Similarity-driven topology finding of surface patterns for structural design

Oval, R.; Mesnil, R.; Van Mele, T.; Baverel, O.; Block, P.

DOI
10.1016/j.cad.2024.103751

Publication date
2024

Document Version
Final published version

Published in
CAD Computer Aided Design

Citation (APA)
Oval, R., Mesnil, R., Van Mele, T., Baverel, O., & Block, P. (2024). Similarity-driven topology finding of surface patterns for structural design. *CAD Computer Aided Design*, *176*, Article 103751. https://doi.org/10.1016/j.cad.2024.103751

Computer-Aided Design 176 (2024) 103751

Contents lists available at ScienceDirect

Computer-Aided Design

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

Research Paper

Similarity-driven topology finding of surface patterns for structural design

R. Oval [a,b,c,*], R. Mesnil [b], T. Van Mele [c], O. Baverel [b], P. Block [c]

[a] *Delft University of Technology, Faculty of Civil Engineering & Geosciences, Delft, The Netherlands*
[b] *Laboratoire Navier, UMR 8205, École des Ponts ParisTech, Université Gustave Eiffel, CNRS, Champs-sur-Marne, France*
[c] *ETH Zurich, Institute of Technology in Architecture, Block Research Group, Zurich, Switzerland*

ARTICLE INFO

Keywords:
Architectural geometry
Structural topology
Computational design
Generative design
Multi-objective design
Surface structures
Discrete structures

ABSTRACT

Structural design is a search for the best trade-off between multiple architecture, engineering, and construction objectives, not only mechanical efficiency or construction rationality. Producing hybrid designs from single-objective optimal designs to explore multi-objective trade-offs is common in the design of structural forms, constrained to a single parametric design space. However, producing topological hybrids offers a more complex challenge, as a combinatorial problem that is not encoded as a finite set of real numbers but as an unbonded series of grammar rules. This paper presents a strategy for the generation of hybrid designs of quad-mesh pattern topologies for surface structures. Based on a quad-mesh grammar, an algebra is introduced to measure the distance between designs, find their similar features, and enumerate designs with different degrees of topological similarity. Structural design applications are shown to highlight the use of topologically hybrid designs as a surrogate for obtaining multi-objective trade-offs.

1. Introduction

1.1. Problem statement

Structural design is a challenging feat where multiple objectives have to be met, stemming from architecture, engineering, and construction. Beyond single-objective optimisation, structural designers are after a design that offers a trade-off between the different project's requirements, which evolve along the design process. For instance, between a design that is optimal for mechanical efficiency [1] and a design that is optimal for construction rationality [2], hybrid designs that offer performance compromises should be sought for [3]. Fig. 1 illustrates this challenge with the patterns of the waffle and isostatic ribbed floor systems.

Parametric design allows for the production of hybrid designs by combining the continuous-valued parameters that describe its geometry, as in state-of-the-art parametric shape multi-objective optimisation strategies. Multi-objective search in architecture has been applied to the optimisation of building physics [4] and structures [5], using evolutionary [6] and data-driven [7] exploration. Evolutionary search algorithms rely on operations recombining genes of the design's genotype, i.e. the set of design parameters, in the search for hybrid designs to propose different multi-objective trade-offs to the designer, forming a Pareto front. A heuristic suggested by the study of rule-based topology finding with self-organising maps based on design performance is that designs with similar topologies have similar performances [8]. Finding designs with different degrees of topological similarity should offer different degrees of performance similarity for multi-objective design. This assumption relaxes the problem from the downstream performance space to the upstream topological space. However, the generation of topologically hybrid designs is not as clearly defined as for parametrically defined geometries. Indeed, topological exploration requires a different design approach, like grammatical generative design, using a grammar of rules [9–11], resulting in a combinatorial design space that is challenging to explore, let alone for merging design options based on their similarity. In continuum mechanical topology optimisation, similarity has been integrated into solvers to inform the search [12]. Developments include reverse engineering to obtain a specific similarity with regression models [13], Generative Adversarial Networks [14], or Convolutional Neural Networks [15], informed by an input design, potentially drawn [16], using different similarity metrics [17], and domain clustering for modularity [18–20]. However, this approach of topological design does not apply to structural patterns represented by meshes as structured graphs, as opposed to surfaces or bitmaps, which can be computed as lightweight objects independently from density tuning and geometrical exploration.

1.2. Research objectives

This search for topological hybrids is applied to topology finding of structured quad meshes for structural surfaces [21]. Quad meshes

[*] Corresponding author at: Delft University of Technology, Faculty of Civil Engineering & Geosciences, Delft, The Netherlands.
E-mail address: r.oval@tudelft.nl (R. Oval).

https://doi.org/10.1016/j.cad.2024.103751
Received 7 January 2024; Received in revised form 10 June 2024; Accepted 20 June 2024
Available online 24 June 2024

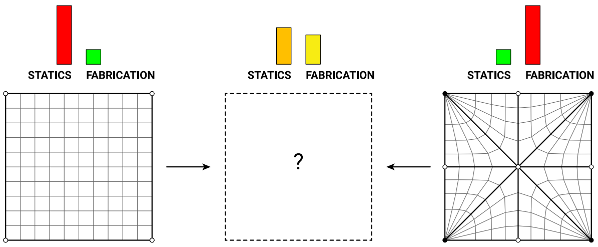
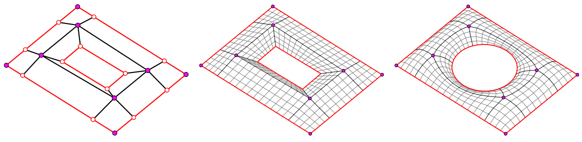


Fig. 1. The challenge of finding hybrid topological pattern designs offering multi-objective trade-offs based on input designs derived for single-objective performance.



(a) Topology (b) Density (c) Geometry

Fig. 2. Approaching exploration of singularities in quad meshes via a coarse quad mesh in black decouples topology from density and geometry. The boundaries are in red, the singularities in pink, and the dense quad mesh in grey.

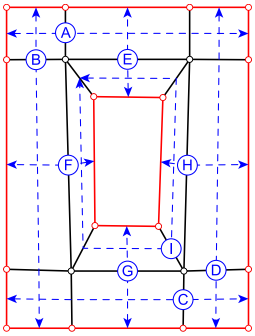


Fig. 3. The strip structure, in blue, completely defines the topology of a quad mesh and its singularities, with nine strips from A to I here. The strip data is collected as the lists of topologically opposite edges across the quad faces.

are natural parameterisations for the design of 2D surface patterns, for shells, gridshells, vaults, or nets. The singularities, the irregular vertices of the pattern, have the most significant influence on the qualitative flow of mesh elements, and therefore on the structural components that they represent. Topological exploration is performed on the design space of singularities, described with coarse quad meshes, independently from later densification and geometrical processing, a common approach in computer graphics, as shown with the workflow in Fig. 2.

Rule-based topology finding allows for the comprehensive exploration of the singularity design space of quad meshes [22]. The grammar is based on a pair of reciprocal rules that modify the topology of the strips in the quad mesh. These strips are formed by a continuous series of quad faces, which fully describe the topology of the quad mesh, as shown in Fig. 3. Strip-based modelling already found applications for digital [23,24] or physical [25,26] approaches, also referred to as loops, rings, or chords.

These strips are modified through polyedges, their dual counterpart, which consist of a continuous series of mesh edges. Through the addition of strips, by inserting it along a polyedge, and deletion of strips, by collapsing them into a polyedge, as the examples shown in Fig. 4, the topology of the quad mesh can be modified, editing its set of singularities and the flow of mesh elements. Other structured patterns can be obtained by applying Conway operators or other transformations by considering the quad mesh as a parameterisation of the surface, not the pattern itself [27–29].

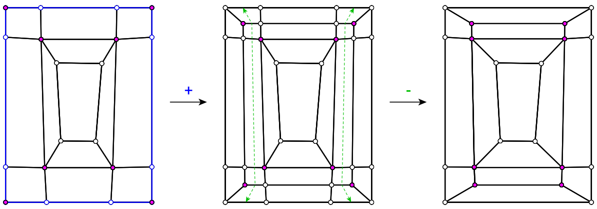


Fig. 4. Adding a strip along the blue polyedge and deleting the two green strips using the two-rule grammar changes the connectivity of the singularities of the coarse mesh.

Through open exploration, the designer can search for the rules that yield topologies with strips that improve performance, however, relying on the designer's experience [30]. Through the generation of hybrid designs based on the strip description of quad meshes and the editing grammar rules, design space exploration can be informed by initial designs, stemming from intuition, experience, or any other exploration and optimisation strategies.

1.3. Contributions

This paper introduces *similarity-driven topology finding*, focusing on quad-mesh patterns for surface structures, as a strategy to produce topological hybrids with different degrees of similarity with input designs, resulting in an algebra with novel mesh operators. Section 2 develops the methodology: clarifies the topological equality between two quad meshes based on their strip structure; defines a topological distance between two quad meshes to assess their similarity; implements an algorithm for computing the distance along with the dissimilar strips to modify; introduces the *intersection submesh* and *union supermesh* objects; and develops a strategy to generate hybrid quad meshes sharing different degrees of similarity with a set of input quad meshes to combine. Section 3 illustrates how to apply this approach to similarity-driven topology finding of designs based on single-objective heuristic or optimal quad meshes to perform multi-objective design.

2. Methodology

In this section, the methodology for producing topological hybrid quad meshes based on input meshes is presented, based on quad-mesh similarity.

2.1. Mesh isomorphism

Mesh equality, in the topological sense, is assessed based on graph isomorphism. The algorithm of [31] for graph isomorphism is used through its implementation in NetworkX [32].

Two graphs are isomorphic if a bijection exists that maps the vertices of one graph to the other while preserving the edge connectivity. Several bijections may exist, which will be leveraged to enrich topological combination (Section 2.3). Fig. 5 illustrates this concept with a graph that is isomorphic to another, through two bijections, while not being isomorphic to another one.

The isomorphism of the meshes is based on the isomorphism of the graphs using the mesh edges. The following approach is limited to manifold meshes with a genus of zero, so that the graphs are planar, i.e. can be embedded in the plane without edge crossing. First, for a given mesh, a graph is built with the edges of the mesh. The face connectivity data from the mesh is integrated into the graph by adding a 'boundary' attribute to graph edges that correspond to boundary mesh edges. Therefore, two meshes are isomorphic if their graphs are isomorphic with matching 'boundary' edge attributes. Fig. 6 illustrates

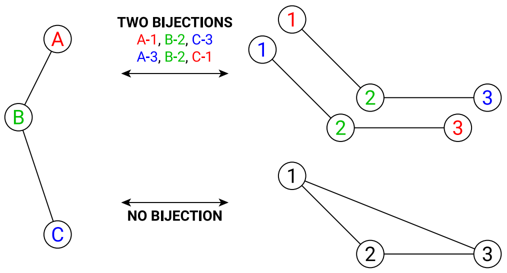


Fig. 5. Two graphs are isomorphic if at least one bijection exists between the vertices that preserves the edge connectivity.

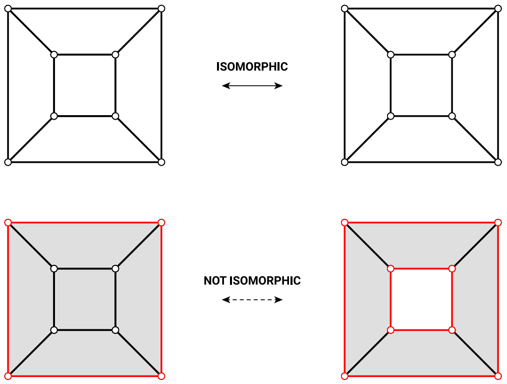


Fig. 6. Two meshes are isomorphic if their edge graphs with 'boundary' edge attributes, in red, are isomorphic.

the necessity to include boundary labels, shown in red, for valid mesh isomorphism through graph isomorphism.

For meshes with pseudo quads, i.e. faces that are geometrically like triangles but topologically like quads with double vertices to mark the pole singularity, the pole vertices are given a 'pole' attribute in the graph that needs to be matched, like the 'boundary' edge attribute.

2.2. Topological similarity

The evaluation of mesh isomorphism, or equality, serves as a basis for defining and computing the topological similarity between two quad meshes, using a distance, or metric.

2.2.1. Definition

The distance defined is not only based on topology (like the set of singularities in the mesh) but also on the grammar of addition and deletion strip rules, which informs how to obtain a mesh from another. The distance is defined as *the minimal number of strips to add and delete to go from one quad mesh to another*. The proposed distance has already been used for finding two-colour quad meshes [33].

A parallel exists with string metrics that measure the distance between two strings as the minimum number of character modifications to apply. The Levenshtein distances uses character insertions, deletions, and substitutions [34], whereas the Hamming distance only uses substitutions [35], for instance. However, the two-rule grammar does not consider direct modifications of the strips, only deletions and additions.

This distance d applies to two elements of the space $E = E_{g,N}$ of quad meshes with the same number of handles g and boundaries N and yields a positive integer in $\mathbb{N}^{+}$: $d : E \rightarrow \mathbb{N}^{+}$. The three distance properties are verified. The distance is symmetric, as the same minimal number of rules apply from a topology A to a topology B and from B to A, as for each addition or deletion strip rule there is a reciprocal one:

$$\forall (A, B) \in E^2, d(A, B) = d(B, A). \tag{1}$$

The distance from a topology to itself is null because no rules need to be applied and if no rules need to be applied then two topologies are the same:

$$\forall (A, B) \in E^2, d(A, B) = 0 \iff A = B. \tag{2}$$

The triangle inequality is satisfied as the same or a lower number of rules have to be applied to go directly from a topology A to a topology C than going through an intermediary topology B:

$$\forall (A, B, C) \in E^3, d(A, C) \leq d(A, B) + d(B, C). \tag{3}$$

Equality occurs if no rules between A and B and between B and C are reciprocal and can compensate each other.

During rule-based exploration, successive application of strip rules generally increases by one the distance and therefore decreases by one the topological similarity with the initial design. Only the application of a reciprocal rule cancelling a previous rule reduces by one the distance. Fig. 7 shows the application of two addition rules and two deletion rules, not reciprocal ones, resulting in a distance of four between the initial and final quad meshes.

Deleting some strips can cause collateral deletion of other strips whose faces are all included in the deleted strips. In Fig. 8, deleting the strip in pink in a four-strip quad mesh results in a two-strip quad mesh at a distance of two due to the secondary deletion of the strip in green. Although one rule is applied, the distance is not one but two. Therefore, the application of a deletion rule counts as the application of multiple deletion rules, equal to one plus the number of collateral deletions.

The distance could be measured on a lighter object like the strip graph of the quad mesh [33] with a modified Graph Edit Distance [36] to include additions and deletions of graph nodes and/or edges. However, the space of strip graphs is not isomorphic to the space of quad meshes, flawing this alternative, which necessitates refinement.

2.2.2. Approach

The computation of the topological distance is based on mesh isomorphism check while trying combinations of an increasing number of strip deletions. Along with the distance, tracking the similar and dissimilar strips will provide the rules to go from one topology to another for topological combination (Section 2.3).

The distance between two quad meshes is defined as the minimum number of strips to add and delete to go from one to another. Searching for combinations of addition and deletion rules to apply on one quad mesh involves infinite combinatorial possibilities for strip additions. However, the number of strip deletions is bounded by 2^n for a quad mesh with n strips. Therefore, the distance results from finding the two minimum sets of strip deletions to apply on the two quad meshes M_1 and M_2 that yield the same submesh $\underline{M}$, up to an isomorphism. In Fig. 9, the deletion of the strips in pink yields two non-isomorphic submeshes $\underline{M'}$ and $\underline{M''}$, as the submesh is not necessarily unique. A submesh represents the largest similar structure of strips in common between the two quad meshes in terms of number of strips. As strip deletion rules are not reciprocal and cannot compensate each other, processing only with deletion rules guarantees having the minimum number of strips to modify. Therefore, finding such a submesh yields the equality case of the triangle inequality:

$$d(M_1, M_2) = d(M_1, \underline{M}) + d(\underline{M}, M_2). \tag{4}$$

Let n_1 and n_2 be the number of strips of the two quad meshes M_1 and M_2, respectively, and supposedly $n_1 \geq n_2$. To find the submesh, an increasing number of strips are deleted on both meshes in parallel. The meshes have a minimum distance:

$$d_{min} = n_1 - n_2, \tag{5}$$

due to a potentially different number of strips. Starting with $k = 0$, all pairs of combinations to delete $k + d_{min}$ strips in mesh M_1 and k strips

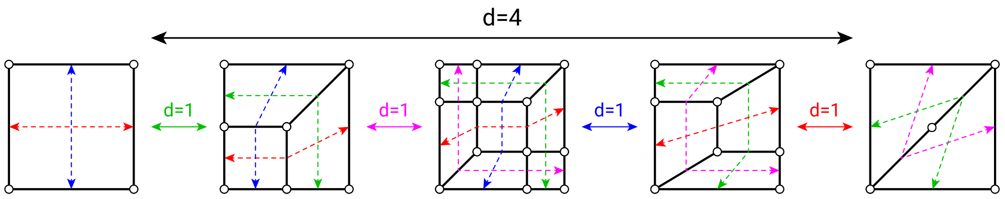


Fig. 7. Applying addition and deletion rules that modify different strips on a quad mesh increases the distance d by one, and decreases the topological similarity by one.

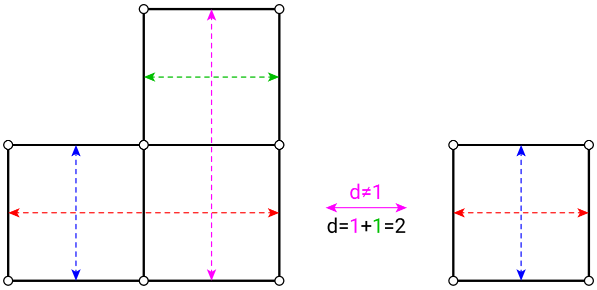


Fig. 8. Collateral strip deletion occurs when deleting one or several strips, like the one in pink, results in the deletion of other strips, the one in green. Applying such a deletion rule counts as multiple deletion rules.

in mesh M_2 are tested. If a pair of combinations yields two isomorphic quad meshes after deletion of the respective combinations of strips, then the process stops. Otherwise, k is incremented by one, and the process is repeated to find the submesh.

If there is no such submesh, then the largest similar structure is a degenerated single strip. The distance between the two quad meshes is then the maximum:

$$d_{max} = (n_1 - 1) + (n_2 - 1) = n_1 + n_2 - 2. \tag{6}$$

When the search finds such a submesh, summing the two distances to the submesh yields the distance between the input meshes, as the equality case of the triangle inequality. Indeed, the enumeration of combinations of rules in an increasing order guarantees finding the largest mesh, resulting from the minimum number of rules needed. If the submesh $\underline{M}$ has n_0 strips, then the distance equals:

$$\begin{aligned} d(M_1, M_2) &= d(M_1, \underline{M}) + d(\underline{M}, M_2) \\ &= n_1 - n_0 + n_2 - n_0 \\ &= n_1 + n_2 - 2n_0. \end{aligned} \tag{7}$$

The distance can be expressed by the number of iterations k as well as:

$$n_0 = n_2 - k, \tag{8}$$

therefore:

$$d(M_1, M_2) = n_1 - n_2 + 2k. \tag{9}$$

Fig. 9 shows the computation of the distance between two meshes with five strips each, finding two non-isomorphic submeshes $\underline{M'}$ and $\underline{M''}$ with four strips, so a distance of two. Finding a single submesh is sufficient to compute the distance, however, collecting all of the non-isomorphic submeshes enriches topological combination (Section 2.3).

Deletion rules with collateral deletions are discarded to prevent overestimation of the distance, as they are equivalent to multiple deletion rules that should appear only for a higher value of k.

2.2.3. Implementation

The implementation of the computation of the topological distance between two quad meshes is detailed in Algorithm 1. It is still assumed $n_1 > n_2$ for the input quad meshes M_1 and M_2. The algorithm returns the distance; all the non-isomorphic submeshes; and the strip deletions to apply between each pair of input mesh and submesh.

```
start empty list of results
$n_1$ and $n_2$ = number of strips in $M_1$ and $M_2$
assume $n_1 > n_2$
for $k$ from 0 to $n_1 - 2$ do
    for combinations of $n_1 - n_2 + k$ strips $S_1$ in $M_1$ do
        copy $M_1$ as $M_{10}$
        delete strips $S_1$ in $M_{10}$
        if collateral strip deletion in $M_{10}$ then
            skip this combination
        end
        for combinations of $k$ strips $S_2$ in $M_2$ do
            copy $M_2$ as $M_{20}$
            delete strips $S_2$ in $M_{20}$
            if collateral strip deletion in $M_{20}$ then
                skip this combination
            end
        end
        if meshes $M_{10}$ and $M_{20}$ are isomorphic then
            if $M_{10}$ is not isomorphic to any submesh in results then
                $\underline{M} = M_{10}$
                $d = n_1 - n_2 + 2k$
                add to results $d$, $S_1$, $S_2$, and $\underline{M}$
            end
        end
    end
    if results are not empty then
        return results
    end
    return max distance = $n_1 + n_2 - 2$
end
```

Algorithm 1: Pseudo-code for computing the topological distance and the dissimilar strips between two quad meshes M_1 and M_2 through their submesh.

2.2.4. Verification

The examples in Fig. 10 are used to verify the distance computation, with numerical results in Table 1 including the different number of strips, the distance, as well as the computation time. The examples highlight all potential outcomes, multiple (Fig. 10(a)), single (Fig. 10(b)), or no (Fig. 10(c)) submeshes. The example in Fig. 10(e) highlights the increase in computation time, from under a second to a dozen seconds for larger distances, due to a combinatorial increase of isomorphism checks to perform.

Several improvements can be considered regarding computation efficiency. First, using parallel computing with multi-threading for the enumeration of rules and isomorphism checks. Second, performing a preliminary isomorphism check on the strip graph of quad meshes [33], as a necessary but not sufficient condition, as the strip graph describes part of the connectivity of the quad mesh with a lighter though partial data structure.

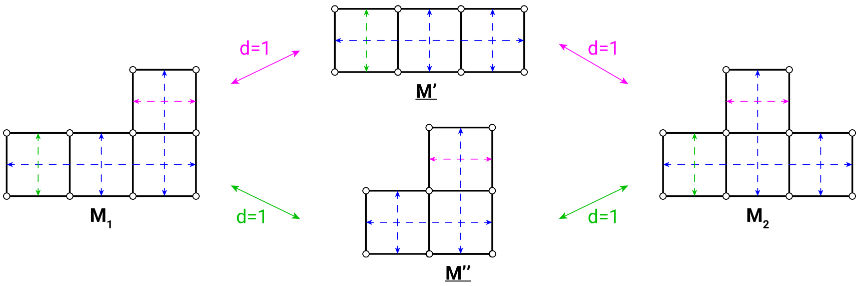
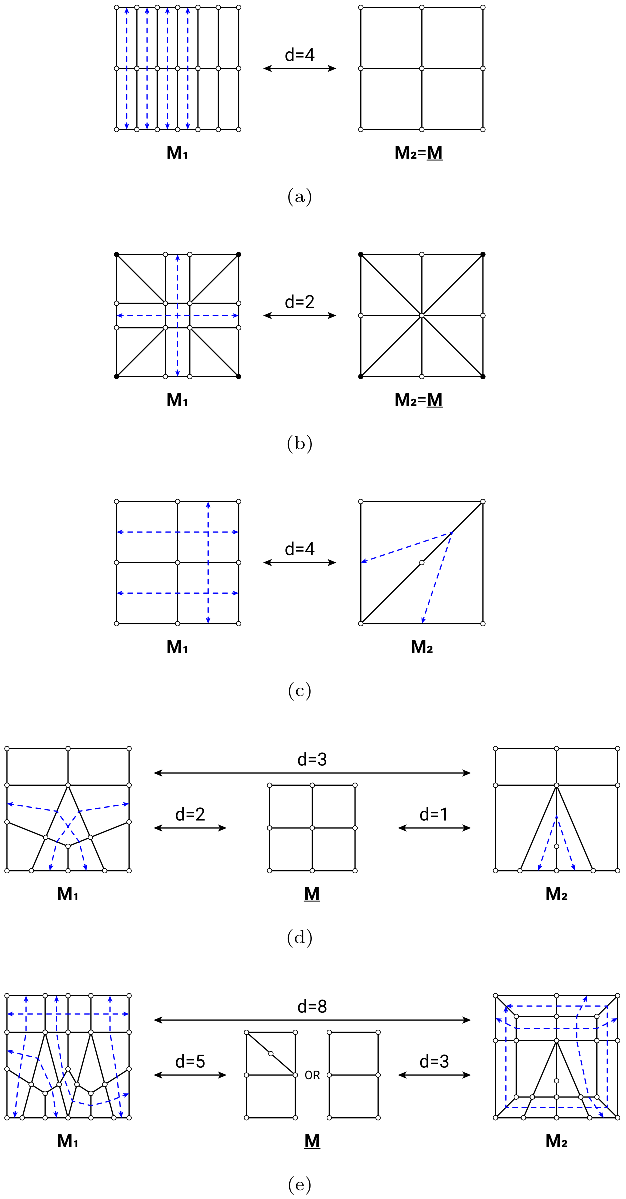


Fig. 9. Finding the distance between two quad meshes M_1 and M_2 via a common submesh $\underline{M}$ using strip deletion rules only. Multiple submeshes, $\underline{M'}$ and $\underline{M''}$, that are non-isomorphic and have the same number of strips and at the same distance of input quad meshes M_1 and M_2 can exist due to the deletion of different combinations of strips, resulting from the deletion of the pink or green strips.



(a)

(b)

(c)

(d)

(e)

Fig. 10. Verification examples for computing the topological distance between two quad meshes. The distances between the meshes M_1, M_2, and their submeshes $\underline{M}$ are provided along with one of the possible combinations of strips to delete in blue. Pole singularities are marked with black vertices.

Table 1
Numerical results for the verification examples for the computation of the topological distance between two quad meshes.

Figure	10(a)	10(b)	10(c)	10(d)	10(e)
n_1	8	10	4	6	8
n_2	4	8	2	5	6
n_0	4	8	–	4	3
k	0	0	2	1	3
d	4	2	4	3	8
# submeshes	15	1	–	1	2
# iso. checks	55	41	5	62	1516
comp. time [s]	0.58	0.52	0.025	0.30	12.4

2.3. *Quad-mesh combination*

Based on the computation of the distance between two meshes along with the identification of their similar and dissimilar strips, hybrid quad meshes can be generated to produce an interpolation between the input meshes at different distances with varying degrees of topological similarity.

2.3.1. *Approach*

The approach for the generation of hybrid meshes is as follows:

1. the **input meshes** M_1 and M_2 are obtained by any generation means;
2. the **submeshes** $\underline{M}$ are computed, which comprise the largest sets of common strips of the input meshes. A submesh is the *intersection* of the sets of strips;
3. the **supermeshes** $\overline{M}$ are computed, which comprise the smallest set of all the strips of the input meshes. A supermesh is the *union* of the sets of strips;
4. the **intermediary meshes** are enumerated, which comprise different combinations of sets of strips of the input meshes.

Fig. 11 illustrates these steps. In this example, two input meshes with respectively three and four strips yield seven hybrid meshes that are not isomorphic and therefore offer different design options. These hybrid meshes include one submesh with two strips, three supermeshes with five strips, and three intermediary meshes with three to four strips. Each hybrid mesh can result from different combinations of input strips, due to symmetry, for instance, but only one of these combinations is shown here. The coloured strips highlight the similarity between each hybrid mesh and the input meshes, provided as the similarity s, based on the distance d to the input meshes. They respectively correspond to the number of strips present in the common strip structure, and the absent ones, related to a given input mesh M_i with n_i strips as:

$$s_i = n_i - d_i. \tag{10}$$

2.3.2. *Implementation*

The generation of the hybrid meshes proceeds first with the intersection submeshes, then the union supermeshes, and finally the intermediary meshes.

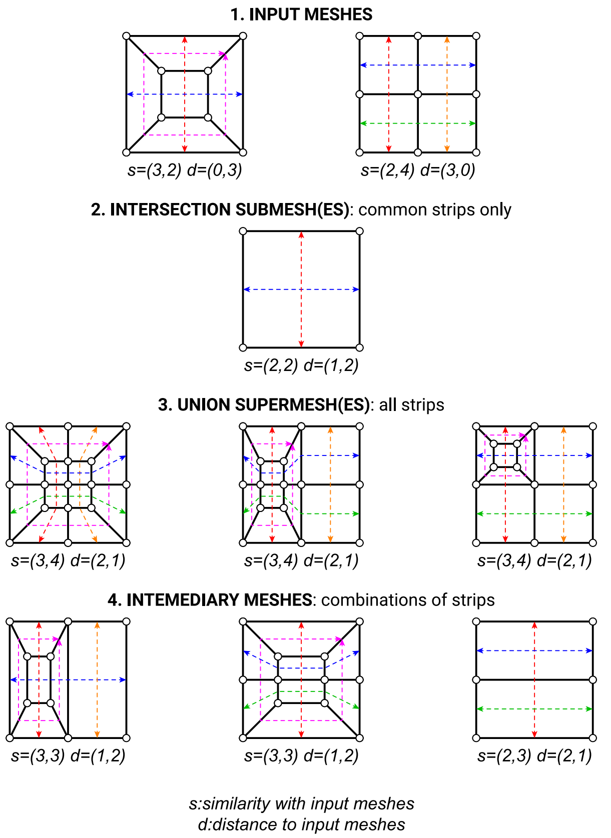

1. INPUT MESHES
s=(3,2) d=(0,3)
s=(2,4) d=(3,0)
2. INTERSECTION SUBMESH(ES): common strips only
s=(2,2) d=(1,2)
3. UNION SUPERMESH(ES): all strips
s=(3,4) d=(2,1)
s=(3,4) d=(2,1)
s=(3,4) d=(2,1)
4. INTEMEDIARY MESHES: combinations of strips
s=(3,3) d=(1,2)
s=(3,3) d=(1,2)
s=(2,3) d=(2,1)
s:similarity with input meshes
d:distance to input meshes

Fig. 11. Generating hybrid quad meshes with different degrees of topological similarity with input quad meshes. The colours correspond to similar strips, supporting the values on the similarity s and dissimilarity/distance d with the input meshes.

Intersection submeshes. Computing the distance between the input meshes also yields the submeshes (Section 2.2). All non-isomorphic submeshes with the largest number of strips are collected. This process is extended to multiple input meshes as follows:

1. the submeshes $\underline{M_{1,2}}$ between two input meshes M_1 and M_2 are computed;
2. new submeshes $\underline{M_{1,2,3}}$ between each submesh $\underline{M_{1,2}}$ and another input mesh M_3 are computed;
3. new submeshes $\underline{M_{1,\dots,i}}$ between each submesh $\underline{M_{1,\dots,i-1}}$ and another input mesh M_i are computed;
4. the final submeshes are $\underline{M_{1,\dots,N}}$, which are the non-isomorphic meshes that share the largest strip structure in common between the N input meshes M_1 to M_N are computed.

The submeshes contain all the similar strips. The reciprocal polyedges of the dissimilar strips serve as input for an addition rule to obtain these dissimilar strips. When deleting a strip S^j in a mesh M_i, the resulting polyedge P_i^j is stored. This polyedge is updated during the deletion of the other strips. For each submesh, there is a set of polyedges that correspond to the addition rules to apply to revert the applied deletion rules. These rules are used to obtain supermeshes from the submeshes. Let n_0 be the number of strips in a submesh $\underline{M}$ between two meshes M_1 and M_2 with n_1 and n_2 strips, respectively. Exploring all combinations of strip deletions from two input meshes to a submesh provides a sum for the number of the combination of rules $r_{M_{1,2}\to\underline{M}}$ to obtain potential submeshes:

$$r_{M_{1,2}\to\underline{M}} = 2^{n_1-n_0} + 2^{n_2-n_0}. \tag{11}$$

Union supermeshes. Supermeshes are the non-isomorphic meshes that contain the strip structure of each input mesh. For each submesh, strip additions are applied to all stored polyedges. After each strip addition, the polyedges for the remaining addition rules is updated. The addition rules on the longest polyedges are applied first as some polyedges may initially correspond to a single vertex.

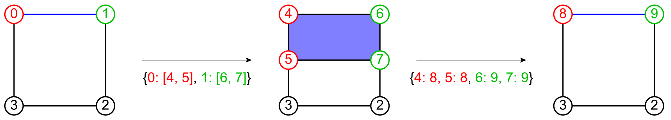


Fig. 12. Maps for vertex relabelling, in red and green, after strip addition and strip deletion rules, in blue.

A supermesh $\overline{M}$ is obtained by adding the missing strips on a submesh $\underline{M}$. Exploring all combinations of strip deletions through their reciprocal strip additions provides a product for the number of the combination of rules $r_{\underline{M}\to\overline{M}}$ to obtain potential supermeshes:

$$r_{\underline{M}\to\overline{M}} = 2^{n_1-n_0} \cdot 2^{n_2-n_0} = 2^{n_1+n_2-2n_0}; \tag{12}$$

Full combination increases the potential number of hybrid meshes. In the general case of N input meshes, the number of combinations is:

$$\prod_{i=1}^{N} 2^{n_i-n_0} = 2^{\sum_{i=1}^{N}(n_i-n_0)}. \tag{13}$$

The supermesh combines all the strips before exploring its combinations of strips.

Intermediary meshes. The intermediary meshes result from the combination of deletions of dissimilar strips on supermeshes. Deleting no strips yields the supermesh with the maximum number of strips $n_{max} = \sum_i (n_i - n_0)$. Deleting all the strips yields the submesh with the minimum number of strips $n_{min} = n_0$. Other deletion combinations yield the intermediary meshes with n_{min} to n_{max} strips, strictly. For $\Delta n = n_{max} - n_{min}$ strips to delete in a supermesh, the combinations to apply equals:

$$\sum_{k=0}^{\Delta n} \binom{\Delta n}{k} = 2^{\Delta n}. \tag{14}$$

2.3.3. Data structure

During the generation process, strip rules are transposed and reversed from one mesh to another using the input and output polyedges thanks to specific management of the polyedge and strip data.

Updating polyedges. Applying strip rules modifies the topology, the number of elements and their connectivity in the quad mesh. As a result, the labelling of the vertices changes, as shown in Fig. 12. Therefore, polyedges must be updated to apply multiple rules.

Adding a strip duplicates the vertices of the input polyedge. The new edges of a polyedge must be selected to rebuild a new continuous polyedge. Moreover, the polyedge must have its extremities on the boundary, unless the corresponding strip is closed or has pole extremities. The polyedge update considers the shortest edge path to rebuild a polyedge out of the several possibilities, found using an A* search [37]. On the other hand, deleting a strip merges the vertices of the edges of the input strip.

Mapping strips. The strips are mapped from one mesh to another by mapping polyedges for rule application. From the input meshes, strips are deleted to obtain the submesh. From the submeshes, strips are added to obtain the supermeshes. Fig. 13 shows this process for two input meshes yielding two non-isomorphic submeshes, four non-isomorphic supermeshes, and two non-isomorphic intermediary meshes. The strip structure of one of the input meshes is coloured in all meshes to highlight the similarity with this input mesh.

Fig. 14 details the mapping process of the polyedges from the input meshes to the submeshes, along which strips can be added to obtain the

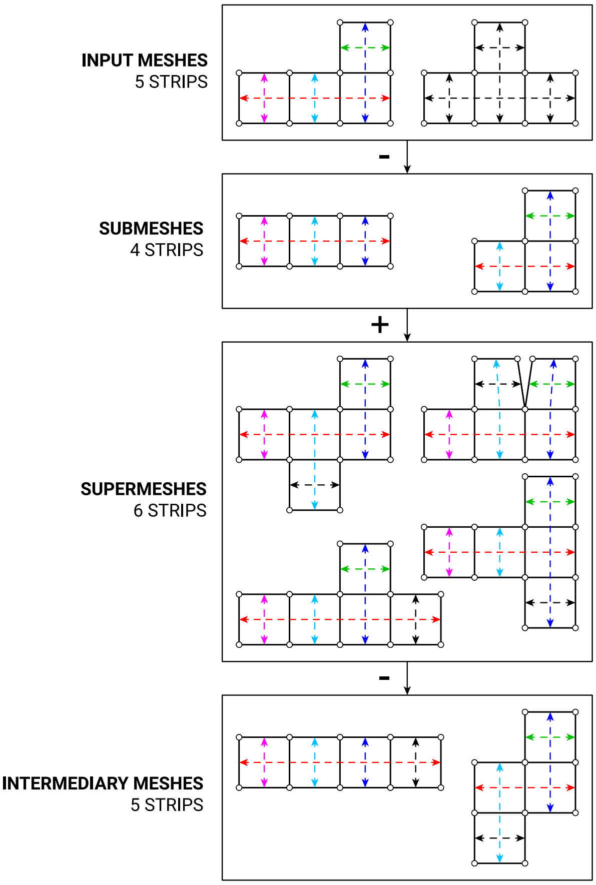


Fig. 13. Deleting the dissimilar strips between input meshes yields the submeshes. Adding the dissimilar strips to the submeshes yields the supermeshes. Deleting combinations of dissimilar strips to the supermeshes yields the intermediary meshes. The strip structure of one of the input meshes is coloured in all hybrid meshes to highlight the similarity with this input mesh.

supermeshes, all shown in Fig. 13. As the input meshes have two non-isomorphic submeshes, the polyedge-mapping process applies to each.

Deleting the dissimilar strips on each input mesh M_i results in a submesh $\underline{M_i}$ that is isomorphic to the other ones $\underline{M_j}$. These submeshes $\underline{M_i}$ and $\underline{M_j}$ are isomorphic, but there can be multiple bijections that map one submesh to another. In Fig. 14 for instance, the red polyedge in $\underline{M_i}$ from M_i and the blue polyedge in $\underline{M_j}$ from M_j can be combined in two ways. This situation results from the four possible bijections between the twofold symmetrical submeshes, two of which are not isomorphic. The deleted strips provide reciprocal polyedges to apply the addition rule to obtain the supermesh. However, several possibilities exist to map the polyedges, two in this case. The multiple strips are added, while updating the other polyedges, resulting in the supermeshes. In Fig. 14, the two submeshes have two polyedge maps each, resulting in the four supermeshes shown in Fig. 13. Deleting strips in the supermeshes yields the intermediary meshes shown in Fig. 13. Here, only one strip is deleted, to obtain intermediary meshes with five strips, between the four-strip submeshes and the six-strip supermeshes. Most are isomorphic to the other meshes, except for two intermediary meshes.

3. Structural design application

This section shows how to perform similarity-driven topology finding to search for hybrid topological designs offering different multi-objective trade-offs using the previous generation methodology. Examples show the potential of hybrid topology search for enumeration based on a single design (Section 3.1) and combination from multiple designs (Section 3.2). Different workflows, state-of-the-art form-finding and shape-optimisation strategies, and performance analyses and metrics are considered to highlight the flexibility and complementarity of topology finding within various design workflows.

3.1. *Enumeration from a single design*

In this example, 2^n potential designs are enumerated by deleting the different combinations of strips from an input design with n strips. By applying n rules on a design, n other designs are generated. This set of designs, reduced as some are isomorphic, provides a gradient of similarities with the input design.

3.1.1. *Design problem*

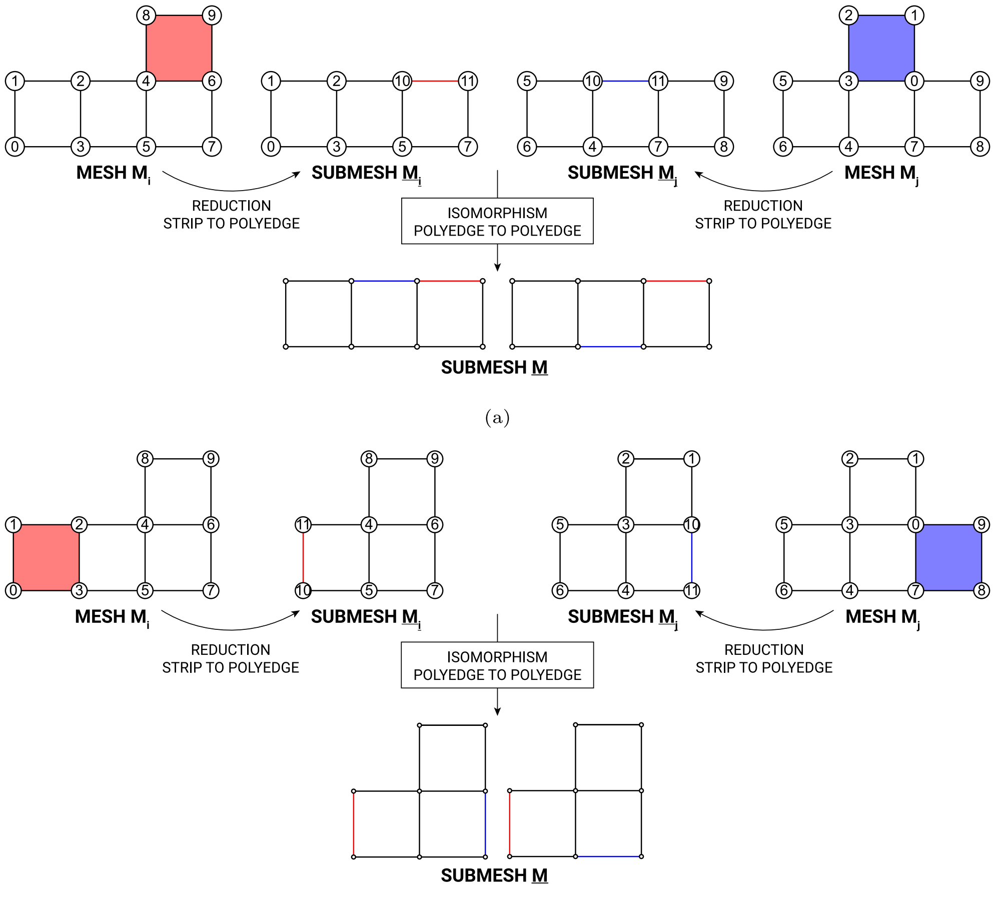
Let us consider the design of a quad-mesh pattern for a form-found gridshell on an elliptic boundary of 15 m by 10 m. First, a compression-only geometry is found using the Force Density Method [38]. The load path LP is used to assess structural performance as a metric to minimise $LP = \sum_{e \in E} f_e l_e$ with E the set of edges in the mesh, f_e and l_e the force and length of edge e, respectively. Then, the faces are planarised for fabrication rationality, which induces a deviation from the initial compression-only geometry, assessed with the deviation metric D equal to the average movement of the vertices when enforcing planarisation. The remaining average curvature C of the panels is assessed, as well as the skewness S of the panels, to minimise material loss during the cutting process. Curvature is measured as $C = 1/A_m \sum_f A_f C_f$ with A_m the mesh area, A_f the area of face f, and C_f the curvature of face f as $C_f = 2dL/(L_1 + L_2)$ where L_1 and L_2 are the lengths of the diagonals of the quad face and dL the shortest distance between them. Skewness is measured as $S = 1/A_m \sum_f A_f S_f$ with S_f the skewness of face f as $S_f = \max((\theta_{max} - 90)/90, (90 - \theta_{min})/90)$ where θ_{min} and θ_{max} are the minimal and maximal angles in degrees between two consecutive edges in the quad face.

3.1.2. *Pattern design*

A couple of heuristics informed the rule-based design of the input topology of the coarse mesh in Fig. 15, including strips that form a grid to reduce face skewness and strips that follow the curvature of an ellipsoidal dome to reduce face curvature. The addition rules are shown with dashed blue curves following the polyedges along which to add the strip. The dense quad mesh in grey helps for visualisation without representing the actual density. The eight strip additions on the initial two-strip topology result in a ten-strip topology, considered as the supermesh, shown in Fig. 16. The ten strips are gathered in six groups to preserve the twofold symmetry, represented by the colour scheme. The enumeration of the $2^6 = 64$ deletions of the strip groups yields the 26 quad meshes in Fig. 17. This reduction from 64 to 26 stems from: excluding isomorphic topologies that result from the deletion of different sets of strips; excluding deletions of strips that result in topologies with a different number of boundaries; and excluding redundant denser versions of the same topology, like the 2 × 2 grid and the 1 × 1 grid.

The topologies shown in Fig. 17 are organised based on topological similarity with the supermesh, mesh 0, the closer the more similar. However, this organisation does not reflect the distance between other pairs of meshes.

The topologies are all processed with the same workflow to obtain the quad-mesh patterns. The strips are subdivided equally with a target of a total of 500 mesh faces. The mesh is relaxed with area-weighted Laplacian smoothing with boundary vertices constrained to the elliptic boundary. The mesh is form-found with the Force Density Method [38] with fixed boundary vertices, a uniform force density of 1, and a uniform vertical load distributed on the vertices weighted by their projected tributary area. The mesh is planarised with an iterative procedure that independently projects each face to its best-fit plane



(a)

(b)

Fig. 14. Combining vertex maps to find polyedges across meshes with different numbers and labels of vertices. The strips become reduced polyedges through the rule maps from the input meshes M_i and M_j to the submeshes $\underline{M_i}$ and $\underline{M_j}$, which are mapped across the submeshes $\underline{M}$ via isomorphism search. Other submeshes between the same initial meshes result in different maps and therefore other possibilities for the generation of hybrid meshes.

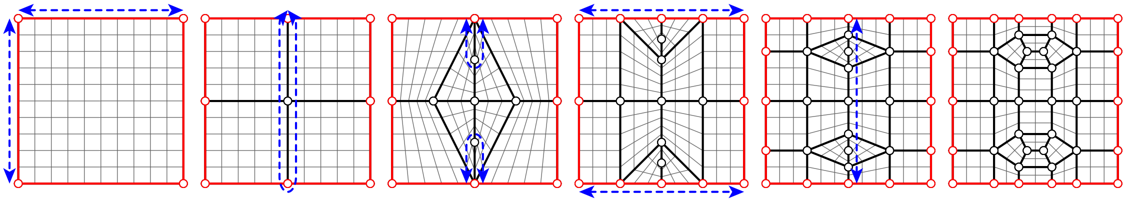


Fig. 15. Heuristic rule-based exploration with addition of strips along the polyedges highlighted with blue dashed lines.

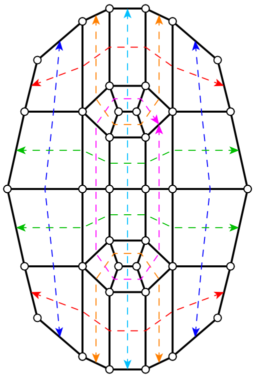


Fig. 16. The supermesh has ten strips, gathered in six coloured groups to preserve the twofold symmetry.

before merging the disconnected vertices [39,40]. The resulting designs are shown in Fig. 18, with the mesh singularities in pink, the dense mesh in grey, the coarse mesh in black with the boundary in red, and the bar charts of the multiple performance objectives to minimise.

3.1.3. Numerical results

Table 2 shows the numerical results of each design for each objective, plotted as bar charts in Fig. 18. The lower the metrics, the better the design. The maximum value of each metric X is used to normalise it as $\overline{X}$. The designs present different trade-offs between the multiple objectives, with a Pareto front consisting of five designs (0, 1, 14, 18, and 19). The Pareto designs, which represent 19% of the design pool, are underlined in Table 2 and Fig. 18. With a low number of designs and a high number of objectives, the Pareto front can only be represented coarsely. Releasing the Pareto condition considers more designs close to the Pareto front, controlling performance loss and maintaining diversity in the pool of designs for upcoming considerations in the design process [41]. The 0.95-Pareto front adds nine designs (2, 5, 6, 7, 13,

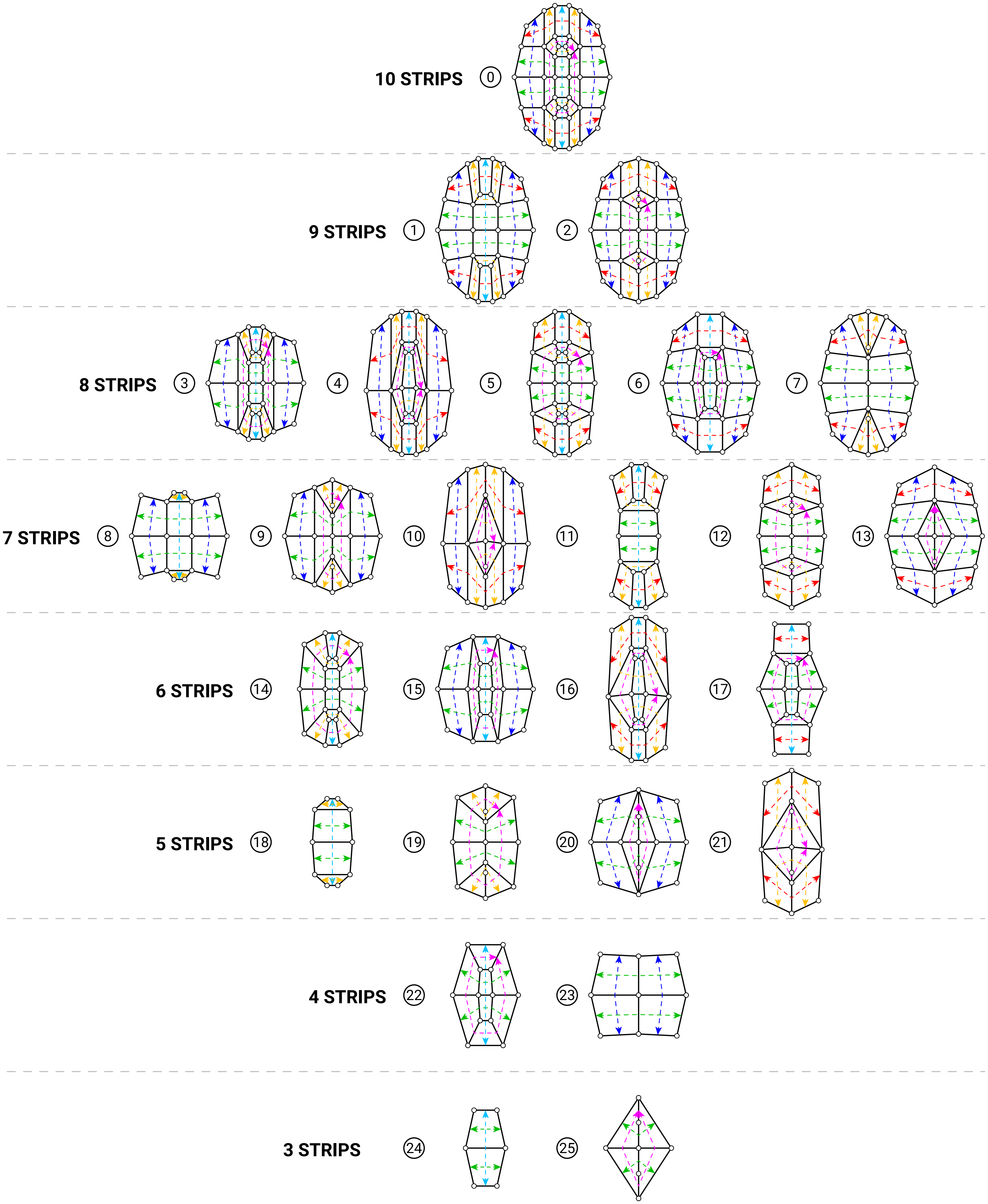


Fig. 17. Enumeration of the deletions of the coloured strip groups in mesh 0 to generate 25 other meshes. These meshes share a different number of these strips and, therefore, feature different degrees of similarity with mesh 0.

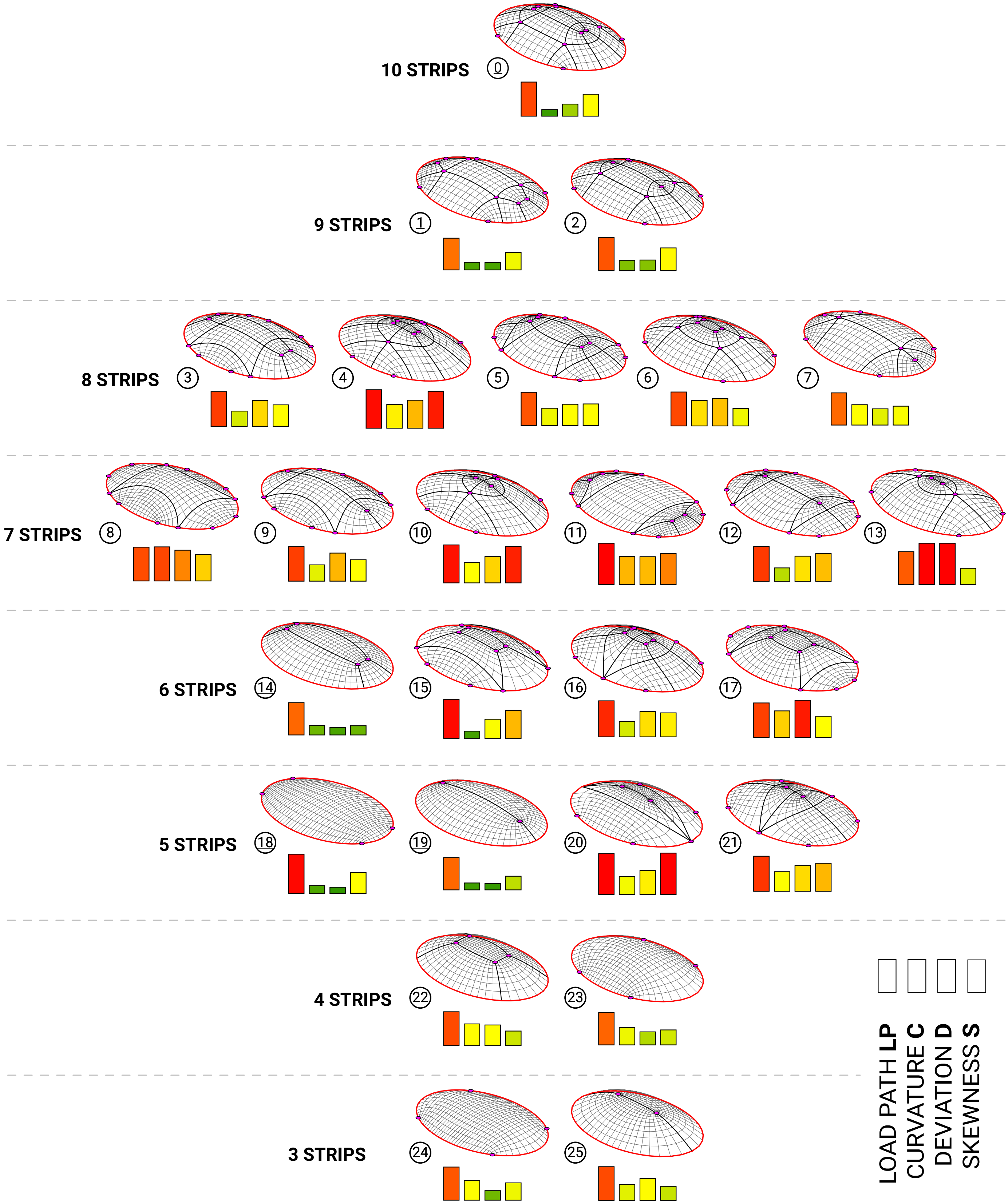


Fig. 18. Enumerating different topological designs for a gridshell on an elliptic boundary. The designs are organised based on topological similarity with design 0. The bar charts show the multi-objective trade-offs. The Pareto designs are underlined.

Table 2
Performance metrics of the hybrid designs in Fig. 18: normalised load path $\overline{LP}$, normalised panel curvature $\overline{C}$, normalised deviation due to planarisation $\overline{D}$, and normalised panel skewness $\overline{S}$. The Pareto designs are underlined, the lowest values in green, and the highest ones in red.

	$\overline{LP}$ [-]	$\overline{C}$ [-]	$\overline{D}$ [-]	$\overline{S}$ [-]
0	0.83	0.16	0.29	0.53
1	0.77	0.18	0.18	0.42
2	0.81	0.25	0.26	0.55
3	0.84	0.37	0.63	0.52
4	0.93	0.58	0.68	0.89
5	0.81	0.42	0.52	0.53
6	0.82	0.62	0.66	0.42
7	0.78	0.49	0.39	0.46
8	0.82	0.82	0.74	0.64
9	0.84	0.39	0.68	0.52
10	0.92	0.49	0.64	0.88
11	1	0.68	0.67	0.75
12	0.84	0.32	0.61	0.67
13	0.8	1	1	0.39
14	0.78	0.23	0.18	0.22
15	0.94	0.17	0.46	0.68
16	0.87	0.37	0.61	0.58
17	0.83	0.64	0.9	0.51
18	0.95	0.19	0.15	0.5
19	0.78	0.17	0.15	0.33
20	0.99	0.43	0.58	1
21	0.85	0.48	0.62	0.68
22	0.82	0.53	0.5	0.35
23	0.78	0.42	0.32	0.36
24	0.8	0.48	0.23	0.42
25	0.82	0.39	0.53	0.34
min.	0.77	0.16	0.15	0.22
max.	1.00	1.00	1.00	1.00
avrg.	0.85	0.46	0.53	0.55
st. dev.	0.07	0.23	0.24	0.22

22, 23, 24, and 25) to the strict Pareto front, consisting of 54% of the design pool.

3.2. Combination from multiple designs

By combining designs with n_i strips that have an intersection submesh with n_0 strips, $\prod_i 2^{n_i - n0}$ combinations of rules can be applied to explore designs with different degrees of similarity. However, the intersection submesh may not exist to create a reference to generate the union supermesh, especially when considering a large number of input meshes. However, the designer can add strips to the initial designs to create a submesh before the generation of hybrid meshes.

3.2.1. Design problem

Let us consider the design of a quad-mesh pattern for a gridshell constrained to a pillow-like shape. The NURBS geometry has a square boundary with a 10 m span and a 2 m height. The four corners are vertically and horizontally supported, and the structure has to withstand a uniform surface load of 2 kN/m^2 and a central point load of 100 kN, as shown in Fig. 19. The built-in steel S235 beams have an RO 114.3/4 tubular cross-section, with a stiffer RO 457.2/6.3 boundary cross-section for the local stability of the free edge [42]. The performance objectives consist of three metrics to minimise. A relative edge-length variation L measures geometrical regularity for fabrication as $L = (l_{max} - l_{min})/l_{tot}$ with l_{max} the maximum edge length, l_{min} the minimum edge length, and l_{tot} the total edge length. Two metrics measure the structural efficiency, as the strain energy E_M under the mesh load, and the strain energy E_P under the point load. A second-order mechanical analysis is performed, using the Finite Element Analysis plugin Karamba3D for Grasshopper3D [43].

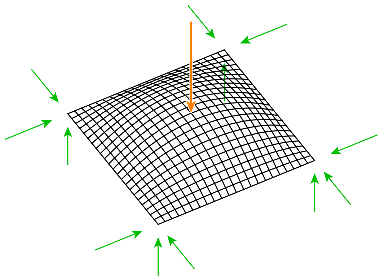
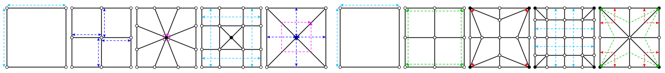


Fig. 19. Design of a pillow-shaped gridshell pinned at the corners and withstanding a point load.



(a) Temporarily adding the strips in cyan to add the strips in blue and pink to obtain a quad-mesh topology with a pole at the centre, to provide many load paths from the point load.

(b) Temporarily adding the strips in cyan to add the strips in green and red to obtain a quad-mesh topology with a pole at each corner, to provide many load paths to the supports.

Fig. 20. Two quad-mesh topologies heuristically generated from a simple coarse quad mesh.

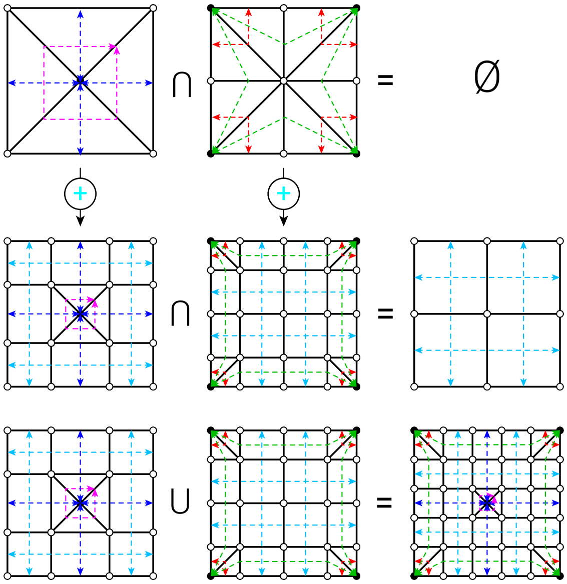


Fig. 21. The intersection between the initial designs does not yield any submesh due to the lack of a common strip structure. Adding the strips in cyan provides such a structure to obtain an intersection submesh and a union supermesh.

3.2.2. Pattern design

The design starts from a regular quad-mesh grid. Fig. 20 shows two heuristic designs obtained by applying addition and deletion rules on the initial design. Coloured dashed polylines highlight the input polyedges and strips of these rules. The colours highlight the clustering of the strips based on the twofold symmetry of the design problem.

The quad-mesh topology in Fig. 20(a) with a pole at the centre provides many load paths from the point load, whereas the quad-mesh topology in Fig. 20(b) with a pole at each corner provides many load paths to the supports. To combine these two topological designs, the intersection mesh must be computed before obtaining the union supermesh. As they do not share any strip structure, addition rules are applied to create this common strip structure, as a third input design, as shown in Fig. 21. The resulting intersection submesh and union supermesh with their reciprocal strip and polyedge elements are shown in Fig. 22.

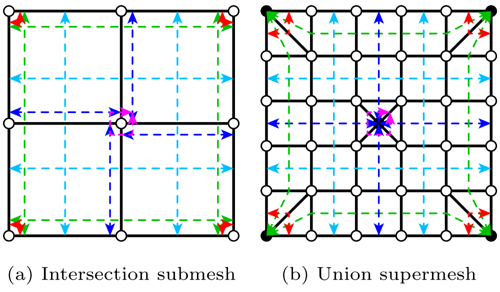

(a) Intersection submesh (b) Union supermesh

Fig. 22. The reciprocal polyedges for addition in the submesh and strips for deletion in the supermesh. The colour scheme representing the strip groups takes into account the symmetry of the design problem.

The seventeen strips in the super mesh represent five symmetrical groups for $2^5 = 32$ combinations of strip deletions. The removal of the redundant isomorphic meshes and of the meshes with a different shape topology results in the fourteen meshes with different strip structures in Fig. 23. The two initial meshes (meshes 1 and 2) and the submesh (mesh 0) mark the corners of a triangular layout, the supermesh (mesh 13) marks the centre, and the other meshes are arranged based on their relative strip similarity, for visualisation.

The coarse quad meshes are densified based on a target number of faces of 500, with each strip having the same density parameter. Surface mapping and relaxation are performed using area-weighted Laplacian smoothing [44]. The resulting fourteen quad-mesh patterns are shown in Fig. 24, arranged based on topological similarity, with their measured performance plotted as bar charts.

3.2.3. Numerical results

The numerical results are provided in Table 3. The maximum value of each metric X is used to normalise it as $\overline{X}$. The lower the metric, the more efficient the design for this metric.

Table 3
Performance metrics of the hybrid designs in Fig. 24: normalised edge-length disparity $\overline{L}$, normalised strain energy for a mesh load $\overline{E_M}$, and normalised strain energy for a point load $\overline{E_P}$. The Pareto designs are underlined, the lowest values in green, and the highest ones in red.

	$\overline{L}$ [-]	$\overline{E_M}$ [-]	$\overline{E_P}$ [-]
0	0.09	0.75	0.91
1	0.75	0.41	0.62
2	0.68	0.59	0.28
3	0.85	0.39	1
4	0.8	0.67	0.54
5	0.55	0.42	0.87
6	0.35	1	0.66
7	0.97	0.51	0.35
8	0.62	0.39	0.28
9	1	0.4	0.22
10	0.85	0.4	0.49
11	0.77	0.45	0.61
12	0.8	0.43	0.5
13	0.51	0.48	0.51
min.	0.09	0.39	0.22
max.	1.00	1.00	1.00
avrg.	0.69	0.52	0.56
st. dev.	0.25	0.18	0.24

Designs 0, 1, and 2 are among the best designs for edge-length regularity, mesh-load energy, and point-load energy, respectively, as intuitively designed. However, these designs do not perform well for the other two metrics. The designs generally offer different trade-offs between the multiple objectives. Particularly, design 13, from the supermesh, shows a balanced trade-off between the three objectives. The Pareto front consists of eight designs (0, 2, 3, 5, 6, 8, 9, and 13),

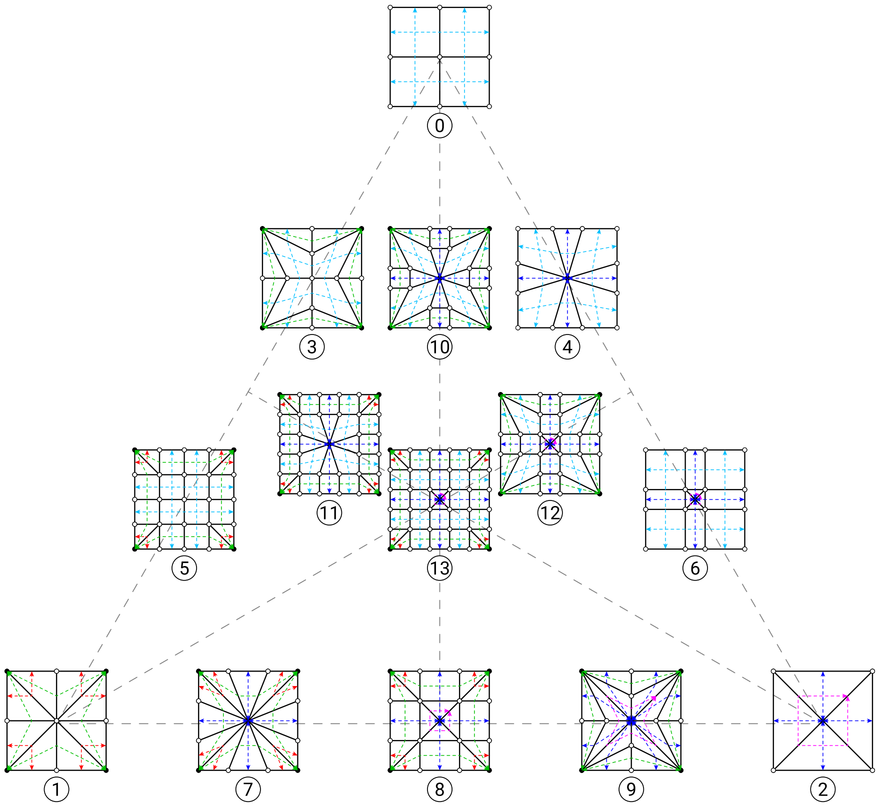


Fig. 23. Combination of the strip structures of the initial meshes and the submesh. The fourteen meshes are arranged based on topological similarity, highlighted by the common coloured strips.

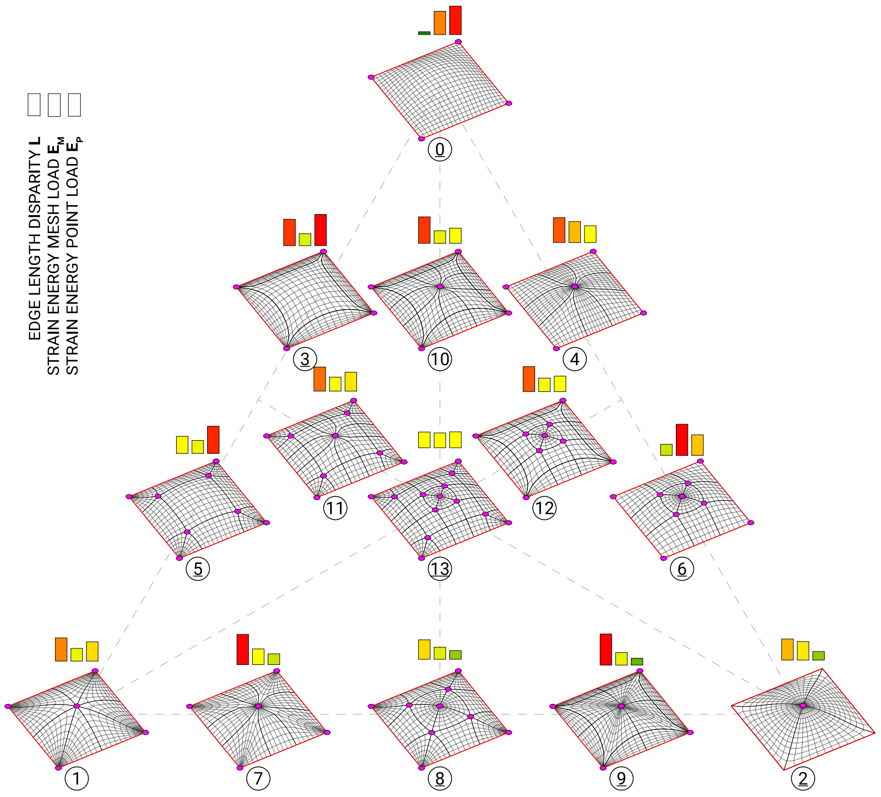


Fig. 24. Combining different topological designs for a pillow-shaped gridshell. The designs are organised based on topological similarity with designs 0, 1, and 2. The designs provide different trade-offs between geometrical regularity and structural efficiency under different load cases. The Pareto designs are underlined.

underlined in Table 3 and Fig. 24, which represent 57% of the design pool.

4. Conclusion

This paper introduced similarity-informed topology finding with the generation of hybrid pattern topologies to explore multi-objective performance trade-offs. This approach is based on generative design with a novel algebra for quad meshes with operators for distance, intersection, and union, based on a quad-mesh strip structure and a two-rule design grammar:

- an assessment was presented for topological similarity between two quad meshes with a distance;
- a strategy was developed for the combination of quad meshes to generate hybrid meshes with different degrees of topological similarity;
- this strategy was used to perform heuristic multi-objective design of quad-mesh patterns based on single-objective designs to find Pareto fronts, demonstrated on different design workflows, computational methods, performance objectives, and load cases.

This approach enables informed exploration through a heuristic combination of previous designs that perform well for different objectives, as a means to explore the daunting combinatorial design space of surface pattern topology.

Topology processing. The processing of the topology through density and geometry design is essential for the final performance of the design itself. Fig. 25 shows the topological interpolation between two quad meshes for a plate supported at its corners, based on stress fields stemming from two load cases, a symmetrical and an anti-symmetrical mesh loads. Fig. 26 illustrates their application to a ribbed concrete floor system. Only some of the hybrid topologies are shown. The integration of the stress field into a coarse quad mesh, the orientation of the edges of the dense quad mesh, and their alignment with the boundary of the plate are important design decisions. The processing of the topology into the design matters. Simultaneous topology, density, and geometry exploration and optimisation, across the different design spaces, are necessary to explore the full potential of a topological design.

Towards optimisation. For future work on the development of a novel framework for grammatical topology optimisation, several aspects must be addressed.

Although the patterns are encoded as coarse quad meshes independent of density and geometry, the combinatorial search explodes for patterns with an increasing number of singularities. Although, the examples shown feature designs with a significant number of singularities for industrial and practical application, the computation of the distance can be accelerated by leveraging parallel computing to perform the multiple independent checks and by exploring the potential of probabilistic strategies.

The discrete nature of the distance can challenge state-of-the-art exploration and optimisation algorithms if not turned into a continuous one. A larger set of descriptors that apply at the level of the mesh can be extracted. These descriptors include the number of vertices and faces, the number of boundaries, the Euler characteristic, the number of singularities per type (valency or index). Statistical methods like interpolation, regression, and dimensionality reduction can enable the creation of a latent space to obtain a continuous distance from such a set of discrete topological descriptors.

The input designs create a bias in the algorithmic search, as they bound the topological design space, making the quality of the output therefore particularly sensitive to the quality of the input, which can limit exploration if chosen poorly. Along with gene recombination, mutation is essential in evolutionary search algorithms to create diversity in the pool of genes to explore alternative design directions. With this aim, the mesh grammar can be expressed as a formal grammar,

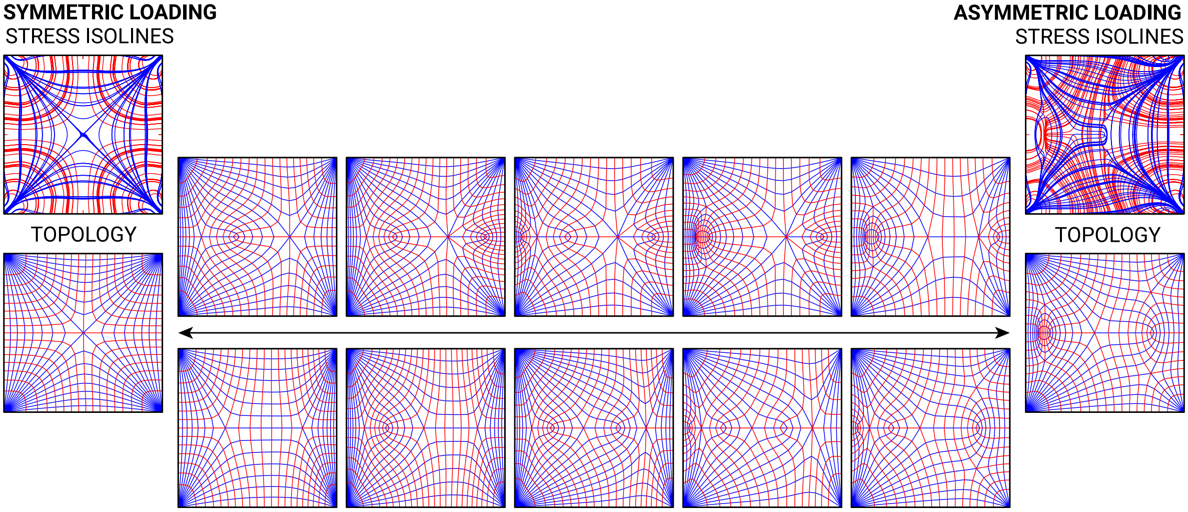


Fig. 25. Similarity-informed topology finding based on structural performance for multiple load cases.

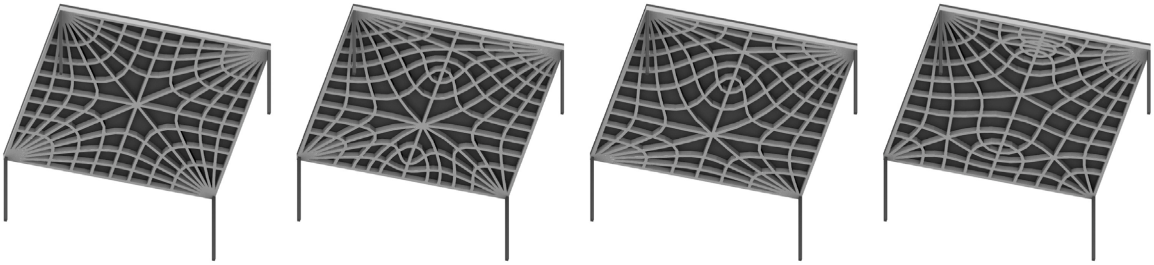


Fig. 26. Similarity-informed topology finding of hybrid topologies for ribbed concrete floors.

where the transformation rules are expressed as a string of characters or a vector of binaries [30], where randomness can be introduced in generative design. Alternatively, reinforcement learning with a reward system could steer design exploration through performance information [11].

CRediT authorship contribution statement

R. Oval: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **R. Mesnil:** Writing – review & editing, Supervision. **T. Van Mele:** Writing – review & editing, Supervision. **O. Baverel:** Writing – review & editing, Supervision, Funding acquisition. **P. Block:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgement

The authors wish to thank Maurizio Brocato who suggested framing this research as an 'algebra'.

References

[1] Bendsøe Martin Philip, Sigmund Ole. Topology optimization: theory, methods, and applications. Springer Science & Business Media; 2013.

[2] Pottmann Helmut, Eigensatz Michael, Vaxman Amir, Wallner Johannes. Architectural geometry. Comput Graph 2015;47:145–64. http://dx.doi.org/10.1016/j.cag.2014.11.002.

[3] Schiftner Alexander, Balzer Jonathan. Statics-sensitive layout of planar quadrilateral meshes. In: Ceccato Cristiano, Hesselgren Lars, Pauly Mark, Pottmann Helmut, Wallner Johannes, editors. Proceedings of the advances in architectural geometry 2010. Springer; 2010, p. 221–36.

[4] Méndez Echenagucia Tomás, Capozzoli Alfonso, Cascone Ylenia, Sassone Mario. The early design stage of a building envelope: Multi-objective search through heating, cooling and lighting energy performance analysis. Appl Energy 2015;154:577–91. http://dx.doi.org/10.1016/j.apenergy.2015.04.090.

[5] Winslow Pete, Pellegrino Sergio, Sharma Shrikant B. Multi-objective optimization of free-form grid structures. Struct Multidiscip Optim 2010;40(1–6):257. http://dx.doi.org/10.1007/s00158-009-0358-4.

[6] Mueller Caitlin T, Ochsendorf John A. Combining structural performance and designer preferences in evolutionary design space exploration. Autom Constr 2015;52:70–82. http://dx.doi.org/10.1016/j.autcon.2015.02.011.

[7] Brown Nathan C, Jusiega Violetta, Mueller Caitlin T. Implementing data-driven parametric building design with a flexible toolbox. Autom Constr 2020;118:103252. http://dx.doi.org/10.1016/j.autcon.2020.103252.

[8] Oval Robin, Rippmann Matthias, Mesnil Romain, Van Mele Tom, Baverel Olivier, Block Philippe. Topology finding of structural patterns. In: Advances in architectural geometry. 2018.

[9] Stiny George, Gips James. Shape grammars and the generative specification of painting and sculpture. In: Proceedings of the congress international federation for information processing 1971. 1971, p. 1460–5.

[10] Stiny George. Shape: talking about seeing and doing. MIT Press; 2006.

[11] Nourian Pirouz, Azadi Shervin, Oval Robin. Generative design in architecture: From mathematical optimization to grammatical customization. In: Computational design and digital manufacturing. Springer; 2023, p. 1–43. http://dx.doi.org/10.1007/978-3-031-21167-6_1.

[12] Yousaf Muhammad Salman, Detwiler Duane, Duddeck Fabian, Menzel Stefan, Ramnath Satchit, Zurbrugg Nathan, et al. Similarity-driven topology optimization for statics and crash via energy scaling method. J Mech Des 2023;145(10). http://dx.doi.org/10.1115/1.4062943.

[13] Bujny Mariusz, Yousaf Muhammad Salman, Zurbrugg Nathan, Detwiler Duane, Menzel Stefan, Ramnath Satchit, et al. Learning hyperparameter predictors for similarity-based multidisciplinary topology optimization. Sci Rep 2023;13(1):14856. http://dx.doi.org/10.1038/s41598-023-42009-0.

[14] Oh Sangeun, Jung Yongsu, Kim Seongsin, Lee Ikjin, Kang Namwoo. Deep generative design: Integration of topology optimization and generative models. J Mech Des 2019;141(11):111405. http://dx.doi.org/10.1115/1.4044229.

[15] Zhang Weisheng, Wang Yue, Du Zongliang, Liu Chang, Youn Sung-Kie, Guo Xu. Machine-learning assisted topology optimization for architectural design with artistic flavor. Comput Methods Appl Mech Engrg 2023;413:116041. http://dx.doi.org/10.1016/j.cma.2023.116041.

[16] Zhang Weisheng, Wang Yue, Youn Sung-Kie, Guo Xu. Machine learning powered sketch aided design via topology optimization. Comput Methods Appl Mech Engrg 2024;419:116651. http://dx.doi.org/10.1016/j.cma.2023.116651.

[17] Dommaraju Nivesh, Bujny Mariusz, Menzel Stefan, Olhofer Markus, Duddeck Fabian. Evaluation of geometric similarity metrics for structural clusters generated using topology optimization. Appl Intell 2023;53(1):904–29. http://dx.doi.org/10.1007/s10489-022-03301-0.

[18] Tyburec Marek, Zeman Jan, Doškář Martin, Kružík Martin, Lepš Matěj. Modular-topology optimization with Wang tilings: an application to truss structures. Struct Multidiscip Optim 2021;63(3):1099–117. http://dx.doi.org/10.1007/s00158-020-02744-8.

[19] Tyburec Marek, Doškář Martin, Zeman Jan, Kružík Martin. Modular-topology optimization of structures and mechanisms with free material design and clustering. Comput Methods Appl Mech Engrg 2022;395:114977. http://dx.doi.org/10.1016/j.cma.2022.114977.

[20] Jie Wang, Tong Gao, Ming Li, Jihong Zhu, Longlong Song, Zhang Weihong. Topology optimization of modular structures with multiple assemblies and applications to airborne shelves. Chin J Aeronaut 2024;37(4):321–32. http://dx.doi.org/10.1016/j.cja.2023.12.014.

[21] Oval Robin. Topology finding of patterns for structural design [Ph.D. thesis], Université Paris-Est; 2019.

[22] Oval Robin, Rippmann Matthias, Mesnil Romain, Van Mele Tom, Baverel Olivier, Block Philippe. Feature-based topology finding of patterns for shell structures. Autom Constr 2019;103:185–201. http://dx.doi.org/10.1016/j.autcon.2019.02.008.

[23] Campen Marcel, Bommes David, Kobbelt Leif. Dual loops meshing: quality quad layouts on manifolds. Assoc Comput Mach Trans Graph 2012;31(4):110. http://dx.doi.org/10.1145/2185520.2185606.

[24] Campen Marcel, Kobbelt Leif. Dual strip weaving: Interactive design of quad layouts using elastica strips. Assoc Comput Mach Trans Graph 2014;33(6):183. http://dx.doi.org/10.1145/2661229.2661236.

[25] Akleman Ergun, Chen Jianer, Gross Jonathan L. Strip sculptures. In: 2010 shape modeling international conference. IEEE; 2010, p. 236–40. http://dx.doi.org/10.1109/SMI.2010.14.

[26] Akleman Ergun, Ke Shenyao, Wu You, Kalantar Negar, Borhani AliReza, Chen Jianer. Construction with physical version of quad-edge data structures. Comput Graph 2016.

[27] Conway John H, Burgiel Heidi, Goodman-Strauss Chaim. The symmetries of things. CRC Press; 2016.

[28] Shepherd Paul, Pearson Will. Topology optimisation of algorithmically generated space frames. In: Proceedings of the annual symposium of the international association for shell and spatial structures 2013. 2013.

[29] Koronaki Antiopi, Shepherd Paul, Evernden Mark. Layout optimization of space frame structures. In: Proceedings of the annual symposium of the international association for shell and spatial structures 2017. 2017.

[30] Oval Robin, Mesnil Romain, Van Mele Tom, Block Philippe, Baverel Olivier. A vector encoding for topology finding of structured quad-based patterns for surface structures. Int J Space Struct 2023;38(4):327–42. http://dx.doi.org/10.1177/09560599231207650.

[31] Cordella Luigi Pietro, Foggia Pasquale, Sansone Carlo, Vento Mario. An improved algorithm for matching large graphs. In: 3rd IAPR-TC15 workshop on graph-based representations in pattern recognition. 2001, p. 149–59. http://dx.doi.org/10.1016/S0167-8655(02)00248-9.

[32] Hagberg Aric, Swart Pieter, S. Chult Daniel. Exploring network structure, dynamics, and function using NetworkX. Technical report, Los Alamos, NM (United States): Los Alamos National Lab.(LANL); 2008.

[33] Oval Robin, Mesnil Romain, Van Mele Tom, Block Philippe, Baverel Olivier. Two-colour topology finding of quad-mesh patterns. Comput Aided Des 2021;137:103030. http://dx.doi.org/10.1016/j.cad.2021.103030.

[34] Levenshtein Vladimir I. Binary codes capable of correcting deletions, insertions, and reversals. Sov Phys Doklady 1966;10(8):707–10.

[35] Hamming Richard W. Error detecting and error correcting codes. Bell Syst Tech J 1950;29(2):147–60. http://dx.doi.org/10.1002/j.1538-7305.1950.tb00463.x.

[36] Abu-Aisheh Zeina, Raveaux Romain, Ramel Jean-Yves, Martineau Patrick. An exact graph edit distance algorithm for solving pattern recognition problems. In: 4th international conference on pattern recognition applications and methods 2015. 2015, http://dx.doi.org/10.5220/0005209202710278.

[37] Hart Peter E, Nilsson Nils J, Raphael Bertram. A formal basis for the heuristic determination of minimum cost paths. IEEE Trans Syst Sci Cybern 1968;4(2):100–7. http://dx.doi.org/10.1109/TSSC.1968.300136.

[38] Schek Hans-Jörg. The force density method for form finding and computation of general networks. Comput Methods Appl Mech Engrg 1974;3(1):115–34. http://dx.doi.org/10.1016/0045-7825(74)90045-0.

[39] Deuss Mario, Deleuran Anders Holden, Bouaziz Sofien, Deng Bailin, Piker Daniel, Pauly Mark. ShapeOp—a robust and extensible geometric modelling paradigm. In: Modelling behaviour. Springer; 2015, p. 505–15. http://dx.doi.org/10.1007/978-3-319-24208-8_42.

[40] Piker Daniel. Kangaroo: form finding with computational physics. Archit Des 2013;83(2):136–7. http://dx.doi.org/10.1002/ad.1569.

[41] Brown Nathan, Tseranidis Stavros, Mueller Caitlin. Multi-objective optimization for diversity and performance in conceptual structural design. In: Proceedings of the annual symposium of the international association for shell and spatial structures 2015. 2015.

[42] Venuti Fiammetta, Bruno Luca. Influence of in-plane and out-of-plane stiffness on the stability of free-edge gridshells: A parametric analysis. Thin-Walled Struct 2018;131:755–68. http://dx.doi.org/10.1016/j.tws.2018.07.019.

[43] Preisinger Clemens. Linking structure and parametric geometry. Archit Des 2013;83(2):110–3. http://dx.doi.org/10.1002/ad.1564.

[44] Botsch Mario, Kobbelt Leif, Pauly Mark, Alliez Pierre, Lévy Bruno. Polygon mesh processing. CRC Press; 2010.