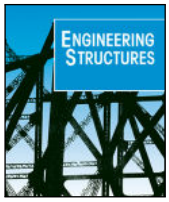
Engineering Structures 284 (2023) 116016

Contents lists available at ScienceDirect

Engineering Structures

journal homepage: www.elsevier.com/locate/engstruct



Reducing the number of different nodes in space frame structures through clustering and optimization

Yuanpeng Liu [a], Ting-Uei Lee [a], Antiopi Koronaki [b], Nico Pietroni [c], Yi Min Xie [a,*]

[a] Centre for Innovative Structures and Materials, School of Engineering, RMIT University, Melbourne, 3001, Australia
[b] Centre for Natural Material Innovation, Department of Architecture, University of Cambridge, 1-5 Scroope Terrace, Cambridge CB2 1PX, United Kingdom
[c] School of Computer Science, University of Technology Sydney, Sydney, 2007, Australia

ARTICLE INFO

Keywords:
Architectural geometry
Clustering
Optimization
Space frame structure
Free-form surface
Node design

ABSTRACT

Space frame structures are increasingly adopted in contemporary free-form architectural designs due to their elegant appearance and excellent structural performance. However, a space frame structure in a doubly-curved form typically comprises nodes of different shapes. This often requires extensive node customization, hence incurring high manufacturing costs. In this study, we propose a new clustering–optimization framework to reduce the number of different nodes in space frame structures. In clustering, nodes are divided into different groups, with similar shapes grouped together, using an enhanced k-means clustering technique. In optimization, nodes within the same group are transformed towards congruence while closely approximating the target surface. Together, by interleaving clustering and optimization, our method can minimize the node shape variety under a user-defined error threshold. The effectiveness of the method is validated through a variety of numerical examples. The potential practical application of our method is demonstrated by re-designing a complex, free-form architectural project.

1. Introduction

Space frames are rigid, lightweight structures composed of a network of linear struts intersected at three-dimensional nodes. They are usually arranged in an array of single, double, or multiple layers of grids. Due to their excellent mechanical performance [1] and visual beauty, such structural forms are favored for architectural applications. Specifically, they are increasingly adopted in contemporary designs that comprise elegant appearances in free-form configurations. However, free-form structures, characterized by their complex doubly-curved geometries, are often extremely difficult and costly to realize in reality. Therefore, in the last decade, significant effort has been focused on rationalizing free-form structures to achieve feasible construction with reduced manufacturing costs [2–4].

A free-form structure, which is often represented as a polygonal mesh, can be rationalized in different ways [5,6]. One main strategy is to planarize the mesh faces to ease panel fabrication. Triangular meshes [7] contain faces that are naturally planar. However, due to their high-valence nodes that are typically difficult to manufacture, further studies have investigated meshes with low-valence nodes, such as quadrilateral meshes [8–10], hexagonal meshes [11–13], and meshes composed of different n-gonal faces [14]. Another strategy is to reduce the number of different elements, including edges, faces, and nodes, to enable mass production of repeated components [15]. The variety of edges can be reduced using the methods proposed in [16–19]. In terms of faces, the idea of using only congruent regular triangles to create three-dimensional shapes has been discussed in [20,21], which conclude that only limited shapes are attainable. Later, several methods have further attempted to reduce the variety of faces in complex free-form geometries by optimizing all faces into multiple groups of similar [22] or congruent shapes [23–26]. However, regarding nodes, it is still an ongoing challenge to reduce their shape variety in free-form structures, with limited studies available in the literature [18,27,28].

Reducing the shape variety of nodes is a critical consideration for the construction of complex free-form structures. Compared to beams, nodes often contain more complex shapes, which require specialized equipment and techniques for manufacturing [29]. Unlike panels that are typically non-structural components, nodes are crucial in load transmission, thus often subject to stringent fabrication requirements [29]. Mass production can significantly simplify manufacturing and improve quality control for node fabrication [30]. In this regard, minimizing the shape variety of nodes to enable mass production holds great potential for lowering construction costs and enhancing building quality.

Several existing studies have attempted to reduce the node types [18,27,28]. The method described in [27] allows using one type of node and nine different types of edges to approximate various surfaces

* Corresponding author.
E-mail address: mike.xie@rmit.edu.au (Y.M. Xie).

https://doi.org/10.1016/j.engstruct.2023.116016
Received 22 September 2022; Received in revised form 3 March 2023; Accepted 16 March 2023
Available online 21 March 2023

through local topological operations. However, the node contains a high valence, only single layer structures are considered, and generally, only coarse approximations of the target surface can be obtained. A set of struts and nodes is designed in [18] which can be used to construct different structures. To reduce the node variety, several different low-valence nodes are merged into one high-valence node by optimization. However, the optimization goal is to avoid the overlapping of holes through rigid body motions; the shapes of low-valence nodes are not modified. The method stated in [28] seeks to reduce node variety by modifying the shapes of nodes through clustering and optimization. However, the proposed objective function is too complex to derive the corresponding gradient, so the optimization is solved using heuristic methods, which generally converge slowly and only produce coarse approximations of the global optimum. Although the efficiency is improved by using several control points to manipulate all the nodes, thereby reducing the number of design variables, this approach typically limits the extent that the geometry can be optimized.

In this study, we propose a new method to reduce the number of different nodes in space frame structures. The core of the method is a clustering–optimization framework. In clustering, nodes are divided into a specified number of groups, with similar shapes grouped together. In optimization, nodes within the same group are transformed towards the corresponding group centroid while closely approximating the given target surface. By interleaving clustering and optimization, the proposed method can determine the minimal group number required to satisfy a user-defined error threshold. The remainder of this paper is organized as follows: In Section 2, we explain the technical details of clustering and optimization. Section 3 presents various numerical tests for method validation, performance comparison, and parametric investigation. In Section 4, we demonstrate potential practical applications of our algorithm based on the case study of a real, complex, free-form architectural project. Section 5 draws conclusions.

The main contributions of this study are summarized as follows:

- We define an evaluation metric to quantify the similarity between two geometrically different nodes. This metric is based on a mathematical formulation that ensures the minimization of the sum of squared distances between corresponding vertex pairs.
- Based on the proposed similarity metric, we adapt the k-means clustering method [31] to partition nodes into different groups of similar shapes. Five different centroid initialization strategies are compared to investigate their suitability for specific node-clustering problems.
- We develop an effective optimization strategy to transform geometrically different nodes into congruent shapes by equalizing corresponding angles, as verified through three benchmark tests. This strategy leads to a well-formulated objective function with attainable gradient information, which enables efficient optimization using gradient-based methods, making it feasible to solve large-scale practical problems.
- We propose a computational framework that interleaves clustering and optimization to reduce the number of different nodes in space frame structures while satisfying a user-defined error threshold. The effects of input parameters on the outcome are investigated, with corresponding suggestions provided for the parameter selection.
- We demonstrate potential practical applications of the proposed computational framework by redesigning both single and double layer space frame structures, based on the complex free-form geometry of a real architectural project, to achieve cost-effective solutions.

2. Methodology

Clustering and optimization are the key components of our algorithm. In clustering, we aim to divide nodes into different groups of similar shapes. A novel similarity metric is proposed in Section 2.1.1, based on which the overall clustering framework is presented in Section 2.1.2. In optimization, the main goal is to transform nodes within the same group toward congruent shapes, as detailed in Section 2.2.1. An additional goal is considered to preserve the overall shape of the input geometry, as stated in Section 2.2.2. By combining all sub-goals, the global objective function is formulated in Section 2.2.3.

2.1. Clustering

2.1.1. Similarity metric

In clustering problems, to divide input data points into different groups, a metric is typically needed to measure the distance between points, such as the commonly used Euclidean distance [32]. In this study, our goal is to divide nodes into different groups of similar shapes; the distance metric should therefore reflect the geometric similarity between different nodes. Although a metric exists that quantifies the difference between nodes by measuring corresponding angles with respect to their best-fit planes [28], here we propose a new similarity metric with a more solid mathematical basis that guarantees the minimization of the sum of squared distances between corresponding vertex pairs.

Given two different nodes, N_p and N_q, with valence v (the number of edges intersected at a node), their similarity can be calculated as follows: First, we move both nodes to the origin with their neighboring nodes projected onto a unit sphere, so that each node can be normalized and represented as an ordered set of its neighboring vertices. Then, we compute their best-fit configuration by fixing N_p and rotating N_q to minimize the sum, s, of squared distances between corresponding vertex pairs, using the method described in [33]. Because of the different choices of the starting vertex and the order of the neighboring vertices, there are $2v$ possible permutations of N_q. As each permutation represents a different vertex correspondence between N_p and N_q, $2v$ different best-fit configurations need to be considered, each corresponding to a unique s. Among all the obtained s, the minimal one is chosen to determine the similarity metric S. Finally, S can be calculated as Eq. (1), where d_i is the distance between the ith vertex pairs (see Fig. 1) and j is the permutation index.

$$S = \min_{j=1}^{2v} \left(\min \left(\sqrt{\frac{\sum_{i=1}^{v} (d_i)^2}{v}} \right) \right)_j \tag{1}$$

2.1.2. Clustering method

Based on the defined similarity metric, we adapt the widely-used k-means clustering technique [31] to partition nodes into a user-specified number of groups, as shown in Algorithm 1. k-means clustering is simple, fast, and effective, yet it suffers from the major issue that its results are sensitive to the selection of initial centroids [34]. In this regard, further elaboration on centroid initialization is necessary. Two commonly used initialization strategies are k-means++ [34] and farthest point sampling [35]. Both strategies aim to distribute the initial centroids sparsely by gradually increasing the number of centroids by one at each iteration. However, their ideas for selecting the new centroid (Line 5) are different. Specifically, k-means++ chooses the new centroid based on probability through *roulette wheel selection* [36]; the farther a node is from its current centroid, the more likely it will be chosen as the new centroid. On the contrary, the farthest point sampling strategy directly selects the farthest point from its current centroid as the new centroid. To evaluate their performance for the node-clustering problem defined in this study, these two strategies are compared later in Section 3.3. The results suggest that both initialization strategies can enhance the clustering outcome compared to randomly selecting the initial centroids. However, using farthest point sampling tends to yield better results than using k-means++. Additionally, for both strategies, the choice of the starting centroid (Line 1) has a minor impact on the final outcome.

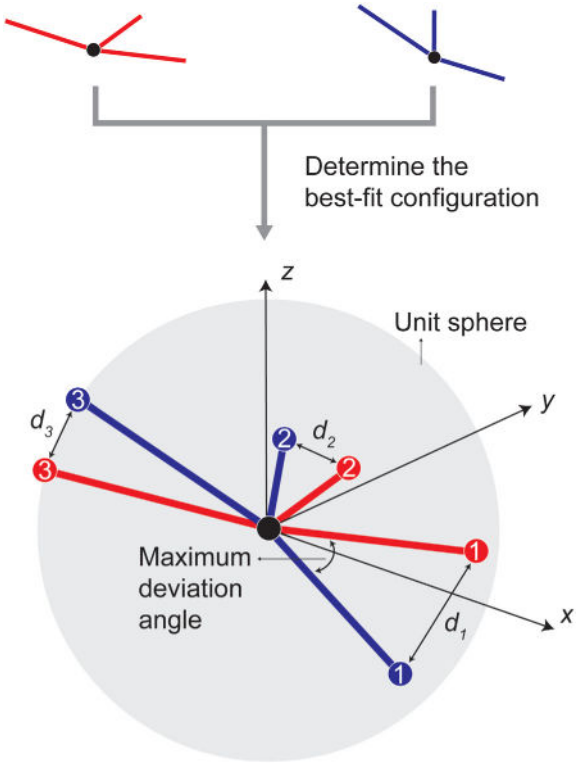


Fig. 1. Given two different nodes (red and blue) with valence three, they are first moved to the origin with their neighboring nodes projected onto a unit sphere. Then, by fixing either one and rotating the other to minimize the sum of squared distances between corresponding vertices (d_1, d_2, and d_3), their best-fit configuration can be determined, based on which one can derive the similarity metric between these two nodes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Algorithm 1 Node clustering procedure

1: choose one node at random as the starting centroid ($k = 1$)
2: **while** $k < K$ **do**
3: calculate the distance from each node to each centroid
4: assign each node to its closest centroid
5: select one node as the new centroid ($k = k + 1$)
6: **end while**
7: define the initial error $\delta = 1$
8: **while** $\delta >$ 1E-3 **do**
9: calculate the distance from each node to each centroid
10: assign each node to its closest centroid
11: for each group, best-fit align all nodes with the centroid to create a group of superimposed nodes
12: in each group, shift each vertex of the centroid to the mean position of the corresponding vertices of the superimposed nodes, and update δ to the maximum displacement
13: **end while**

Note that, to account for practical construction, certain rules are specified in the node clustering procedure. In real applications, a low-valence node N_l can share the same joint with a high-valence node N_h, as long as the holes of N_l can match part of the holes of N_h. This helps to reduce the number of different node molds. Our algorithm considers this by allowing low-valence nodes to be allocated into the groups of high-valence centroids (Lines 4 and 10). However, a high-valence node cannot be divided into the group of a centroid that contains a lower valence number. Therefore, in the clustering framework, the starting centroid (Line 1) is selected from the nodes of the highest valence. Besides, when calculating the distance between N_l and N_h (Eq. (1)), the irreverent vertices of N_h are not involved, and v adopts the valence number of N_l.

2.2. Optimization

2.2.1. Achieving congruent nodes

The main goal of optimization is to transform nodes within the same group into congruent shapes. Specifically, the shape of a node is characterized by the angles between its connected edges, regardless of the edge lengths. One way to achieve congruent nodes is by minimizing the maximum similarity metric between nodes and their respective centroids [28]. However, this optimization problem is inherently discrete due to the consideration of multiple permutations and the selection of the maximum value, making it unsolvable using gradient-based methods. Although heuristic methods can provide solutions, such approaches are typically inefficient and only produce approximate solutions, making them inadequate for large-scale problems with many design variables [37]. Space frames, however, often contain thousands of nodes, each with three design variables. Gradient-based methods are thus much preferred in this study due to their superior efficiency and accuracy [38]. Therefore, our main task is to define an objective function, for which the gradient is available, in order to enable efficient optimization using gradient-based methods.

Here we propose a novel strategy to convert different nodes into congruent shapes by equalizing the corresponding angles between each node and its group centroid. Given a node N with its v neighboring vertices (P_1, P_2, …, P_v), $2v$ angles can be defined as (θ_1, θ_2, …, θ_v, θ_{v+1}, θ_{v+2}, …, θ_{2v}), where θ_i denotes angle $\angle P_i N P_{i+1}$ and θ_{v+i} denotes angle $\angle P_i N P_{i+2}$, as shown in Fig. 2. Constraining the values of these $2v$ angles of a node ensures the node shape is fixed, which can be verified through kinematic analysis [39]. For the system composed of the node and its v neighboring vertices, the total number of degrees of freedom equals $3v + 3$. Specifying the angles brings $2v$ constraints. The rigid body motions of the system account for another 6 degrees of freedom. The neighboring vertices being limited to a unit sphere consumes v additional degrees of freedom. Eventually, the remaining number of degrees of freedom equals $3v + 3 - 2v - 6 - v = -3$. The negative result suggests that the system is statically indeterminate. Thus, no relative movement is allowed for the neighboring vertices, meaning that the node shape is already fixed.

Formally, the corresponding term F_c for equalizing the angles is formulated as Eq. (2), where V is the vertex number of the input mesh, v_i is the valence number of the ith node N_i, θ_{ij} is the jth angle of N_i, $\boldsymbol{n_i}$ is the coordinate vector of N_i, and $\bar{\theta}_{ij}$ is the corresponding angle of the centroid of N_i. The gradient of F_c can be calculated by Eqs. (3)–(5). Eq. (3) gives the overall derivative formula, where $\bar{\theta}_{ij}$ is treated as a constant number, updated at each iteration. The unknown term $\partial\theta_{ij}/\partial\boldsymbol{n_i}$ can be calculated within the triangle that contains both θ_{ij} and N_i. Here, the general problem is to derive the partial derivative of each angle with respect to each vertex in an arbitrary triangle. Given a random triangle $\triangle ABC$ (see Fig. 3), with $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ being the corresponding coordinate vectors of vertices A, B, and C, respectively, $\partial\theta_A/\partial\boldsymbol{a}$ can be obtained using Eq. (4). The unknown term $\partial\cos\theta_A/\partial\boldsymbol{a}$ can be derived from Eq. (5), where all the variables are available from the corresponding geometry at each step. Other cases, such as $\partial\theta_A/\partial\boldsymbol{b}$

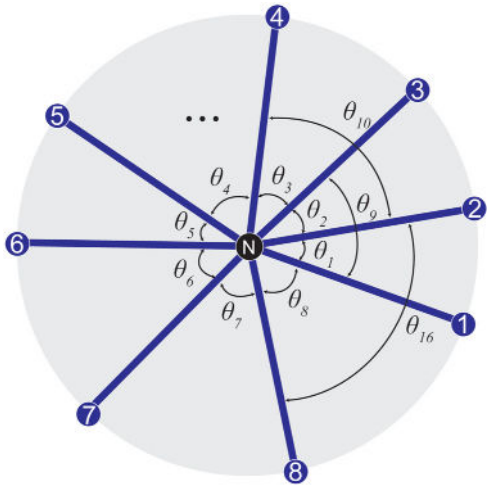


Fig. 2. For a node with eight connected edges, there are sixteen different angles (θ_1, θ_2, …, θ_{16}) that need to be considered during the optimization process.

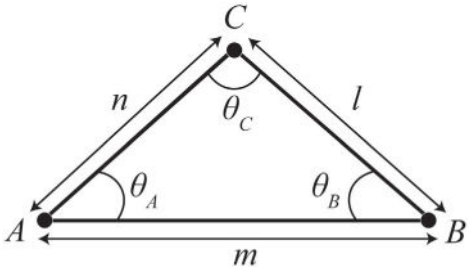


Fig. 3. Triangle $\triangle ABC$ with the labels of its edges and angles.

and $\partial\theta_A/\partial \boldsymbol{c}$, can be calculated in the same manner. Combining all these equations, the gradient of F_c can be obtained, based on which the movement for each vertex to reduce the value of F_c at each iteration can be determined. When F_c is reduced to zero, the corresponding angles are identical for the nodes within the same cluster, which generally ensures their shapes are congruent.

$$F_c = \sum_{i=1}^{V}\sum_{j=1}^{2v_i}(\theta_{ij} - \bar{\theta}_{ij})^2 \tag{2}$$

$$\frac{\partial F_c}{\partial \boldsymbol{n_i}} = \sum_{i=1}^{V}\sum_{j=1}^{2v_i} 2(\theta_{ij} - \bar{\theta}_{ij})\left(\frac{\partial \theta_{ij}}{\partial \boldsymbol{n_i}}\right) \tag{3}$$

$$\frac{\partial \theta_A}{\partial \boldsymbol{a}} = \frac{\partial \arccos(\cos\theta_A)}{\partial \boldsymbol{a}} = \frac{-1}{\sqrt{1-(\cos\theta_A)^2}}\frac{\partial \cos\theta_A}{\partial \boldsymbol{a}} \tag{4}$$

$$\frac{\partial \cos\theta_A}{\partial \boldsymbol{a}} = \frac{\partial \frac{m^2+n^2-l^2}{2nm}}{\partial \boldsymbol{a}} = \frac{2mn(2m\frac{\partial m}{\partial \boldsymbol{a}} + 2n\frac{\partial n}{\partial \boldsymbol{a}} - 2l\frac{\partial l}{\partial \boldsymbol{a}})}{4m^2n^2} - \frac{(m^2+n^2-l^2)(2n\frac{\partial m}{\partial \boldsymbol{a}} + 2m\frac{\partial n}{\partial \boldsymbol{a}})}{4m^2n^2} \tag{5}$$

Note that, in rare cases, two nodes with identical angles may comprise different shapes. One example is shown in Fig. 4, where two nodes contain the same first five neighboring vertices and differ in the sixth. Their corresponding angles are identical, but their shapes are not congruent. In such cases, the proposed optimization strategy becomes ineffective as it cannot similarize the node shapes. However, it should be noted that such undesired cases can be avoided in the global clustering–optimization framework. Due to the different shapes, the nodes are often pre-divided into different groups. Then, the optimization only fine-tunes the nodes that are already similar. Eventually, congruent nodes identified in the algorithm lie in the same group, contain almost identical angles, and the similarity metric between them also becomes nearly zero.

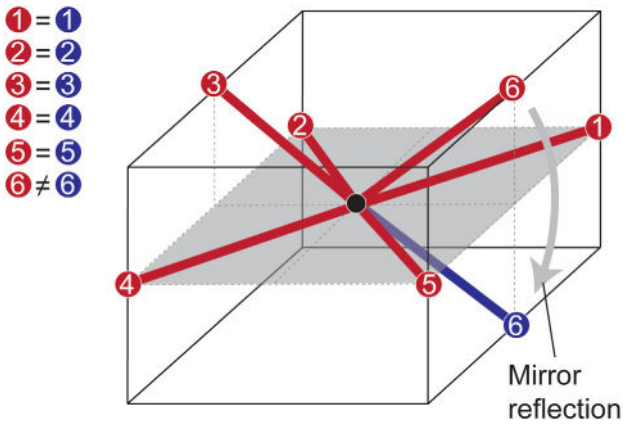


Fig. 4. Two nodes share the same first five neighboring vertices and differ in the sixth. Specifically, the first, second, fourth, and fifth vertices are on the same plane, and those two sixth vertices are mirror images of each other with respect to this plane. Hence, the corresponding angles of these two nodes are identical. However, their shapes are not congruent.

2.2.2. Preserving the overall shape

For architectural applications, the overall shape of the optimized geometry is often expected to closely approximate the given target surface. This can be achieved by constraining all vertices to stay close to the target surface, boundary vertices to align with the boundary curves, and corner vertices to be fixed [23]. In our algorithm, the corresponding shape term F_s is formulated as:

$$F_s = \omega_{sur}\sum_{i=1}^{V} {d_{si}}^2 + \omega_{bou}\sum_{j=1}^{V_{bou}} {d_{bj}}^2 + \omega_{cor}\sum_{k=1}^{V_{cor}} {d_{ck}}^2 \tag{6}$$

where d_{si}, d_{bj}, and d_{ck} are the closest distances, respectively, from the ith vertex to the target surface, from the jth boundary vertex to the boundary curve, and from the kth corner vertex to its initial position. ω_{sur}, ω_{bou}, and ω_{cor} are the corresponding weights for each term. V, V_{bou}, and V_{cor} are the corresponding vertex numbers. The calculation of the gradient of F_s is straightforward, hence neglected.

2.2.3. Global objective function

Combining all the sub-goal terms, the global objective function can be formulated as:

$$F = \omega_c F_c + \omega_s F_s \tag{7}$$

where ω_c and ω_s are the weights for the congruence and shape terms, respectively. Essentially, this is a multi-objective optimization problem, with the positions of vertices treated as design variables. Additional geometric goals (or constraints) can be formulated as sub-goal terms and incorporated into the objective function. By manipulating the weight values, users can control the bias toward each sub-goal. Minimizing F will lead to a final outcome that best meets the user-defined objectives. However, when the applied geometric goals are in conflict, the obtained result is typically a comprising solution with each sub-goal only partially achieved.

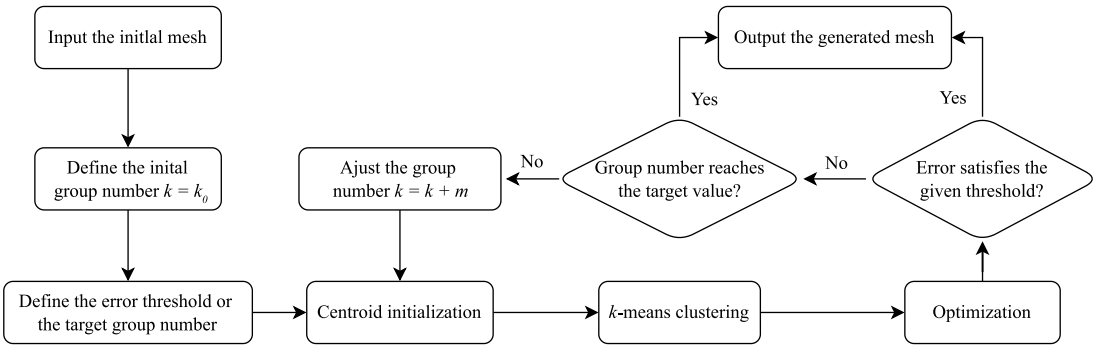


Fig. 5. The computational workflow of the global clustering–optimization framework.

3. Numerical analysis and discussion

3.1. Implementation details

By assembling the proposed clustering and optimization techniques, we propose a computational framework to reduce the number of different nodes in space frame structures, as shown in Fig. 5. Given an input geometry with a specified initial group number k_0, the algorithm begins the first cycle of clustering and optimization. With the obtained new geometry, the error, which could be any user-defined metric, can be updated accordingly. If the current error satisfies the given threshold, or if the current group number k reaches the specified target group number K, the algorithm will stop and output the final geometry. Otherwise, the group number is increased by m, followed by a new round of clustering and optimization, until any stop criterion is triggered. Eventually, the algorithm can either find the minimum group number that satisfies a given error threshold or determine the errors that correspond to a specified target group number.

A variety of numerical examples are presented in the following sections, as shown in Figs. 6–10. The maximum angle difference, σ_c, is used as the error metric in this study, which assesses the largest value among all maximum deviation angles (see Fig. 1) between each node and its corresponding centroid. This metric reflects the similarity between the nodes within the same group, which has a clear physical meaning and is therefore intuitive for real-world fabrication. Another error metric, σ_s, is adopted to evaluate the maximum distance from each vertex to the target surface. This metric shows the extent of the optimized geometry approximating the target surface. Additionally, for fair comparison, all models are uniformly scaled such that the maximum edge of the corresponding bounding box is of unit length. Different optimization terms in the global objective function are normalized to make their values comparable. All optimization problems are solved using the nonlinear conjugate gradient method [40]. The global framework has been scripted as a plugin using C# codes in the Rhino-Grasshopper CAD platform. All numerical tests are carried out on an ordinary personal computer with an i7-7700HQ Intel core and 8 GB of memory.

3.2. Method validation

Three benchmark problems are tested to demonstrate the effectiveness of the proposed method for achieving congruent nodes, as shown in Fig. 6. Given three geometries with all nodes being different, the goal is to optimize their shapes such that only one type of node is required per case ($\omega_c = 1$, $\omega_s = 0$, and $K = 1$). The geometries in (a) and (b) are two-dimensional single layer structures, and the one in (c) is a three-dimensional double layer structure. After optimization, it can be seen that low-valence nodes are divided into high-valence groups, and the node shapes become nearly congruent, with σ_c being extremely low for all cases. Specifically, all quads in (a) are transformed into rectangles with 90° internal angles. All triangles in (b) are optimized into right triangles with internal angles of 60°. In (c), the nodes are translated onto the same plane for both layers, and all internal angles become 90°. Besides, each node in the bottom layer intersects with the top layer at the face center by moving it along the face normal direction. These benchmark problems well demonstrate the effectiveness of the proposed optimization strategy to achieve congruent nodes and its capacity for solving space frame structures of multiple layers.

3.3. Comparison of different clustering methods

Several general clustering methods are compared to understand their applicability to the specific problem of node-clustering defined in this study. Here, the comparison is based on the form of an ordinary doubly-curved surface, as shown in Fig. 7. The methods (1)-(5) are all based on standard k-means clustering while adopting different centroid initialization strategies, as detailed in Fig. 7. The nodes of the input geometry are divided into 10, 20, 30, 40, 50, 60, and 70 groups, using different methods, respectively. For each case, the geometry is further optimized to achieve congruent nodes while preserving its overall shape ($\omega_c = \omega_s = 1$).

Several interesting outcomes can be observed from Fig. 7. Firstly, only by clustering, both methods (2) and (3) mostly produce better results than (1). This indicates the importance of carefully choosing the initial centroids and also demonstrates the effectiveness of the adopted initialization strategies. Besides, it can be seen that (3) is generally better than (2), which suggests that the randomness introduced in (2) may not improve the result in the specific problem of node-clustering. Secondly, the clustering results of (4) and (5) are slightly better than those of (2) and (3), respectively. This is because different initial seeds are examined and the best result is chosen as the final output in (4) and (5). Such a method can somewhat decrease the error but requires multiple repetitions of the whole process. Therefore, it is effective for small-scale problems where a single-run is sufficiently quick, but generally not suitable for large-scale problems due to its lack of efficiency. Finally, by including optimization, the errors are further decreased, which again validates the effectiveness of the optimization process. However, it can be seen that a clustering configuration with lower error may not guarantee a better optimization result. This is because the current clustering method is only based on node shapes without considering the subsequent optimization. How to determine the clustering configuration such that the best optimization result can be obtained is rather challenging and requires further research. In the following numerical examples presented in Sections 3.4 and 4, we use the farthest point sampling strategy for centroid initialization, since it generally produces good optimization outcomes (see Fig. 7). To reduce the runtime, we randomly select one node as the starting centroid without examining multiple initial seeds. For specific practical applications, since the relation between clustering and optimization outcome remains uncertain, it is recommended to experiment with different clustering methods and select the best result as the final outcome.

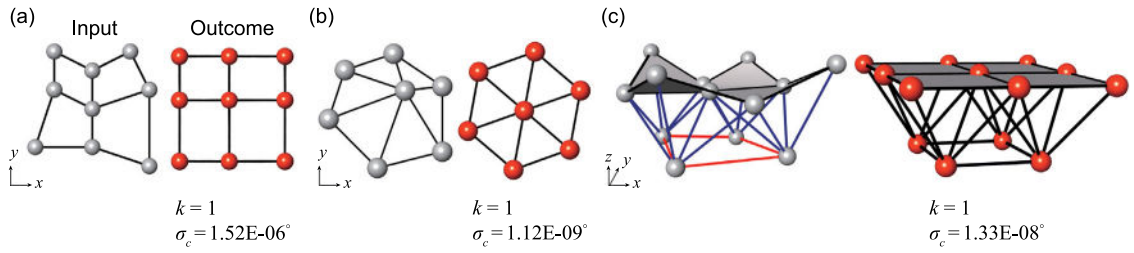


Fig. 6. Validation of the proposed optimization strategy for achieving congruent nodes. (a) and (b) are two-dimensional single layer structures, and (c) is three-dimensional double layer structure. Different nodes are optimized into congruent shapes per geometry.

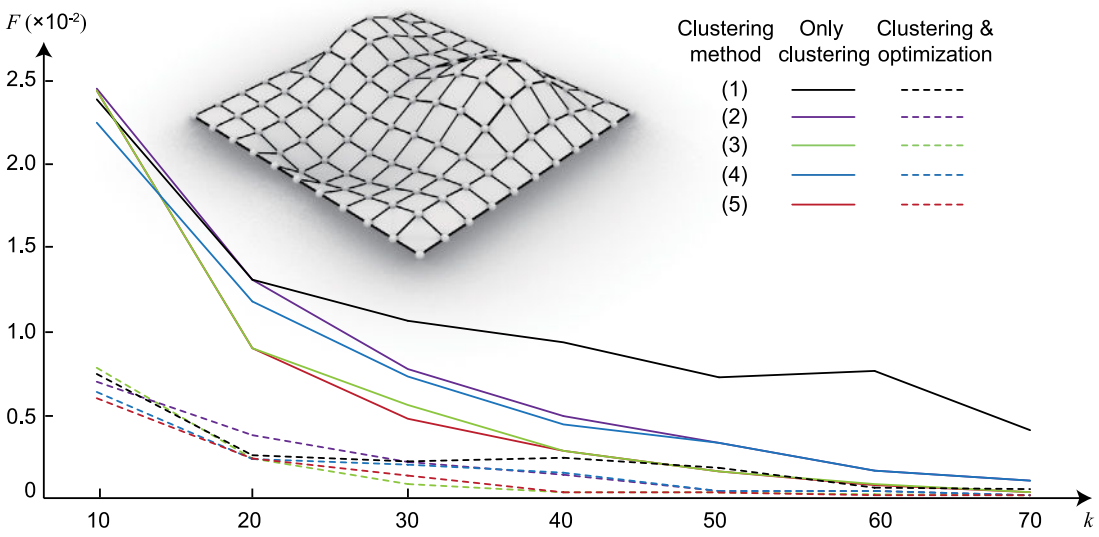


Fig. 7. Comparison of different clustering methods for the specific node-clustering problem based on an ordinary doubly-curved surface. F denotes the value of the objective function. The details of different clustering methods are as follows: (1) The standard k-means clustering method [31], with initial centroids chosen at random. (2) k-means++ [34] to determine the initial centroids, followed by k-means clustering. (3) Farthest point sampling [35] to determine the initial centroids, followed by k-means clustering. (4) k-means++ to determine the initial centroids, followed by k-means clustering; repeating the process five times with different starting centroids, with the best result chosen as the final output. (5) Farthest point sampling to determine the initial centroids, followed by k-means clustering; repeating the process five times with different starting centroids, with the best result chosen as the final output.

3.4. Parametric investigation

Aside from the clustering method, the term F_s for preserving the overall shape, the initial group number k_0, and the step length of the group number m, do all have an effect on the final outcome. Here, we conduct a series of independent tests with different input parameters to investigate their corresponding effects on the final outcome. Specifically, the investigation is based on an ordinary doubly-curved surface with 256 nodes, as shown in Fig. 8(a). The parameter settings and results of all numerical examples are summarized in Table 1.

3.4.1. The effect of preserving the overall shape

For the input geometry, to satisfy the given error threshold only by clustering ($\omega_c = \omega_s = 0$), the required group number k is 99, as shown in Fig. 8(b). The corresponding variation of the objective function F with respect to k is depicted in Fig. 8(c). Then, we include the proposed optimization term F_c ($\omega_c = 1$) for achieving congruent nodes to improve the results. It is found that the required group number can be significantly reduced to 1, as shown in Fig. 8(d). However, the overall shape is largely modified; the original free-form geometry is deformed into a flattened surface. For architectural applications, this is not acceptable as the target shape is severely violated. Therefore, we further include the optimization term F_s ($\omega_s = 1$) to preserve the overall shape of the input geometry. The new results are presented in Fig. 8(e), where the overall shape is successfully preserved, and k is increased to 34 due to the extra constraints. However, a new problem is noticed in the obtained geometry: the mesh pattern is largely distorted.

Table 1
Data for the numerical tests using the global clustering–optimization framework. N is the number of nodes. ω_c and ω_s are the optimization weight values for achieving congruent nodes and preserving the overall shape, respectively. In the global iteration, k_0 is the starting group number, m is the step length of the group number, and K is the target group number. For the output geometry, k is the final group number, σ_c denotes the maximum deviation angle between nodes and their centroids, and σ_s denotes the maximum deviation distance from nodes to the target surface.

Figure	N	ω_c	ω_s	k_0	m	Stop criterion	k	σ_c (°)	σ_s	Runtime (min)
8(b)	256	0	0	1	1	$\sigma_c < 1°$	99	0.988	0	9.8
8(d)	256	1	0	1	1	$\sigma_c < 1°$	1	0.999	0.202	0.1
8(e)	256	1	1	1	1	$\sigma_c < 1°$	34	0.890	0.012	2.5
8(f)	256	1	1	1	1	$K = 1$	1	8.070	0.022	0.1
8(g)	256	1	1	20	1	$\sigma_c < 1°$	30	0.995	0.004	1.0
8(h)	256	1	1	20	5	$K = 30$	30	1.100	0.004	0.3
8(i)	256	1	1	20	10	$K = 30$	30	1.517	0.004	0.2
9	1535	1	1	40	5	$\sigma_c < 3°$	75	2.882	0.007	8.1
10	3006	1	1	90	20	$\sigma_c < 3°$	410	3.000	0.010	150.0

3.4.2. The effect of the initial group number k_0

The reason for the severe distortion in the mesh pattern is the value of the initial group number k_0 being too small. A small k_0 would introduce overly strong constraints during optimization, which often lead to a large deformation in the vertices, hence causing a distorted mesh pattern. For example, $k_0 = 1$ requires all the nodes to be congruent. The generated geometry after trying to merge all nodes into congruent shapes is shown in Fig. 8(f), where the mesh pattern is

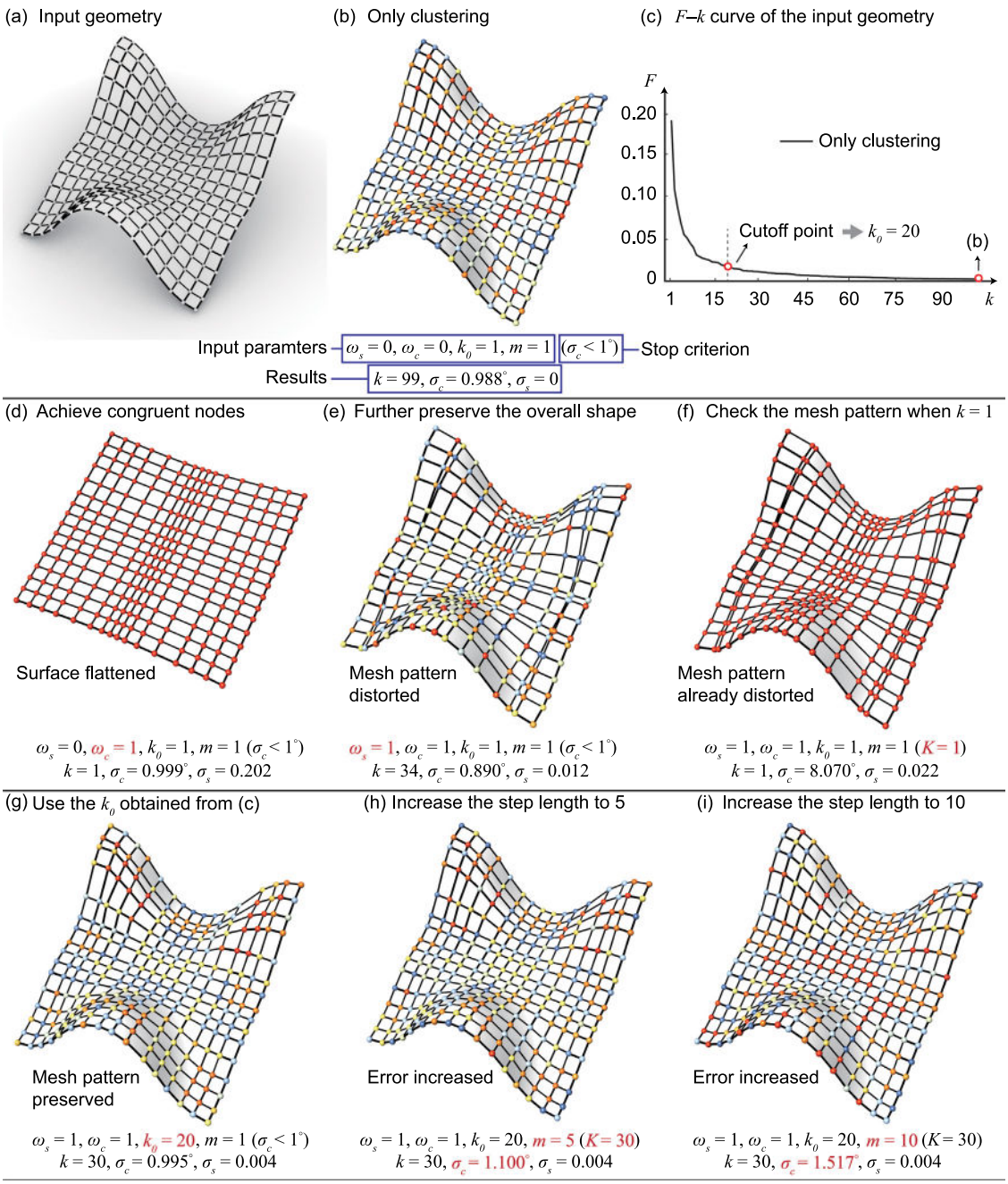


Fig. 8. Investigating the effects of different parameters on the final outcome. (a) The input geometry. (b) The obtained results only by clustering. (c) The variation of the objective function F with respect to the group number k for the input geometry through only clustering. (d) The obtained results after including the optimization term for achieving congruent node ($\omega_c = 1$), where the obtained geometry is deformed into a flattened surface. (e) The obtained results by further adding the optimization term for preserving the overall shape ($\omega_s = 1$), where the overall shape is preserved while the mesh pattern is largely distorted. (f) The results at the early stage ($k = 1$) of the global iteration, showing that the mesh pattern is already distorted after the first round of clustering and optimization. (g) Using an appropriate $k_0 = 20$, as determined in (c), leads to a solution where the mesh pattern is well preserved. (h) Compared to (g), increasing m to 5 and stopping at the same group number ($k = 30$) leads to a higher error σ_c. (i) Compared to (h), increasing m to 10 and stopping at the same group number ($k = 30$) leads to a higher error σ_c.

already distorted. The following clustering and optimization iterations are continued based on the current geometry, which cannot recover the pattern to the initial state.

To alleviate the distortion in the mesh pattern, choosing an appropriate k_0 is rather important. Here, the goal is to find a k_0 that can capture the main features of the input mesh. In that case, the optimization only fine-tunes the geometry with the vertices merely undergoing slight movement, hence the initial pattern can be mainly preserved. In this study, the *elbow method* [41] is used to determine the k_0, which is chosen as the *cutoff point* of the F–k curve with respect to the input geometry (Fig. 8(c)). The idea is that the first clusters will add much information about the mesh pattern, so increasing k will significantly reduce F; however, once k exceeds the necessary number (i.e., the cutoff point), the added information will drop sharply because it is only adding the details, and the F–k curve will flatten out. Specifically, the cutoff point is defined as the point after which F starts decreasing in a nearly linear fashion. In the case of Fig. 8(c), the cutoff point is calculated as $k = 20$. The resultant geometry with $k_0 = 20$ is shown in Fig. 8(g), where it can be seen the mesh pattern is successfully preserved.

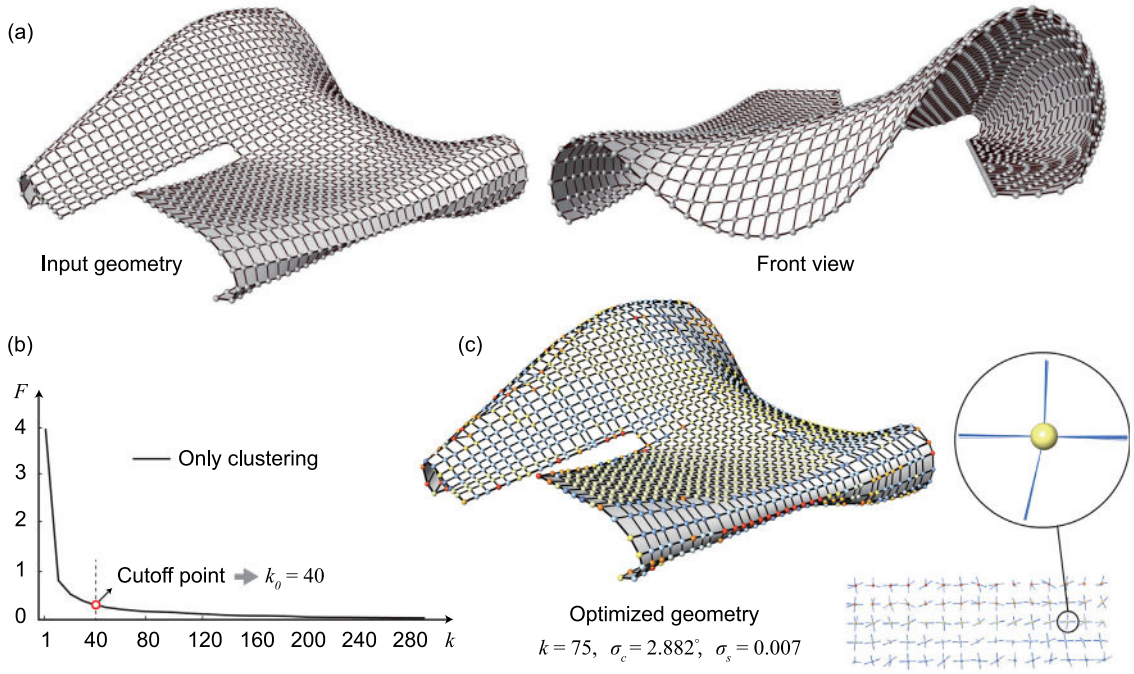


Fig. 9. Results of the single layer structure. (a) The input geometry with 1535 nodes. (b) The variation of the objective function F with respect to the group number k for the input geometry through only clustering. (c) The optimized geometry with $k = 75$, $\sigma_c = 2.882°$, and $\sigma_s = 0.007$. All nodes are superimposed per group on the bottom.

3.4.3. The effect of the step length of the group number m

Lastly, we investigate the effect of the step length m on the final results. In Fig. 8(g), k is increased from 20 to 30 with $m = 1$. Here, we conduct two additional tests with $m = 5$ and 10, respectively. Fig. 8(h) and (i) show the corresponding results, which indicate that a smaller m tends to produce a lower error. This is because, for the highly nonlinear optimization problem in this study, lots of local minimum exist, and the gradient-based algorithm can very easily become stuck at one of them. Gradually increasing the group number with a small m allows the algorithm to proceed further every time a local minimum is reached. Eventually, this may generate a final result that is closer to the global optimum. However, a smaller m typically leads to more cycles of clustering and optimization, hence costing longer runtime. The consideration of runtime is also important when determining the value of m, particularly for large-scale problems.

4. Case study: Heydar Aliyev center

We demonstrate the potential practical application of the proposed method by applying it to re-design a complex, free-form architectural project—the Heydar Aliyev Museum (designed by Zaha Hadid Architects). The detailed parameter settings and results are summarized in Table 1. The input geometry is a recreated quad-dominant mesh [28,42,43], which is uniformly scaled beforehand such that the maximum edge of its bounding box is of unit length. Both single layer and double layer cases are considered in this study. The single layer structure is directly represented by the input mesh, as shown in Fig. 9(a). The double layer structure is created by offsetting the vertices of the dual mesh along the face normal directions by 0.02, and then connecting the corresponding vertices, as shown in Fig. 10(a). m is set as 5 and 20 rather than 1 for the single layer and double layer structures, respectively, to accelerate the overall process. k_0 is determined using the elbow method. We set $\omega_c = \omega_s = 1$ to achieve congruent nodes while preserving the overall shape. It should be noted that for the double layer structure, only the top layer vertices are constrained by F_s. The error threshold is exemplarily set as $\sigma_c < 3°$ for both cases.

The obtained results for the single and double layer structures are shown in Figs. 9 and 10, respectively. For the single layer case, the input geometry (Fig. 9(a)) contains 1535 nodes, with valence ranging from three to five. We set $k_0 = 40$ based on the F–k curve (Fig. 9(b)) obtained by only clustering the nodes of the input geometry. Fig. 9(c) depicts the optimized final geometry, which requires only 75 different groups of nodes (4.9% of the total node number) to achieve $\sigma_c < 3°$. The overall runtime is 8.1 min. The maximum deviation σ_s is 0.007. For the double layer case, the input geometry (Fig. 10(a)) contains 3006 nodes, with 1535 nodes in the top layer and 1471 nodes in the bottom layer. The node valence ranges from three to ten, which is much more complex than the single layer case. k_0 is set to 90 based on the F–k curve shown in Fig. 10(b). The final geometry obtained is shown in Fig. 10(c), with only 410 different groups of nodes (13.6% of the total node number) required for $\sigma_c < 3°$. The whole iteration costs 150 min. The maximum deviation σ_s is 0.010.

In both single layer and double layer cases, the numbers of different nodes are significantly reduced. The mesh patterns are slightly modified. The generated geometries closely approximate the target surface with small deviations. In practical applications, the maximum deviation σ_s may be constrained by certain thresholds. Users can adjust the corresponding weight ω_s to manipulate the value of σ_s until satisfying the given requirements. Overall, the obtained results well demonstrate the capacity of our method for solving complex, large-scale, free-form frame structures. However, for cases with a large number of nodes, our algorithm is not very fast. For the double layer case, 150 min are required to finish the whole iteration process. The runtime could be reduced by increasing the step length m, but this generally results in more groups of nodes. Another option is to improve the efficiency of the clustering strategy, but this requires further research. Additionally, the current algorithm focuses on the geometric aspect, specifically the shape variety of nodes, without considering the overall structural performance. On the one hand, since only geometric considerations are involved, the proposed method is equally effective for space frame and space truss structures. On the other hand, although the positions of nodes are only fine-tuned with the mesh topology remaining unchanged, the specific influence of the node movements on the structural performance is uncertain. For the future research, we intend to incorporate the sensitivity analysis of structural performance with respect to the node movements [44] into the global computational framework to achieve fabrication-aware structural designs.

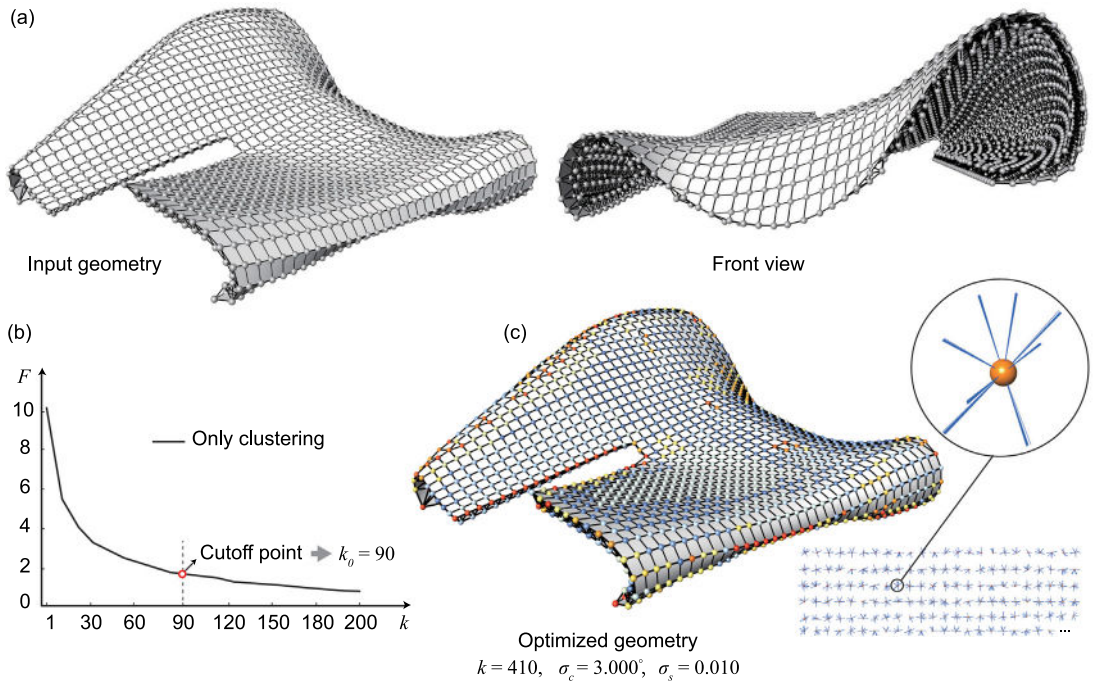


Fig. 10. Results of the double layer structure. (a) The input geometry with 3006 nodes. (b) The variation of the objective function F with respect to the group number k for the input geometry through only clustering. (c) The optimized geometry with $k = 410$, $\sigma_c = 3.000°$, and $\sigma_s = 0.010$. A part of nodes are superimposed per group on the bottom.

5. Conclusion

This study presents a clustering–optimization framework to reduce the number of different nodes in space frame structures. First, an evaluation metric is proposed to quantify the similarity between different nodes, which contains a solid mathematical basis that ensures the minimization of the sum of squared distances between corresponding vertex pairs. Based on the proposed similarity metric, we adapt the k-means clustering method to partition nodes into a specific number of groups of similar shapes. By comparing five different centroid initialization strategies, we find that both farthest point sampling and k-means++ can improve clustering results, and farthest point sampling tends to yield better outcomes. Also, it is observed that the starting centroid has a minor effect on the clustering results for both initialization methods.

Furthermore, an effective optimization strategy is developed that can transform geometrically different nodes into near-congruent shapes by equalizing corresponding angles, as validated through three benchmark tests. This strategy leads to a well-formulated objective function with attainable gradient information, which enables efficient optimization using gradient-based methods, making it feasible to solve large-scale practical problems. An additional geometric goal is included to better approximate the target surface for architectural applications.

By interleaving clustering and optimization, we propose a computational framework to reduce the number of different nodes in free-form space frame structures. The effects of input parameters on the final outcome are investigated, with corresponding suggestions provided on the parameter selection. To demonstrate potential practical applications of this method, a case study based on the Heydar Aliyev Center, a complex architectural project, is presented. Both single and double layer structures are considered, and the final group number is reduced to 4.9% and 13.6% of the total node number, respectively, while ensuring the maximum deviation angle between nodes and corresponding centroids below 3°.

The proposed clustering–optimization framework is generic and can be applied to a wide range of space frames composed of single, double, or multiple layers of grids, regardless of mesh topology. By adjusting the error threshold, our method can be applied to various practical projects with different angle tolerances. Overall, the proposed framework provides a flexible way for designing free-form space frame structures to achieve cost-effective solutions with node-fabrication considerations.

Replication of results

The results of the optimized designs and the basic code of this work are available from the corresponding author on reasonable request.

CRediT authorship contribution statement

Yuanpeng Liu: Investigation, Methodology, Software, Validation, Visualization, Writing – original draft. **Ting-Uei Lee:** Methodology, Writing – review & editing. **Antiopi Koronaki:** Methodology, Writing – review & editing. **Nico Pietroni:** Methodology, Writing – review & editing. **Yi Min Xie:** Supervision, Conceptualization, Methodology, Writing – review & editing.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the Australian Research Council (FL190100014 and DP200102190).

References

[1] Lan TT. Space frame structures. In: Structural engineering handbook. CRC Press LLC, Boca Raton; 1999.

[2] Austern G, Capeluto IG, Grobman YJ. Rationalization methods in computer aided fabrication: A critical review. Autom Constr 2018;90:281–93. http://dx.doi.org/10.1016/j.autcon.2017.12.027.

[3] Fujita S, Ohsaki M. Shape optimization of free-form shells using invariants of parametric surface. Int J Space Struct 2010;25(3):143–57. http://dx.doi.org/10.1260/0266-3511.25.3.143.

[4] Wang H, Pellis D, Rist F, Pottmann H, Müller C. Discrete geodesic parallel coordinates. ACM Trans Graph 2019;38(6):1–13. http://dx.doi.org/10.1145/3355089.3356541.

[5] Pottmann H, Asperl A, Hofer M, Kilian A. Architectural geometry. Exton, Pennsylvania: Bentley Institute Press; 2007.

[6] Pottmann H, Eigensatz M, Vaxman A, Wallner J. Architectural geometry. Comput Graph 2015;47:145–64. http://dx.doi.org/10.1016/j.cag.2014.11.002.

[7] Botsch M, Kobbelt L, Pauly M, Alliez P, Lévy B. Polygon mesh processing. New York: CRC Press; 2010.

[8] Liu Y, Pottmann H, Wallner J, Yang Y-L, Wang W. Geometric modeling with conical meshes and developable surfaces. ACM Trans Graph 2006;25(3):681–9. http://dx.doi.org/10.1145/1141911.1141941.

[9] Pottmann H, Liu Y, Wallner J, Bobenko A, Wang W. Geometry of multi-layer freeform structures for architecture. ACM Trans Graph 2007;26(3):65–es. http://dx.doi.org/10.1145/1276377.1276458.

[10] Liu Y, Xu W, Wang J, Zhu L, Guo B, Chen F, Wang G. General planar quadrilateral mesh design using conjugate direction field. ACM Trans Graph 2011;30(6):1–10. http://dx.doi.org/10.1145/2070781.2024174.

[11] Zimmer H, Campen M, Herkrath R, Kobbelt L. Variational tangent plane intersection for planar polygonal meshing. In: Advances in architectural geometry 2012. Paris, France: Springer, Vienna; 2012, p. 319–32. http://dx.doi.org/10.1007/978-3-7091-1251-9_26.

[12] Li Y, Liu Y, Wang W. Planar hexagonal meshing for architecture. IEEE Trans Vis Comput Graphics 2015;21(1):95–106. http://dx.doi.org/10.1109/TVCG.2014.2322367.

[13] Pluta K, Edelstein M, Vaxman A, Ben-Chen M. PH-CPF: Planar hexagonal meshing using coordinate power fields. ACM Trans Graph 2021;40(4):1–19. http://dx.doi.org/10.1145/3450626.3459770.

[14] Jiang C, Tang C, Vaxman A, Wonka P, Pottmann H. Polyhedral patterns. ACM Trans Graph 2015;34(6):1–12. http://dx.doi.org/10.1145/2816795.2818077.

[15] Schling E. Repetitive structures (Dissertation), Müchen: Technische Universität Müchen; 2018, http://dx.doi.org/10.14459/2018md1449869.

[16] Lee T-U, Liu Y, Xie YM. Dividing a sphere hierarchically into a large number of spherical pentagons using equal area or equal length optimization. Comput Aided Des 2022;148:103259. http://dx.doi.org/10.1016/j.cad.2022.103259.

[17] Liu Y, Lee T-U, Rezaee Javan A, Xie YM. Extending Goldberg's method to parametrize and control the geometry of Goldberg polyhedra. R Soc Open Sci 2022;9(8):220675. http://dx.doi.org/10.1098/rsos.220675.

[18] Brütting J, Senatore G, Fivet C. Design and fabrication of a reusable kit of parts for diverse structures. Autom Constr 2021;125:103614. http://dx.doi.org/10.1016/j.autcon.2021.103614.

[19] Lu H, Xie YM. Reducing the number of different members in truss layout optimization. Struct Multidiscip Optim 2023;66(3):52. http://dx.doi.org/10.1007/s00158-023-03514-y.

[20] Huard M, Eigensatz M, Bompas P. Planar panelization with extreme repetition. In: Advances in architectural geometry 2014, September 2014, London, England. Cham, Switzerland: Springer; 2015, p. 259–79. http://dx.doi.org/10.1007/978-3-319-11418-7_17.

[21] Jiang C, Tang C, Tomičí M, Wallner J, Pottmann H. Interactive modeling of architectural freeform structures: Combining geometry with fabrication and statics. In: Advances in architectural geometry 2014, September 2014, London, England. Cham, Switzerland: Springer; 2015, p. 95–108. http://dx.doi.org/10.1007/978-3-319-11418-7_7.

[22] Eigensatz M, Kilian M, Schiftner A, Mitra NJ, Pottmann H, Pauly M. Paneling architectural freeform surfaces. ACM Trans Graph 2010;29(4):1–10. http://dx.doi.org/10.1145/1778765.1778782.

[23] Liu Y, Lee T-U, Rezaee Javan A, Pietroni N, Xie YM. Reducing the number of different faces in free-form surface tessellations. Autom Constr 2023. submitted for publication.

[24] Fu C-W, Lai C-F, He Y, Cohen-Or D. K-set tilable surfaces. ACM Trans Graph 2010;29(4):1–6. http://dx.doi.org/10.1145/1778765.1778781.

[25] Singh M, Schaefer S. Triangle surfaces with discrete equivalence classes. ACM Trans Graph 2010;29(4):1–7. http://dx.doi.org/10.1145/1778765.1778783.

[26] Liu Z, Zhang Z, Zhang D, Ye C, Liu L, Fu X-M. Modeling and fabrication with specified discrete equivalence classes. ACM Trans Graph 2021;40(4):1–12. http://dx.doi.org/10.1145/3450626.3459843.

[27] Zimmer H, Lafarge F, Alliez P, Kobbelt L. Zometool shape approximation. Graph Models 2014;76(5):390–401. http://dx.doi.org/10.1016/j.gmod.2014.03.009.

[28] Koronaki A, Shepherd P, Evernden M. Rationalization of freeform space-frame structures: Reducing variability in the joints. Int J Autom Comput 2020;18(1):84–99. http://dx.doi.org/10.1177/1478077119894881.

[29] Ramaswamy GS, Eekhout M, R. SG. Analysis, design and construction of steel space frames. Thomas Telford; 2002.

[30] Groover MP. Automation, production systems, and computer-integrated manufacturing. Pearson Education India; 2016.

[31] Lloyd S. Least squares quantization in PCM. IEEE Trans Inform Theory 1982;28(2):129–37. http://dx.doi.org/10.1109/TIT.1982.1056489.

[32] Hennig C, Meila M, Murtagh F, Rocci R. Handbook of cluster analysis. CRC Press; 2015.

[33] Sorkine-Hornung O, Rabinovich M. Least-squares rigid motion using svd. Computing 2017;1(1):1–5.

[34] Arthur D, Vassilvitskii S. k-means++: the advantages of careful seeding. Technical Report 2006–13, Stanford InfoLab; 2006, http://ilpubs.stanford.edu:8090/778/.

[35] Eldar Y, Lindenbaum M, Porat M, Zeevi YY. The farthest point strategy for progressive image sampling. IEEE Trans Image Process 1997;6(9):1305–15. http://dx.doi.org/10.1109/83.623193.

[36] Lipowski A, Lipowska D. Roulette-wheel selection via stochastic acceptance. Phys A 2012;391(6):2193–6. http://dx.doi.org/10.1016/j.physa.2011.12.004.

[37] Hare W, Nutini J, Tesfamariam S. A survey of non-gradient optimization methods in structural engineering. Adv Eng Softw 2013;59:19–28. http://dx.doi.org/10.1016/j.advengsoft.2013.03.001.

[38] Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search. Acta Mech 2010;213(3–4):267–89. http://dx.doi.org/10.1007/s00707-009-0270-4.

[39] Maxwell JC. L. on the calculation of the equilibrium and stiffness of frames. Lond Edinb Dublin Philos Mag 1864;27(182):294–9. http://dx.doi.org/10.1080/14786446408643668.

[40] Dai Y-H, Yuan Y. A nonlinear conjugate gradient method with a strong global convergence property. SIAM J Optim 1999;10(1):177–82. http://dx.doi.org/10.1137/S1052623497318992.

[41] Thorndike RL. Who belongs in the family. In: Psychometrika, Vol. 18. Citeseer; 1953, p. 267–76. http://dx.doi.org/10.1007/BF02289263.

[42] Sanchez-Alvarez J. Practical aspects determining the modelling of the space structure for the free-form envelope enclosing Baku's Heydar Aliyev Cultural Centre. In: Symposium of the international association for shell and spatial structures, 50th. Valencia. Editorial Universitat Politècnica de València; 2010, p. 1263–74, http://hdl.handle.net/10251/7071.

[43] Winterstetter T, Alkan M, Berger R, Watanabe M, Toth A, Sobek W. Engineering complex geometries–the Heydar Aliyev centre in Baku. Steel Constr 2015;8(1):65–71. http://dx.doi.org/10.1002/stco.201520005.

[44] Wang H, Chen Z, Wen G, Ji G, Xie YM. A robust node-shifting method for shape optimization of irregular gridshell structures. Structures 2021;34:666–77. http://dx.doi.org/10.1016/j.istruc.2021.08.003.