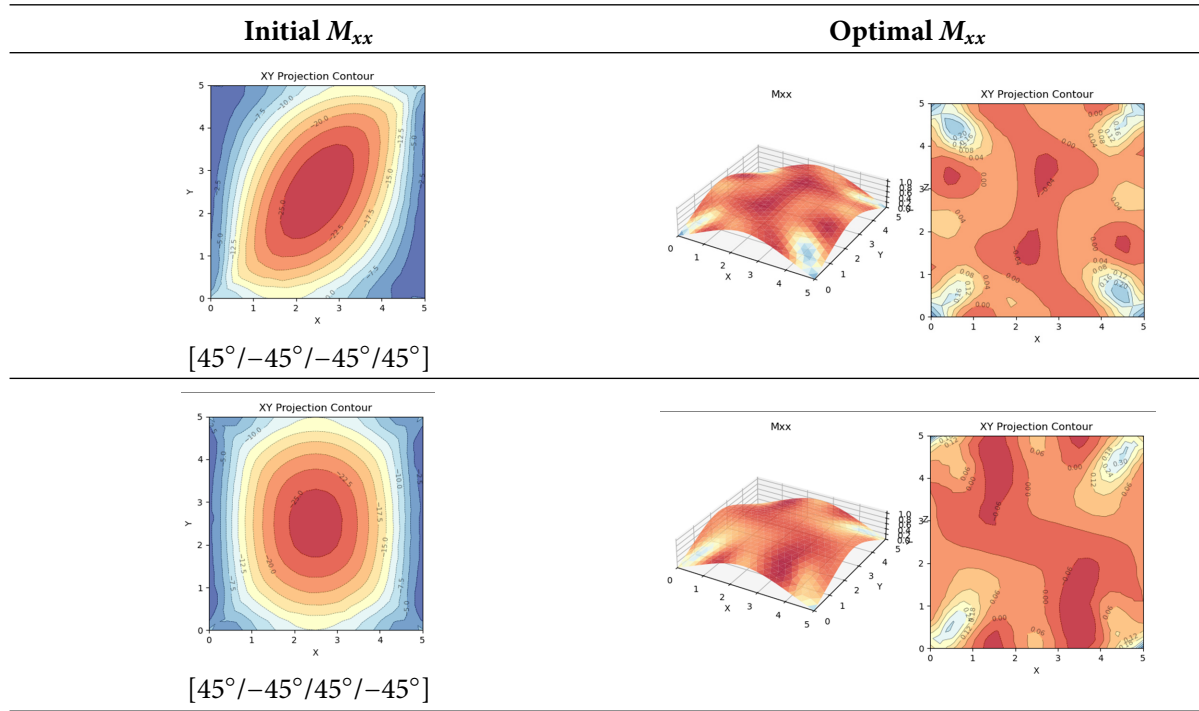
It is worth noting in Table 15 that the initial configurations of the unsymmetric laminates exhibit substantial membrane forces (reaching over 2000 kN/m) under purely transverse loading, whereas their symmetric counterparts yield exactly zero membrane forces. This phenomenon is a direct consequence of the bending-extension coupling inherent to unsymmetric layups, mathematically governed by the non-zero coupling stiffness matrix $\mathbf{D}_{mb}$ (see Table A2 in Appendix C). Under transverse displacement, the induced curvatures κ couple with $\mathbf{D}_{mb}$ to generate internal membrane stress resultants. Because the corners of the shell are modeled with in-plane immovable restraints ($u_x = 0, u_y = 0$), the structure cannot undergo rigid body relaxation. Consequently, the coupling effect induces high-magnitude membrane forces near the zones of maximum curvature change (see Figs. A1 and A2 in Appendix C). Global equilibrium checks were performed by verifying that the boundary reaction integrals perfectly counterbalanced the applied transverse force vector, validating the structural sanity of the CS-DSG3-A solver under intense coupling states. After optimization, the bending behavior is significantly reduced, resulting in large reduction of bending-extension effect and the induced membrane forces.

The distribution of bending moment M_{xx} of initial and optimal shapes is illustrated graphically in Table 16. In the initial flat shell, M_{xx} is concentrated in specific regions, while the bending moment decreases sharply and is evenly distributed across the shell surface after optimization. This demonstrates that the geometric curvature obtained from optimization has activated the membrane load-bearing capacity, eliminating the initial severe concentration of bending stress. Furthermore, the layer structure directly influences the moment distribution in the initial state, but its influence is less pronounced after the shell has been optimized.

Table 16: Illustration of the bending moment M_{xx} of initial and optimal composite shells.



(Continued)

Table 16 (continued)

To quantitatively assess the computational efficiency of the integrated framework, a thorough profiling of the CPU time spent on each operational phase within a single evaluation was performed. The average execution times and their corresponding percentages are compiled in Table 17 (for $l_c = 0.5$). The numerical profile demonstrates that the structural finite element analysis is the main computational bottleneck, consuming 89.023% of the total budget. In contrast, the k-NNC classifier requires only 5.0×10^{-4} s per call, representing a minor fraction (0.039%) of the total execution time. This quantitative evidence supports the premise that the overhead introduced by the k-NNC algorithm is practically negligible for the investigated design complexities.

Table 17: Computational time profiling per design evaluation in the laminate shell optimization.

Phase	Average Time (s)	Percentage (%)	BCMO	BCMO/k-NNC
New candidate creation	6.00e-5	0.005	Yes	Yes
k-NNC filtering	5.00e-4	0.039	No	Yes
Gmsh geometry & meshing	0.07	5.466	Yes	Yes
FEM solver analysis	1.14	89.023	Yes	Yes
Fitness evaluation	0.07	5.466	Yes	Yes

7 Conclusion

This paper has presented an efficient, automated, and integrated Python-based workflow for the shape optimization of laminated composite shells. Within the proposed framework, an enhanced 3-node triangular flat shell element (CS-DSG3-A) incorporating Allman membrane strains has been successfully

deployed to effectively eliminate both shear locking and stiffness matrix degeneracy. Numerical investigations demonstrate that this shell element model achieves high stability and accuracy while operating on thousands of continuously changing meshes throughout the optimization loop. By fully separating the geometric modeling—parameterized via spline curves and control points by Gmsh API—from the structural finite element analysis, the dimensionality of the design space is significantly reduced compared to traditional nodal-based FEM, while ensuring geometric smoothness. The synchronous integration of the metaheuristic Balancing Composite Motion Optimization (BCMO) algorithm and the k-Nearest Neighbor Comparison (k-NNC) classification model provides a substantial reduction in computational effort. Acting as a lightweight binary surrogate filter, the k-NNC mechanism successfully screens and discards unpromising design candidates before executing the expensive finite element solver. Most notably, the integration of this filter reduces the overall computational burden by 56% to 70% in terms of the number of function evaluations (NFEs). However, this efficiency-accuracy trade-off implies that the large reduction in NFEs may be accompanied by a noticeable loss in the best attainable objective value for highly multimodal layouts, such as angle-ply laminates, where standard BCMO maintains a stronger global exploration capabilities. Nonetheless, the optimized shell profiles consistently reveal a fundamental transition in the load-carrying mechanism, evolving from a bending-dominated state to a highly membrane-dominated mechanism.

Although the proposed workflow has proven robust and conceptually reproducible through the provided formulations, several directions can be pursued for future research. First, the geometric shape optimization could be expanded into a simultaneous, multi-objective optimization problem that co-optimizes the shell surface profile along with the laminate stacking sequence configurations (e.g., fiber orientation, layer thickness, and material layout). Second, other constraints like interlaminar stresses controlling the ply-level failure, natural vibration frequencies and structural stability limitations against buckling under complex environments, should be integrated into the optimization pipeline to fulfill more comprehensive and demanding real-world engineering applications. Additionally, a primary geometric limitation of the current framework is that only the z-coordinates of the spline control points are assigned as design variables, keeping the x- and y-coordinates fixed to guarantee mesh validity and preserve in-plane boundary conditions. Consequently, the optimization drive is confined to height and curvature modifications rather than true general free-form shape optimization. Future expansions will focus on overcoming this constraint by allowing multi-directional control point movement, paired with advanced moving-boundary or in-plane reshaping techniques, to achieve a wider and more flexible design domain.

Acknowledgement: Not applicable.

Funding Statement: This research is funded by Hanoi University of Civil Engineering (HUCE) under grant number 44-2025/KHXD-TĐ.

Author Contributions: The authors confirm contribution to the paper as follows: conceptualization, Hoang-Anh Pham; methodology, Hoang-Anh Pham; software, Hoang-Anh Pham; validation, Hoang-Anh Pham; formal analysis, Hoang-Anh Pham; investigation, Van-Binh Tran, Tien-Chuong Vu; resources, Hoang-Anh Pham, Van-Binh Tran, Tien-Chuong Vu; data curation, Hoang-Anh Pham; writing—original draft preparation, Hoang-Anh Pham; writing—review and editing, Hoang-Anh Pham, Minh-Tu Tran; visualization, Van-Binh Tran, Tien-Chuong Vu; supervision, Minh-Tu Tran; project administration, Hoang-Anh Pham; funding acquisition, Hoang-Anh Pham, Van-Binh Tran. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The authors confirm that the data supporting the findings of this study are available within the article. Due to institutional and proprietary constraints, the integrated custom Python source code is not publicly available in a repository. The source code may be made available by the corresponding author upon reasonable academic request.

Ethics Approval: Not applicable.

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

Appendix A

Membrane gradient matrix $\mathbf{B}_m$ and matrix $\mathbf{C}$:

According to Allman [7], the membrane strain vector of the element ϵ_e ($\epsilon_e = \{\epsilon_x, \epsilon_y, \epsilon_{xy}\}^T$) is calculated as

$$\epsilon_e = \mathbf{B}_0 \alpha \quad (\text{A1})$$

where α is a vector containing five strain components, $\alpha = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}^T$, determined by

$$\alpha = \mathbf{C} \mathbf{d}_e \quad (\text{A2})$$

with $\mathbf{d}_e$ is the vector of 9 membrane nodal displacements, i.e., $\mathbf{d}_e = \{u_1, v_1, \beta_{z1}, u_2, v_2, \beta_{z2}, u_3, v_3, \beta_{z3}\}^T$.

The matrix $\mathbf{C}$ is a 5×9 matrix given by

$$\mathbf{C} = \begin{bmatrix} y_{23} & 0 & 1/2 y_1 y_{23} & y_{31} & 0 & 1/2 y_2 y_{31} & y_{12} & 0 & 1/2 y_3 y_{12} \\ 0 & 0 & -1/2 y_{23} & 0 & 0 & -1/2 y_{31} & 0 & 0 & -1/2 y_{12} \\ 0 & x_{32} & 1/2 x_1 x_{23} & 0 & x_{13} & 1/2 x_2 x_{31} & 0 & x_{21} & 1/2 x_3 x_{12} \\ 0 & 0 & -1/2 x_{23} & 0 & 0 & -1/2 x_{31} & 0 & 0 & -1/2 x_{12} \\ x_{32} & y_{23} & 1/2(x_1 y_{32} + y_1 x_{32}) & x_{13} & y_{31} & 1/2(x_2 y_{13} + y_2 x_{13}) & x_{21} & y_{12} & 1/2(x_3 y_{21} + y_3 x_{21}) \end{bmatrix} \quad (\text{A3})$$

where x_i, y_i ($i = 1, 2, 3$) are, respectively, the x-coordinate and y-coordinate of the i -th node; $x_{ij} = x_i - x_j$, and $y_{ij} = y_i - y_j$.

Combining Eqs. (A1) and (A2) gives

$$\epsilon_e = \mathbf{B}_0 \mathbf{C} \mathbf{d}_e = \mathbf{B}_m \mathbf{d}_e \quad (\text{A4})$$

Thus, the membrane gradient matrix is $\mathbf{B}_m = \mathbf{B}_0 \mathbf{C}$.

Appendix B

Mesh convergence and sensitivity verification

To verify the numerical stability of the optimized solutions, a mesh-sensitivity study was conducted on the optimized $[45^\circ/-45^\circ/45^\circ/-45^\circ]$ shell configuration. The Gmsh characteristic length lc was varied across four levels: 1.0, 0.75, 0.5, and 0.25 m. As shown in Table A1, the baseline mesh utilized in this study ($lc = 0.5$) yields a strain energy of 38.2606 J. Further refinement to $lc = 0.25$ results in a strain energy of 38.5048 J, corresponding to a minor variance of just 0.64% despite a significant increase in element count and CPU time. Thus, the chosen mesh density ($lc = 0.5$) guarantees sufficient numerical accuracy for the shape optimization process while maintaining a reasonable computational budget.

Table A1: Mesh-sensitivity verification results for optimized $[45^\circ/-45^\circ/45^\circ/-45^\circ]$ shell configuration.

lc	1 m	0.75 m	0.5 m	0.25 m
No. nodes	54	86	158	561
No. elements	82	138	270	1032
CPU time (s)	0.4149	0.6659	1.2188	4.4776
Strain energy (J)	49.4677	44.8044	38.2606	38.5048

Appendix C

Elastic matrices of the initial flat configuration in the laminate reference system

Table A2 presents the representative laminate stiffness matrices ($\mathbf{D}_m$, $\mathbf{D}_{mb}$, $\mathbf{D}_b$) of the initial flat laminated shell in [Section 6.3](#). For the unsymmetric layups, the presence of non-zero coefficients in the $\mathbf{D}_{mb}$ matrix establishes a strong coupling mechanism, which, under the immovable corner restraints, leads to the development of the very high membrane forces observed in the contour plots ([Figs. A1 and A2](#)).

Table A2: Representative laminate stiffness matrices ($\mathbf{D}_m$, $\mathbf{D}_{mb}$, $\mathbf{D}_b$) of the initial flat laminated shells.

	$\mathbf{D}_m$	$\mathbf{D}_{mb}$	$\mathbf{D}_b$
[0/90/90/0]	$\begin{bmatrix} 2.27\text{e6} & 8.49\text{e4} & 0 \\ 8.49\text{e4} & 2.27\text{e6} & 0 \\ 0 & 0 & 1.73\text{e5} \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1.96\text{e2} & 4.42 & 0 \\ 4.42 & 4.06\text{e1} & 0 \\ 0 & 0 & 8.98 \end{bmatrix}$
[0/90/0/90]	$\begin{bmatrix} 2.27\text{e6} & 8.49\text{e4} & 0 \\ 8.49\text{e4} & 2.27\text{e6} & 0 \\ 0 & 0 & 1.73\text{e5} \end{bmatrix}$	$\begin{bmatrix} -6.21\text{e3} & 0 & 0 \\ 0 & 6.21\text{e3} & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1.18\text{e2} & 4.42 & 0 \\ 4.42 & 1.18\text{e2} & 0 \\ 0 & 0 & 8.98 \end{bmatrix}$
[45/-45/-45/45]	$\begin{bmatrix} 1.35\text{e6} & 1.01\text{e6} & 0 \\ 1.01\text{e6} & 1.35\text{e6} & 0 \\ 0 & 0 & 1.09\text{e6} \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 70.32 & 52.35 & 38.82 \\ 52.35 & 70.32 & 38.82 \\ 38.82 & 38.82 & 56.92 \end{bmatrix}$
[45/-45/45/-45]	$\begin{bmatrix} 1.35\text{e6} & 1.01\text{e6} & 0 \\ 1.01\text{e6} & 1.35\text{e6} & 0 \\ 0 & 0 & 1.09\text{e6} \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & -3.11\text{e3} \\ 0 & 0 & -3.11\text{e3} \\ -3.11\text{e3} & -3.11\text{e3} & 0 \end{bmatrix}$	$\begin{bmatrix} 70.32 & 52.35 & 0 \\ 52.35 & 70.32 & 0 \\ 0 & 0 & 56.92 \end{bmatrix}$

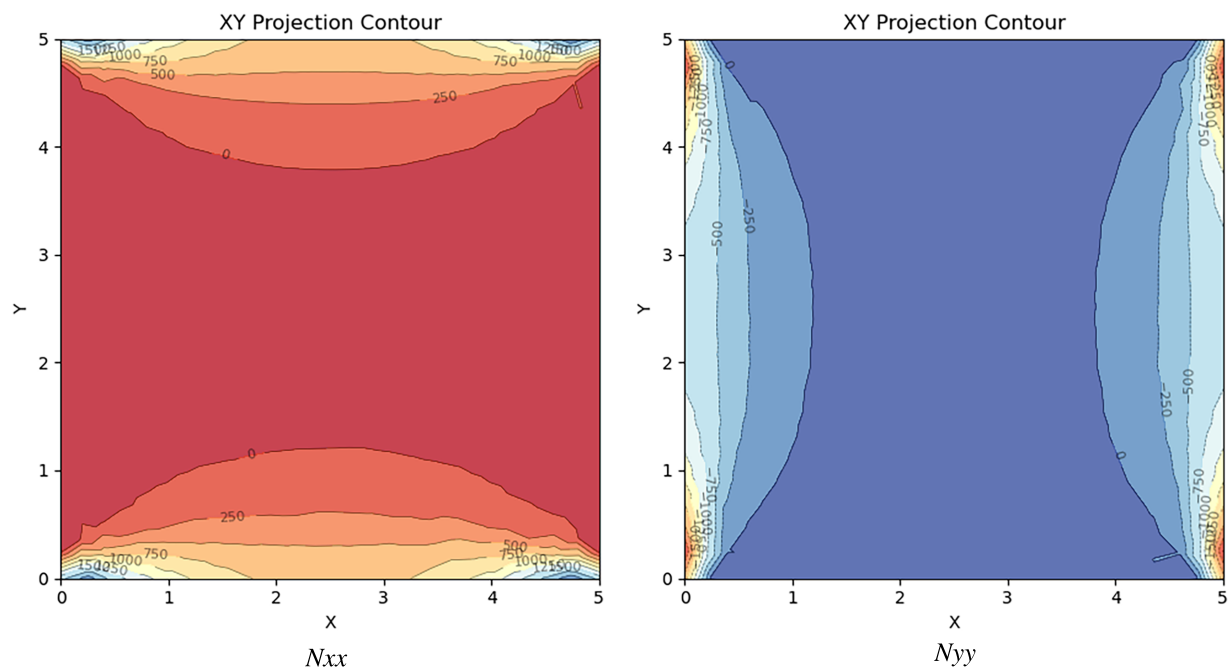


Figure A1: Contour plot (N_{xx} , N_{yy}) for laminate configuration $[0^\circ/90^\circ/0^\circ/90^\circ]$.

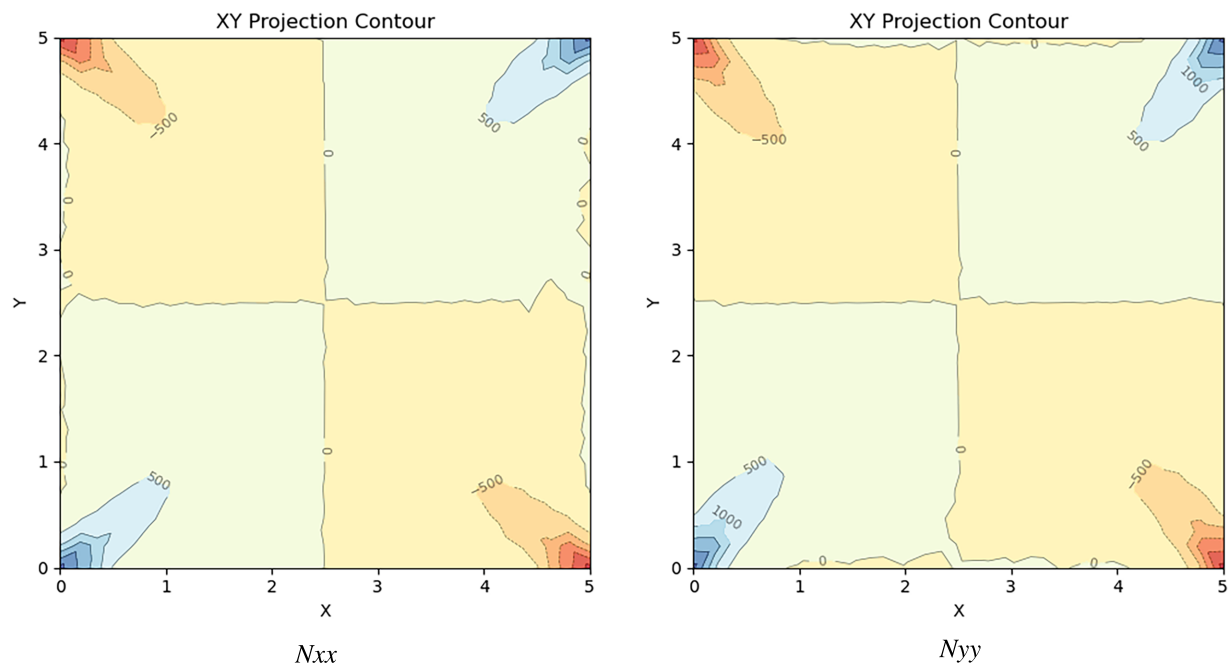


Figure A2: Contour plot (N_{xx} , N_{yy}) for laminate configuration $[45^\circ/-45^\circ/45^\circ/-45^\circ]$

References

1. Reddy JN. *Mechanics of laminated composite plates and shells: theory and analysis*. Boca Raton, FL, USA: CRC Press; 2003, 10.1201/b12409.
2. Sadraee Far MN, Foroozandeh S, Golmakani ME, Sadeghian M. Three-dimensional analysis of FG-CNTRC shell subjected to internal pressure utilizing Carrera Unified Formulation. *Mech Based Des Struct Mach*. 2024;52(8):4865–86. doi:10.1080/15397734.2023.2242472.
3. Bletzinger KU, Bischoff M, Ramm E. A unified approach for shear-locking-free triangular and rectangular shell finite elements. *Comput Struct*. 2000;75(3):321–34. doi:10.1016/s0045-7949(99)00140-6.
4. Nguyen-Thoi T, Phung-Van P, Thai-Hoang C, Nguyen-Xuan H. A cell-based smoothed discrete shear gap method (CS-DSG3) using triangular elements for static and free vibration analyses of shell structures. *Int J Mech Sci*. 2013;74(1):32–45. doi:10.1016/j.ijmecsci.2013.04.005.
5. Hoa PQ, Van TT, Dat PT, Hau DT, Ha NV, Hung NM, et al. Static and free vibration analyses of laminated composite shells by cell-based smoothed discrete shear gap method (CS-DSG3) using three-node triangular elements. *Vietnam J Mech*. 2018;40(1):89–103. doi:10.15625/0866-7136/10579.
6. Le-Xuan D, Pham-Quoc H, Tran-The V, Nguyen-Van N. Static and free vibration analysis of functionally graded shells using a cell-based smoothed discrete shear gap method and triangular elements. In: *Proceedings of the International Conference on Advances in Computational Mechanics 2017*. Singapore: Springer; 2018. p. 381–94. doi:10.1007/978-981-10-7149-2_26.
7. Allman DJ. A compatible triangular element including vertex rotations for plane elasticity analysis. *Comput Struct*. 1984;19(1-2):0045794984901974. doi:10.1016/0045-7949(84)90197-4.
8. Cook RD. A plane hybrid element with rotational d.o.f. and adjustable stiffness. *Numer Meth Eng*. 1987;24(8):1499–508. doi:10.1002/nme.1620240807.
9. Cook RD. Further development of a three-node triangular shell element. *Numer Meth Eng*. 1993;36(8):1413–25. doi:10.1002/nme.1620360809.
10. Nguyen-Minh N, Nguyen-Thoi T, Bui-Xuan T, Vo-Duy T. Static and free vibration analyses of stiffened folded plates using a cell-based smoothed discrete shear gap method (CS-FEM-DSG3). *Appl Math Comput*. 2015;266:212–34. doi:10.1016/j.amc.2015.05.042.
11. Le-Anh L, Nguyen-Thoi T, Ho-Huu V, Dang-Trung H, Bui-Xuan T. Static and frequency optimization of folded laminated composite plates using an adjusted Differential Evolution algorithm and a smoothed triangular plate element. *Compos Struct*. 2015;127(2):382–94. doi:10.1016/j.compstruct.2015.02.069.
12. Nguyen-Thoi T, Bui-Xuan T, Liu GR, Vo-Duy T. Static and free vibration analysis of stiffened flat shells by a cell-based smoothed discrete shear gap method (CS-FEM-DSG3) using three-node triangular elements. *Int J Comput Methods*. 2018;15(6):1850056. doi:10.1142/s0219876218500561.
13. Chau-Dinh T. Analysis of shell structures by an improved 3-node triangular flat shell element with a bubble function and cell-based strain smoothing. *Thin Walled Struct*. 2023;182:110222. doi:10.1016/j.tws.2022.110222.
14. Chau-Dinh T, Cao-Nhu T, Le-Phuong B, Ton-That HL. A CS-MITC18+ shell element for static and frequency analyses of laminated composite plate and shell structures. *Arch Appl Mech*. 2024;94(7):2059–83. doi:10.1007/s00419-024-02627-0.
15. Le-Phuong B, Chau-Dinh T. Static and free vibration analyses of multilayer FG GPLRC open shells using a TSDT-type CS-MITC3+ flat shell element. *Compos Struct*. 2026;386:120279. doi:10.1016/j.compstruct.2026.120279.
16. Nguyen SH. Linear composite curvature MITC3+ flat shell elements. *Comput Struct*. 2024;295(3):107272. doi:10.1016/j.compstruc.2024.107272.
17. Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput Meth Appl Mech Eng*. 2005;194(39–41):4135–95. doi:10.1016/j.cma.2004.10.008.
18. Nguyen VP, Anitescu C, Bordas SPA, Rabczuk T. Isogeometric analysis: an overview and computer implementation aspects. *Math Comput Simul*. 2015;117(0):89–116. doi:10.1016/j.matcom.2015.05.008.
19. Wang Y, Wang Z, Xia Z, Hien Poh L. Structural design optimization using isogeometric analysis: a comprehensive review. *Comput Model Eng Sci*. 2018;117(3):455–507. doi:10.31614/cmes.2018.04603.

20. Ruess M, Schillinger D, Özcan AI, Rank E. Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries. *Comput Meth Appl Mech Eng.* 2014;269(21–26):46–71. doi:10.1016/j.cma.2013.10.009.
21. Yu Y, Wang Y, Lin Y. IsoGeometric Analysis with non-conforming multi-patches for the hull structural mechanical analysis. *Thin Walled Struct.* 2023;187:110757. doi:10.1016/j.tws.2023.110757.
22. Gervasio P, Marini F. The INTERNODES method for the treatment of non-conforming multipatch geometries in Isogeometric Analysis. *Comput Meth Appl Mech Eng.* 2020;358(21–22):112630. doi:10.1016/j.cma.2019.112630.
23. Zhao H, Kamensky D, Hwang JT, Chen JS. Automated shape and thickness optimization for non-matching isogeometric shells using free-form deformation. *Eng Comput.* 2024;40(6):3495–518. doi:10.1007/s00366-024-01947-7.
24. Zienkiewicz OC, Campbell JS. Shape optimization and sequential linear programming. In: Gallagher R, Zienkiewicz OC, editors. *Optimum structural design.* New York, NY, USA: Wiley; 1973. p. 109–26.
25. Braibant V, Fleury C. Shape optimal design using B-splines. *Comput Meth Appl Mech Eng.* 1984;44(3):247–67. doi:10.1016/0045-7825(84)90132-4.
26. Wu J, Burgueño R. An integrated approach to shape and laminate stacking sequence optimization of free-form FRP shells. *Comput Meth Appl Mech Eng.* 2006;195(33–36):4106–23. doi:10.1016/j.cma.2005.07.015.
27. Zhu S, Ohsaki M, Guo X, Zeng Q. Shape optimization for non-linear buckling load of aluminum alloy reticulated shells with gusset joints. *Thin Walled Struct.* 2020;154(3):106830. doi:10.1016/j.tws.2020.106830.
28. Biancolini ME. *Fast radial basis functions for engineering applications.* Cham, Switzerland: Springer International Publishing; 2017. doi:10.1007/978-3-319-75011-8.
29. Abergo L, Morelli M, Guardone A. Aerodynamic shape optimization based on discrete adjoint and RBF. *J Comput Phys.* 2023;477:111951. doi:10.1016/j.jcp.2023.111951.
30. Nikbakt S, Kamarian S, Shakeri M. A review on optimization of composite structures Part I: laminated composites. *Compos Struct.* 2018;195(1):158–85. doi:10.1016/j.compstruct.2018.03.063.
31. Zhang X, Sun G, Wang C, Li H, Zhou S. A review of structural topology optimization for fiber-reinforced composites. *Compos Part B Eng.* 2025;299(2):112393. doi:10.1016/j.compositesb.2025.112393.
32. Holland JH. Genetic algorithms. *Sci Am.* 1992;267(1):66–72. doi:10.1038/scientificamerican0792-66.
33. Kennedy J, Eberhart R. Particle swarm optimization. In: *Proceedings of ICNN'95—International Conference on Neural Networks*; 1998 Nov 27–Dec 1; Perth, WA, Australia. p. 1942–8. doi:10.1109/ICNN.1995.488968.
34. Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim.* 1997;11(4):341–59. doi:10.1023/A:1008202821328.
35. Le-Duc T, Nguyen QH, Nguyen-Xuan H. Balancing composite motion optimization. *Inf Sci.* 2020;520:250–70. doi:10.1016/j.ins.2020.02.013.
36. Thanh Duong H, Chi Phan H, Le TT, Duc Bui N. Optimization design of rectangular concrete-filled steel tube short columns with Balancing Composite Motion Optimization and data-driven model. *Structures.* 2020;28(2):757–65. doi:10.1016/j.istruc.2020.09.013.
37. Phan HC, Le TT, Bui ND, Duong HT, Pham TD. An empirical model for bending capacity of defected pipe combined with axial load. *Int J Press Vessels Pip.* 2021;191(6):104368. doi:10.1016/j.ijpvp.2021.104368.
38. Le-Thanh L, Le-Duc T, Ngo-Minh H, Nguyen QH, Nguyen-Xuan H. Optimal design of an Origami-inspired kinetic façade by balancing composite motion optimization for improving daylight performance and energy efficiency. *Energy.* 2021;219:119557. doi:10.1016/j.energy.2020.119557.
39. Nguyen CH. Damage detection in beam structures using bayesian deep learning and balancing composite motion optimization. In: *Structures.* Amsterdam, The Netherlands: Elsevier; 2022. p. 98–114. doi:10.1016/j.istruc.2022.03.030.
40. Pensupa P, Le TM, Rungamornrat J. An ANN-BCMO approach for material distribution optimization of bi-directional functionally graded nanocomposite plates with geometrically nonlinear behaviors. *Int J Comput Methods.* 2024;21(10):2341010. doi:10.1142/s0219876223410104.
41. Tran VT, Nguyen TK, Nguyen-Xuan H, Vo TP. Meta-heuristic optimization algorithms for vibration and buckling analysis of laminated composite plates. *Eng Anal Bound Elem.* 2024;169(8):105974. doi:10.1016/j.enganabound.2024.105974.

42. Huang T, Wang Q, Chen L, Zhong R. Kriging-based uncertainty optimization of vibration characteristics for laminated elliptical shells considering material and load uncertainties. *Eur J Mech A*. 2025;111(148):105587. doi:10.1016/j.euromechsol.2025.105587.
43. Wang X, Honda S, Sasaki K, Takeda R. Sequential approximate optimization of curvilinear fiber paths for enhanced vibration performance in composite shells. *Compos Struct*. 2025;372(21):119532. doi:10.1016/j.compstruct.2025.119532.
44. Tran VT, Nguyen TK, Nguyen-Xuan H, Abdel Wahab M. Vibration and buckling optimization of functionally graded porous microplates using BCMO-ANN algorithm. *Thin Walled Struct*. 2023;182(10):110267. doi:10.1016/j.tws.2022.110267.
45. dos Santos RR, de Paula Machado TG, Castro SGP. Support vector machine applied to the optimal design of composite wing panels. *Aerospace*. 2021;8(11):328. doi:10.3390/aerospace8110328.
46. Pham HA, Dang VH, Vu TC, Nguyen BD. An efficient k-NN-based *Rao* optimization method for optimal discrete sizing of truss structures. *Appl Soft Comput*. 2024;154(2):111373. doi:10.1016/j.asoc.2024.111373.
47. Nguyen AV, Vu TC, Nguyen BD, Pham HA, Rao RV. Weighted k-NNC: an efficient computation reduction method for metaheuristic-based structural optimization. *Comput Model Eng Sci*. 2026;147(1):1–10. doi:10.32604/cmesci.2026.080453.
48. Geuzaine C, Remacle JF. Gmsh: a 3-D finite element mesh generator with built-in pre- and post-processing facilities. *Numerical Meth Engineering*. 2009;79(11):1309–31. doi:10.1002/nme.2579.
49. Nguyen-Thoi T, Phung-Van P, Nguyen-Xuan H, Thai-Hoang C. A cell-based smoothed discrete shear gap method using triangular elements for static and free vibration analyses of Reissner–Mindlin plates. *Numer Meth Eng*. 2012;91(7):705–41. doi:10.1002/nme.4289.
50. Bischoff M, Bletzinger KU. Stabilized DSG plate and shell elements. In: *Trends in computational structural mechanics*. Barcelona, Spain: CIMNE; 2001. p. 253–63.
51. Deb K. An efficient constraint handling method for genetic algorithms. *Comput Meth Appl Mech Eng*. 2000;186(2–4):311–38. doi:10.1016/s0045-7825(99)00389-8.
52. Krysl P, Chen JS. Benchmarking computational shell models. *Arch Comput Meth Eng*. 2023;30(1):301–15. doi:10.1007/s11831-022-09798-5.
53. Reddy JN. Exact solutions of moderately thick laminated shells. *J Eng Mech*. 1984;110(5):794–809. doi:10.1061/(asce)0733-9399(1984)110:.
54. Yang F, Yu T, Bui C. Morphogenesis of free-form surfaces by an effective approach based on isogeometric analysis and particle swarm optimization. In: *Structures*. Amsterdam, The Netherlands: Elsevier; 2023. p. 2347–53. doi:10.1016/j.istruc.2022.12.069.
55. Zhang Z, Yu H, Wu H, Chen Q. A simultaneous shape and topology optimization approach of shell structures based on isogeometric analysis and density distribution field. *Comput Struct*. 2024;305:107550. doi:10.1016/j.compstruc.2024.107550.