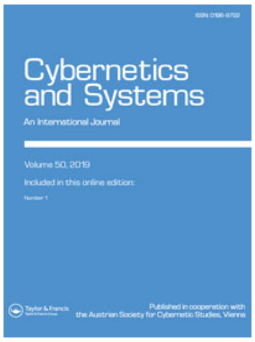


Taylor & Francis
Taylor & Francis Group

Cybernetics and Systems
An International Journal

ISSN: 0196-9722 (Print) 1087-6553 (Online) Journal homepage: https://www.tandfonline.com/loi/ucbs20

A General Meta-graph Strategy for Shape Evolution under Mechanical Stress

Diego Montoya-Zapata, Diego A. Acosta, Oscar Ruiz-Salguero, Jorge Posada & David Sanchez-Londono

To cite this article: Diego Montoya-Zapata, Diego A. Acosta, Oscar Ruiz-Salguero, Jorge Posada & David Sanchez-Londono (2019) A General Meta-graph Strategy for Shape Evolution under Mechanical Stress, Cybernetics and Systems, 50:1, 3-24, DOI: 10.1080/01969722.2018.1558011

To link to this article: https://doi.org/10.1080/01969722.2018.1558011

Published online: 01 Feb 2019.

CYBERNETICS AND SYSTEMS
2019, VOL. 50, NO. 1, 3–24
https://doi.org/10.1080/01969722.2018.1558011

A General Meta-graph Strategy for Shape Evolution under Mechanical Stress

Diego Montoya-Zapata[a,b], Diego A. Acosta[c], Oscar Ruiz-Salguero[a], Jorge Posada[b], and David Sanchez-Londono[a]

[a]Laboratory of CAD CAM CAE, Universidad EAFIT, Medellín, Colombia; [b]Vicomtech, San Sebastián, Spain; [c]Grupo de Desarrollo y Diseño de Procesos (DDP), Universidad EAFIT, Medellín, Colombia

ABSTRACT
The challenges that a shape or design stands are central in its evolution. In the particular domain of stress/strain challenges, existing approaches eliminate under-demanded neighborhoods from the shape, thus producing the evolution. This strategy alone incorrectly (a) conserves disconnected parts of the shape and (b) eliminates neighborhoods which are essential to maintain the boundary conditions (supports, loads). The existing analyses preventing (a) and (b) are conducted in an ad-hoc manner, by using graph connectivity. This manuscript presents the implementation of a meta-graph methodology, which systematically lumps together finite element subsets of the current shape. By considering this meta-graph connectivity, the method impedes situations (a) and (b), while maintaining the pruning of under-demanded neighborhoods. Research opportunities are open in the application of this methodology with other types of demand on the shape (e.g., friction, temperature, drag, and abrasion).

KEYWORDS
Evolutionary structural optimization; finite element analysis; mathematical graph; topology optimization

Introduction

As a result of the process of evolution, natural shapes lose the neighborhoods that do not affect their basic functions. This evolutionary process is influenced by the different stimuli (heat, friction, and stresses) to which the shape is subjected to.

Based on the response of the shape to the different stimuli, one can classify the shape neighborhoods as (1) demanded, if they are highly used to fulfill the functional requirements, or (2) under-demanded, when they are not completely necessary.

This work presents a methodology for structural optimization in which the exact nature of the stimulus may be generic. At the same time, the criterion of material removal may be also generic. Examples of such a criterion are low stress, high exposure to friction, maximization of wave

CONTACT Jorge Posada jposada@vicomtech.org Vicomtech, San Sebastián, Spain
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/ucme.

reflection (e.g., sound), etc. In this article, the material removal obeys to low stressed neighborhoods when subjected to stress/strain stimuli. Notice that, once the stimuli are calculated (by specialized outsourced software), the particular criteria for material removal can be applied in a generic manner.

This article presents a meta-graph methodology, which systematically removes material neighborhoods (represented by subsets of finite elements) of the current shape. By considering this meta-graph connectivity, the method prunes under-demanded neighborhoods while impeding the (a) disconnections on the shape, and (b) elimination of the essential neighborhoods that maintain the boundary conditions (supports, loads).

The present work generalizes the meta-graph strategy developed by Montoya-Zapata et al. (2019), so that it is used at every iteration of the optimization process. This generalization simplifies the process of material removal by using only the connectivity and nodal information of the meta-graph to decide which under-demanded neighborhoods should be eliminated, without affecting the domain connectivity and the predefined structural constraints. In addition, this work (1) presents additional examples to illustrate the behavior of the implemented meta-graph approach, (2) demonstrates the suitability of the algorithm for 3 D domains, and (3) compares the solutions of the implemented algorithm with the solutions of the Topology Optimization module of ANSYS® Academic Student, Release 19.0.

This manuscript is organized as follows: Section "Literature Review" provides a review of the related literature. Section "Methodology" describes the proposed meta-graph-based algorithm and Section "Results" presents and evaluates the results obtained following the meta-graph approach. Finally, Section "Conclusions" contains the conclusions and some possible research lines to extend this work.

Literature Review

Section "Evolutionary Structural Optimization" presents the basis of Evolutionary Structural Optimization (ESO), as well as its applications and recent improvements. Section "Graphs Representations Used with ESO" demonstrates the previous use of ESO in conjunction with graph techniques. Section "Graphs Representations in Other Structural Optimization Algorithms" shows how graph-based strategies have been applied with other optimization algorithms. Finally, Section "Conclusions of the Literature Review" presents the contribution of this work with respect to the previous research.

Evolutionary Structural Optimization

Xie and Steven (1993) introduce a structural optimization method called ESO. ESO removes progressively the low stressed portions of a structure by carrying out iterative FEA simulations. Therefore, the weight of the structure is reduced without affecting its functionality. Bidirectional ESO (BESO) (Querin et al. 1998) is an extension of ESO in which new material can be added in high-stressed zones. One of the main drawbacks of ESO and BESO is the formation of nonvalid configurations as a result of material removal, which cause the end of the optimization process.

Recent publications on improvements of ESO/BESO techniques (Ghabraie 2015; Munk et al. 2017; Da et al. 2018) and on review articles focused on ESO/BESO (Deaton and Grandhi 2014; Munk et al. 2015; Xia et al. 2018) prove that the development of these algorithms is a matter of interest for the academic community.

One of the reasons of the popularity of ESO and BESO is the wide range of engineering problems that can be addressed with these methods. Some examples are: aeronautics (Das and Jones 2011), biomedicine (Chen et al. 2011), and materials design (Huang et al. 2012).

Likewise, much of the research efforts in topology optimization are focusing on additive manufacturing (AM) (Chen et al. 2015; Tang et al. 2018; Seifi et al. 2018). The recent advances in AM allow the exploitation of the full capacities of ESO/BESO techniques. Hence, it is necessary to improve the current topology optimization algorithms so that they adapt to the manufacturing capabilities that have been gained because of AM.

Graphs Representations Used with ESO

Stojanov et al. (2016) use graphs to represent of FEA meshes, so that each graph vertex represents an FEA element. In this work, graphs are used to check the connectivity of the generated structures. Structures that do not meet the connectivity requirement are discarded. In a similar fashion, Munk et al. (2017) use a graph-based connectivity checker to extend BESO.

In these two papers (Stojanov et al. 2016; Munk et al. 2017), graph representations have been mainly used to find valid (or nonvalid) configurations of FEA meshes while using ESO algorithms and when a nonvalid configuration is found, this branch of the optimization process is not taken into account. Additionally, graphs abstractions are not integrated to the material removal process.

Montoya-Zapata et al. (2019) integrate ESO algorithm and graphs to perform 2D structural optimization. In their work, Montoya-Zapata et al. (2019) develop a meta-graph (a graph generated from subsets of elements of the FEA mesh) strategy to be used as part of the material removal routine. However, this strategy is only used in a particular case, when the connectivity of the boundary conditions is compromised.

Graph Representations in Other Structural Optimization Algorithms

Graph representations have been used in conjunction with other structural optimization techniques, apart from ESO. For instance, Giger and Ermanni (2006) focus on the topology optimization of trusses, using genetic algorithms (GA) as the basis of the algorithm to remove the useless material of the truss structure. Graph representations are mainly used to establish a criterion to test if a solution is structurally valid.

Another example of the use of GA and graph representations in structural optimization is presented by Madeira et al. (2010), who apply GA in conjunction with graph theory to the maximization of the structural stiffness. The tree-based representation of individuals ease the generation of feasible solutions, avoiding the use of repairing operators.

Conclusions of the Literature Review

Structural optimization and specifically ESO/BESO are topics of interest because of the multiple application fields in which they can be used. In particular, the use of topology optimization in AM is a necessity. Therefore, it is necessary to improve current topology optimization algorithms.

The use of graph abstractions is common in structural optimization algorithms that employ both ESO (Stojanov et al. 2016; Munk et al. 2017) and non-ESO techniques (Giger and Ermanni 2006; Madeira et al. 2010). However, graphs are mainly used to check the connectivity of the generated solutions and they are not integrated into the material removal algorithm.

As a response to this limitation, the present work implements a generalization of the meta-graph based strategy presented by Montoya-Zapata et al. (2019), so that it is used at every stage of the optimization process. This upgraded strategy administers the neighborhood and static connectedness of the finite elements to support evolutionary shape optimization. In this way, the material removal algorithm is simplified and fully based on the meta-graph information (meta-graph connectivity and meta-nodes degree). In addition, this work: (1) presents additional examples to support and illustrate the behavior of the implemented meta-graph approach, (2) demonstrates the suitability of the algorithm for 3 D domains, and (3) compares the solutions of the implemented algorithm with the solutions of commercial software (ANSYS®).

The present work does not try to evaluate or apply alternative genetic strategies (e.g., mutation and crossover operators), but this could be addressed in subsequent research. Future work can also be focused on widen the variety of stimuli (kinematics, abrasion, temperature), in addition to mechanical stress, that drive evolution in the nature domain.

Methodology

In Section "Problem Statement" is given a formalization of the problem of structural optimization that this work aims to tackle. Section "Structural Optimization Algorithm" presents the implemented structural optimization algorithm and Section "Element Deletion Algorithm" shows with detail the meta-graph-based deletion algorithm. In addition, Section "Boundary Synthesis" presents an alternative for postprocessing the boundary of the resultant shape, so that it is suitable for manufacturing.

Problem Statement

In general, the goal of structural optimization is to produce a final shape (optimal) that is (1) functional, and (2) that uses the least amount of material. The requirement of *functionality* is determined by some permissible levels of demand that the domain can stand when a stimulus is acting over it (e.g., forces, heat, and abrasion). Sections "Given" and "Goal" formalize the concept of structural optimization following a *Given/Goal* scheme.

Given

1. Let $\Omega_0 \subset \mathbb{R}^2$ be a compact and bounded domain that represents an initial oversized domain.
2. A stimulus function S that acts over $\Omega_S \subset \Omega_0$.
3. FEA mesh $M_0 = (N_0, E_0)$ for Ω_0, where N_0 is the set of nodes and E_0 is the set of elements.

Goal

1. To obtain the design domain $\Omega_F \subset \Omega_0$ that solves the optimization problem:

$$\begin{aligned} &\min_{\Omega} \quad A(\Omega) \\ &s.t. \quad f(x) \leq g(x), \qquad \text{for all } x \in \Omega, \Omega \subset \Omega_0 \\ &\Omega_S \subset \Omega \end{aligned}$$

where $A(\Omega)$ is the area of Ω, $f(x)$ is the response for $x \in \Omega$ to the stimuli S, and $g(x)$ expresses the permissible level of demand f that the neighborhood of a point $x \in \Omega_i$ may stand (e.g., permissible stress allowable).

Structural Optimization Algorithm

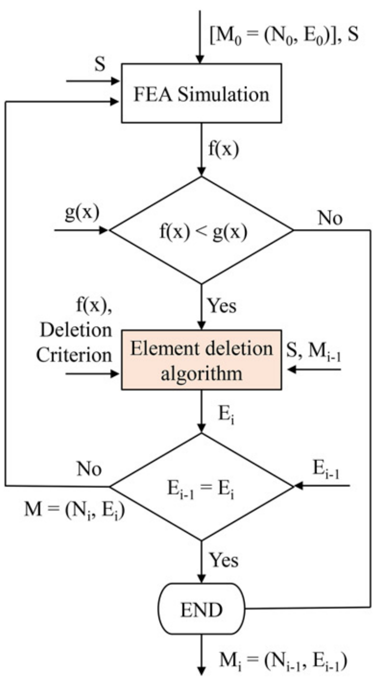
The implemented optimization algorithm follows the procedure described in Figure 1a. First, an FEA simulation is carried out, given the initial FEA mesh $M_0 = (N_0, E_0)$ and the stimulus function S. The FEA simulation allows to find the domain response f to the stimuli S. If f exceeds the permissible limit g, then the algorithm stops. Otherwise, the algorithm proceeds to delete the under-demanded FEA elements. The sub-algorithm that performs the deletion of the FEA elements is presented in Figure 1b. It is described in detail in Section "Element Deletion Algorithm".

Finally, another FEA simulation is performed with the resultant domain after the deletion of the under-demanded FEA elements deletion. The cycle is repeated until no more elements can be deleted.

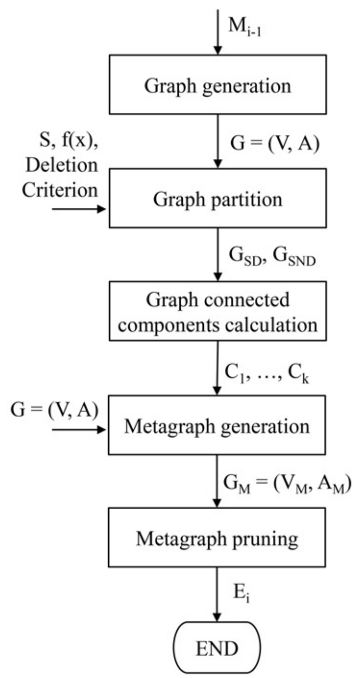
The reader may notice that the algorithm presented in this article reduces Ω_0 by removing under-demanded material (i.e., elements from E_0) given a deletion criterion. The stimulus function S and the deletion criterion are user-defined properties, and f is calculated by using FEA software. Therefore, the presented algorithm is independent to the kind of stimuli (forces, friction, abrasion, humidity, etc.) to which the domain is subjected.

Element Deletion Algorithm

The main objective of the implemented element deletion algorithm is to assure that the resultant configuration (after removing the unnecessary FEA elements) is valid from a structural point of view. The algorithm is based on



(a) Work-flow of the optimization algorithm



(b) Element deletion sub-algorithm

Figure 1. Data flow of the implemented optimization procedure.

a graph abstraction of the design domain, as presented in Figure 1b. The main stages of the algorithm are discussed in the following sections.

Graph Generation

For every FEA mesh $M = (N, E)$, a graph $G = (V, A)$ can be generated with the following procedure:

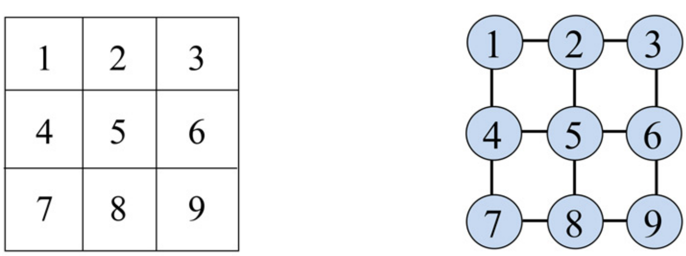
1. Assume that $E = \{e_1, e_2, \ \ldots, e_k\}$. Then, for every $e_i \in E$ create a graph vertex $v_i \in V$.
2. A graph arc $(v_i, v_j) \in A$ exists if and only if the corresponding FEA elements e_i, e_j are adjacent.

Different adjacency relations between elements can be defined for a 2 D FEA mesh. In particular, this article considers two elements as adjacent if they have a common edge. Figure 2 depicts an example of the graph associated to an FEA mesh following this FEA-edges adjacency rule.

Graph Partition

In order to carry out the elimination of the under-demanded material, the graph is partitioned into two sub-graphs G_{SD} and G_{SND} where: (i) G_{SD} contains the graph nodes that are candidates for elimination ($E_D \subset E$) and G_{SND} contains the rest of nodes ($E_{ND} = E - E_D$).

The set of nodes E_D are those nodes associated to the under-demanded FEA elements. These elements are selected based on the response function f and a *Deletion Criterion*. In this work, the *Deletion Criterion* is defined by an admissible limit for the Von Mises stress at each optimization stage. In addition, the FEA elements in which the stimulus function S acts cannot be eliminated, so they always belong to E_{ND}.



(a) FEA mesh (b) Graph edges produced by adjacent FEA edges

Figure 2. FEA mesh to graph conversion using FEA edge adjacency criteria.

Connected Components Calculation and Meta-graph Generation

The procedure to generate the meta-graph G_M associated to 1) the graph $G = (V, A)$ and 2) the set of candidate elements to be deleted $E_D \subset E$ is described below. Figure 3 shows a graphical representation of the given procedure.

1. Find the connected components of G_{SD} and denote them as $\{c_1, c_2, \ldots, c_p\}$ (see Figure 3a).
2. Find the connected components of G_{SND} and denote them as $\{c_{p+1}, c_{p+2}, \ldots, c_{p+r}\}$ (see Figure 3a).
3. Each connected component of G_{SD} and G_{SND} becomes a vertex (meta-node) of the meta-graph.
4. Two meta-nodes c_i, c_j are adjacent if and only if vertices $v_i, v_j \in V$ exist and: (a) the arc $(v_i, v_j) \in A$ exists, (b) vertex v_i belongs to the connected component c_i, and (c) vertex v_j belongs to the connected component c_j (see Figure 3b).

Meta-graph Pruning

The last step of the element deletion algorithm is the elimination of the under-demanded material (meta-graph pruning). For this purpose, the present work considers three different scenarios:

Case 1 - Noncandidate elements for deletion are fully connected: As can be seen in Figure 4a, in this scenario all the noncandidate elements to

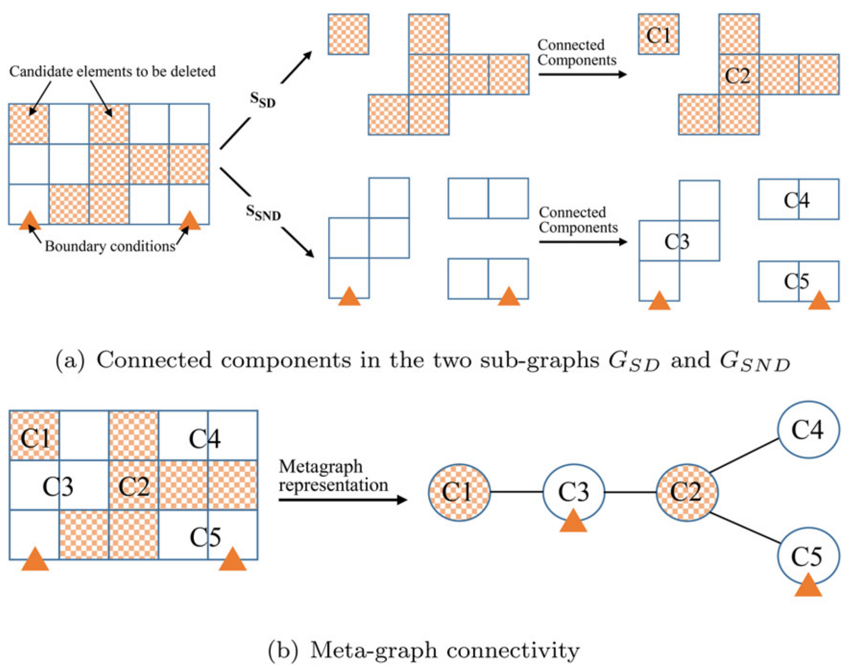


Figure 3. Meta-graph associated to an FEA mesh and the candidate elements to be deleted.

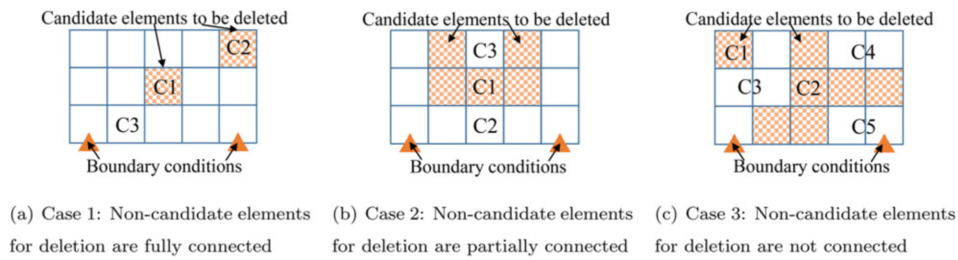


Figure 4. Material removal scenarios.

be deleted lie in the same meta-node (C_3). In this case, all the meta-nodes that contain under-demanded elements are removed (C_1 and C_2).

Case 2 - Noncandidate elements for deletion are partially connected: In this case, the noncandidate elements for deletion are not in the same meta-node. However, all elements under the action of the stimulus S do lie in the same component. An example of this scenario is shown in Figure 4b. Since the deletion of C_1 would annul the action of C_3, in this work both C_1 and C_3 are removed. The only meta-node left after deletion would be C_2. In general, in this case all the meta-nodes but the one that contains the elements with boundary conditions are deleted.

Case 3 - Noncandidate elements for deletion are not connected: In this scenario, elements with boundary conditions are not in the same meta-node. In order to preserve the connectivity of the stimulated subdomain Ω_S, a meta-node C_i is deleted if it meets these conditions: 1) C_i is a connected component of G_{SD} and 2) C_i is of degree 1.

The second condition assures that deleting C_i will not affect the connectivity of the elements with boundary conditions.

For the example illustrated in Figure 4c, the only meta-node that is deleted is C_1, since the deletion of C_2 would generate a disconnection between the meta-nodes with boundary conditions (C_3, C_5).

Boundary Synthesis

Since the presented algorithm works with FEA meshes, the final shapes obtained with the algorithm tend to be rough and difficult to manufacture. For this reason, the boundary of the final designs must be smoothed. Figure 5 shows the process to obtain the smoothed boundary of a given FEA mesh.

Results

Section "Benchmarking Cases" reports the accomplished results with the implemented meta-graph-based algorithm for different problems found in the literature. Section "Other Experiments" reports the results for other

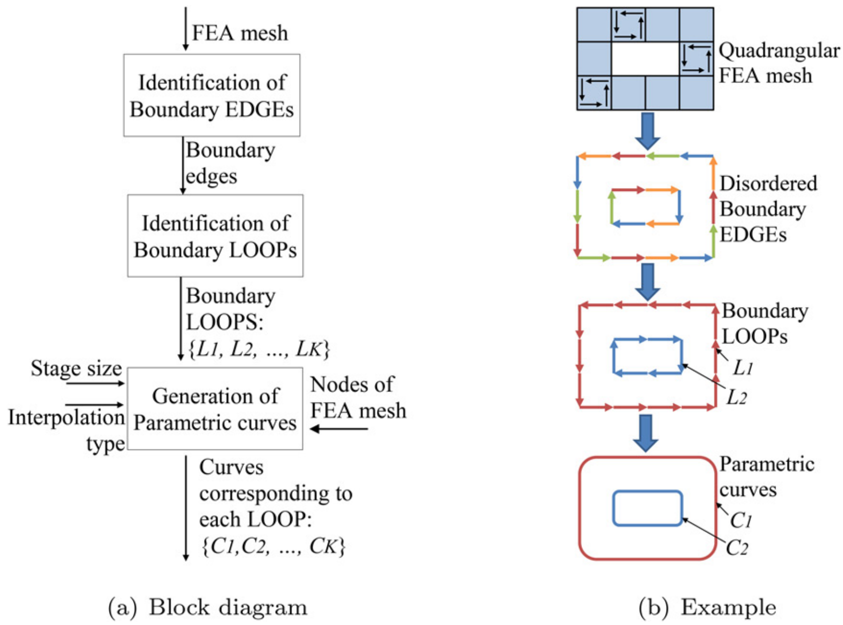


Figure 5. Post-processing for part boundary smoothing.

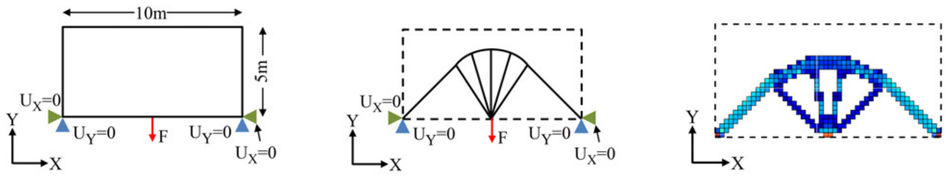
simulations that are useful to illustrate the behavior of the presented algorithm. In all the simulations is shown, at a particular iteration, how the meta-graph approach is applied to keep the domain connected. Section "Computational Demands of the Proposed Algorithm" describes the computational demands of the algorithm, and Section "Boundary Synthesis" presents the results of the smoothing of an FEA mesh. The extension to 3 D domains is discussed in Section "Extension to 3D Domains".

Benchmarking Cases

Michell Structure

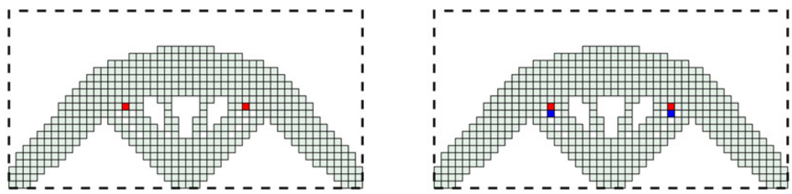
The design problem and theoretical solution of a Michell structure are depicted in Figures 6a and b. The solution obtained with the meta-graph strategy is shown in Figure 6c. In Figures 8a–d, shape evolution can be seen at different iterations. These figures show how the shape evolved until reaching a design that resembles the theoretical solution.

Figures 7a–d show the action of the meta-graph strategy at an intermediate iteration of the optimization process. Figure 7a shows the candidate elements for deletion, and Figure 7b depicts the meta-nodes associated to the configuration in Figure 7a. In Figure 7c, the meta-graph abstraction can be seen for this particular iteration. Based on the algorithm presented in Section "Element Deletion Algorithm", the meta-nodes C_2, C_3, C_4, and C_5 would be removed, and the resultant shape after elimination is shown in Figure 7d.

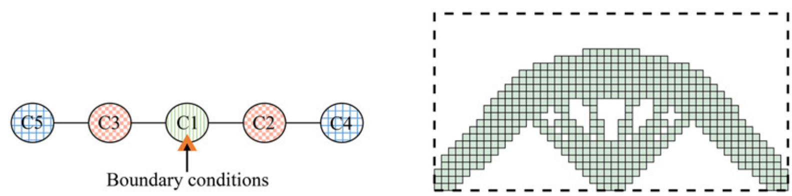


(a) Boundary conditions (b) Theoretical optimization (Hemp 1973) (c) Solution using the meta-graph procedure

Figure 6. Michell structure. Design domain and benchmarking solution.

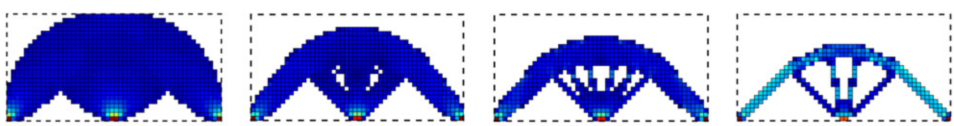


(a) Elements candidate for elimination: red (b) Meta-nodes: (i) C1: apple green region. (ii) C2, C3: red. (iii) C4, C5: blue



(c) Meta-graph generated from Fig. 7(b) (d) Surviving meta-node C1 after elimination of meta-nodes C2, C3, C4, C5

Figure 7. Michell structure. Intermediate iteration of the problem in Figure 6a. Elimination of under-stressed meta-nodes lead to disconnection-based elimination (Case 2).

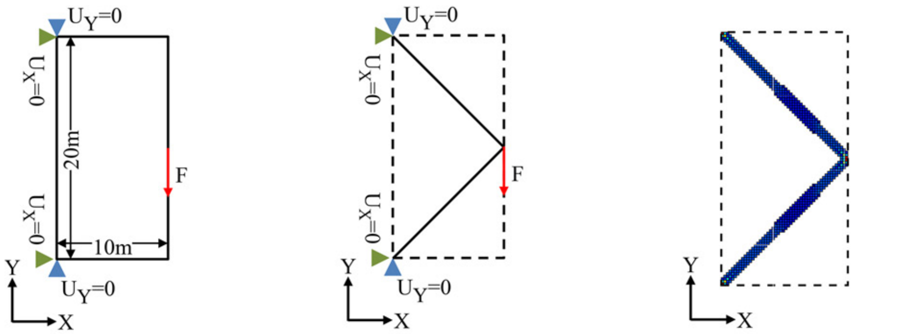


(a) Iteration 40 (b) Iteration 120 (c) Iteration 160 (d) Iteration 240

Figure 8. Michell structure. Evolution dictated by the meta-graph strategy of the problem in Figure 6a. Dotted line denotes the initial design domain.

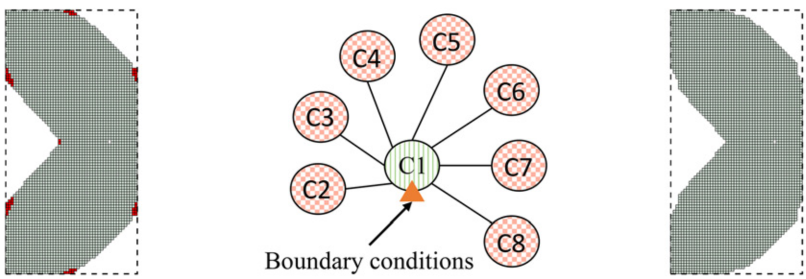
Two Bar Frame

The initial domain and boundary conditions for the design of a two-bar frame are shown in Figure 9a. The theoretical and experimental solutions are illustrated in Figures 9b and c. The evolution of the shape throughout the optimization is shown in Figures 11a–d. As in the Michell structure example, it can be seen that the design obtained with the meta-graph abstraction is similar to the theoretical solution.



(a) Boundary conditions (b) Theoretical solution (Hemp 1973) (c) Solution using the meta-graph procedure

Figure 9. Two bar frame. Design domain and benchmarking solution.



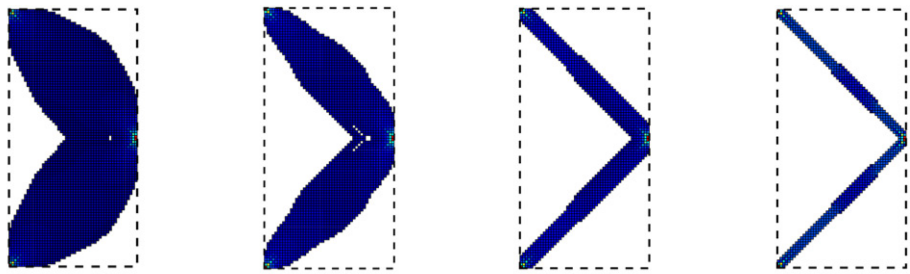
(a) Elements candidate for elimination: red. Meta-nodes: (i) C1: apple green region. (ii) C2, C3, C4, C5, C6, C7, C8: red (b) Meta-graph generated from Fig. 10(a) (c) Surviving meta-node C1 after elimination of meta-nodes C2, C3, C4, C5, C6, C7, C8

Figure 10. Two bar frame. Intermediate iteration of the problem in Figure 9a. Elimination of under-stressed meta-nodes lead to connected domain (Case 1).

The role of the meta-graph modeling in the optimization process is shown in Figure 10. Figure 10a shows the candidate elements for deletion. This is an example of the *Case 1* described in Section "Meta-graph Pruning", since all the noncandidate elements for deletion lie in the same meta-node, as shown in Figure 10b. Thus, all the meta-nodes different to C_1 can be deleted, and the resultant shape is as seen in Figure 10c.

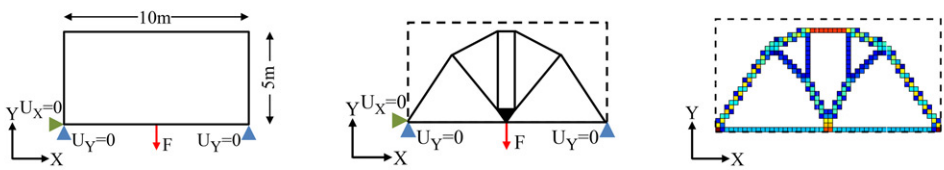
Michell Structure with Alternative Boundary Conditions

The initial domain and boundary conditions for a Michell structure are shown in Figure 12a. In comparison with the first example, the node in the lower-right corner does not have restriction of movement in X direction. Figure 12b exhibits a solution obtained via simulation by Xie and Steven (1993), and Figure 12c shows the solution given by the implemented algorithm. Figures 14a–d show the shape of the domain at different iterations.



(a) Iteration 10 (b) Iteration 50 (c) Iteration 150 (d) Iteration 300

Figure 11. Two bar frame. Evolution dictated by the meta-graph strategy of the problem in Figure 9a. Dotted line denotes the initial design domain.



(a) Boundary conditions (b) Proposal by Xie and Steven (1993) (c) Solution using the meta-graph procedure

Figure 12. Michell structure with alternative boundary conditions. Design domain and benchmarking solution.

Notice that the final shape is similar to the one proposed by Xie and Steven (1993).

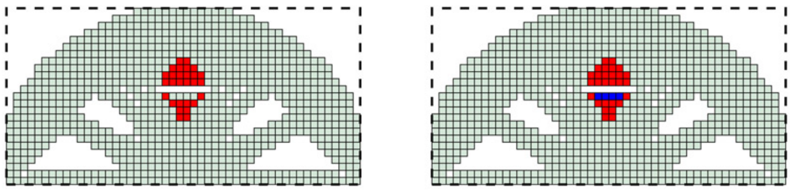
Figure 13 shows the performance of the meta-graph-based algorithm at an intermediate iteration. As in the previous examples, Figures 13a–c show the tentative element for deletion and the meta-graph associated to the current iteration. This is an instance of the *Case 2* presented in Section "Meta-graph Pruning". Figure 13d shows the resultant domain after the application of the corresponding meta-graph pruning strategy.

Other Experiments

To further illustrate the behavior of the algorithm, three additional simulations were executed. Given the absence of analytic solutions for the additional examples, the performance of the algorithm is tested using the software ANSYS® Academic Student, Release 19.0.

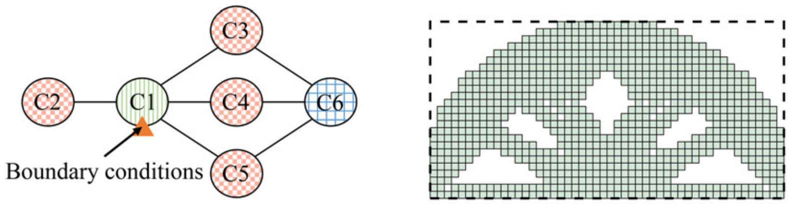
Bar Under Opposite Loads

Figure 15a shows the design domain and the load conditions for a bar subjected to loads of the same magnitude but in opposite directions. This simulation aims to exhibit a clear example of the behavior of the meta-



(a) Elements candidate for elimination: red

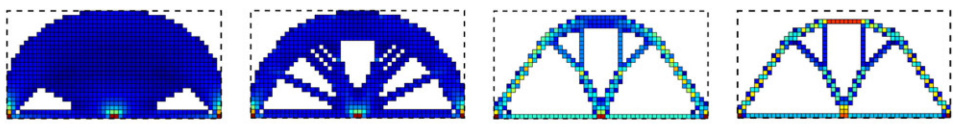
(b) Meta-nodes: (i) C1: apple green region. (ii) C2, C3, C4, C5: red. (iii) C6: blue



(c) Meta-graph generated from Fig. 13(b)

(d) Surviving meta-node C1 after elimination of meta-nodes C2, C3, C4, C5, C6

Figure 13. Michell structure with alternative boundary conditions. Intermediate iteration of the problem in Figure 12a. Elimination of under-stressed meta-nodes lead to disconnection-based elimination (Case 2).



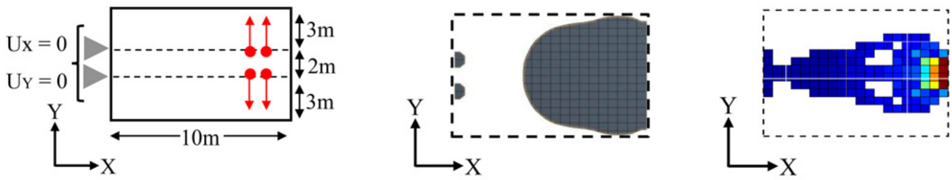
(a) Iteration 40 (b) Iteration 70 (c) Iteration 120 (d) Iteration 135

Figure 14. Michell structure with alternative boundary conditions. Evolution dictated by the meta-graph strategy of the problem in Figure 12a. Dotted line denotes the initial design domain.

graph approach in the *Case 3* (see Section "Meta-graph Pruning"). Figure 15b shows the solution obtained using the topology optimization module of ANSYS®, and Figure 15c depicts the solution obtained using the meta-graph strategy.

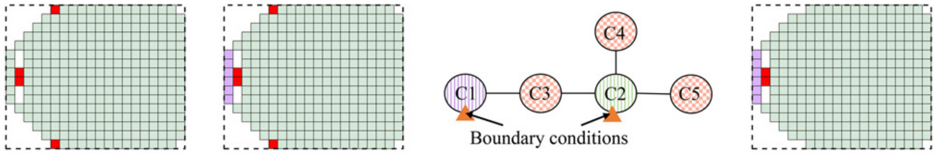
Figures 16a and b show the tentative elements to be deleted and the corresponding meta-nodes. In Figure 16c, where the meta-graph is depicted, is clear that the elements with boundary conditions are not in the same component, and the deletion of meta-node C_3 would lead a nonconnected domain. Thus, following the approach presented in Section "Meta-graph Pruning", only the meta-nodes C_4 and C_5 can be removed. Figure 16d exhibits the resultant shape.

Figure 17 shows the evolution of the shape during the optimization process. As opposed to the solution proposed by ANSYS® (Figure 15b), the solution obtained by the implemented algorithm remains connected during all the optimization process.



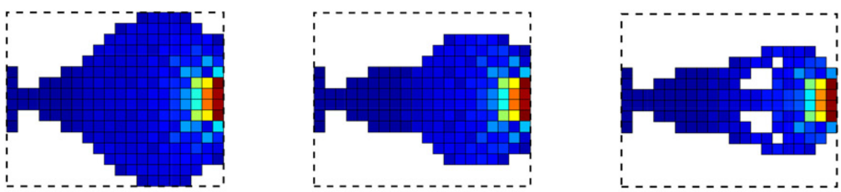
(a) Boundary conditions (b) ANSYS® solution (c) Solution using the meta-graph procedure

Figure 15. Bar under opposite loads. Design domain and ANSYS® solution.



(a) Elements candidate for elimination: red (b) Meta-nodes: (i) C1: magenta region. (ii) C2: apple green region. (iii) C3 - C5: red (c) Meta-graph generated from Fig. 16(b) (d) Surviving meta-nodes C1, C2, C3 after elimination of meta-nodes C4, C5

Figure 16. Bar under opposite loads. Intermediate iteration of the problem in Figure 15a. Elimination of under-stressed meta-nodes lead to disconnection-based elimination (Case 3).



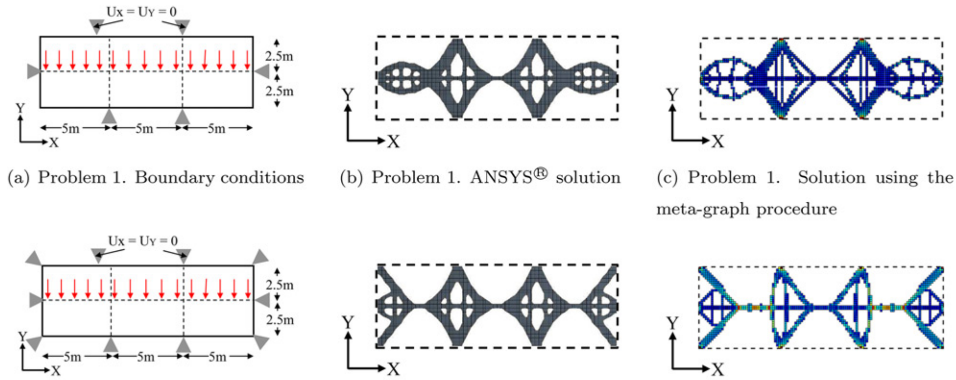
(a) Iteration 20 (b) Iteration 30 (c) Iteration 40

Figure 17. Bar under opposite loads. Evolution dictated by the meta-graph strategy of the problem in Figure 15a. Dotted line denotes the initial design domain.

Viaduct Simulations

In order to show the capacity of the implemented algorithm to replicate to some extent some well-known engineering structures, two different simulations were executed. They try to resemble the aspect of a viaduct. Figures 18a and d show the design domains and the load conditions for the two simulations. In both cases, a distributed load along the length of the domain is applied. Figures 18b and e show the solution retrieved by ANSYS® for each load case. Likewise, Figures 18c and f depict the solution given by the implemented algorithm.

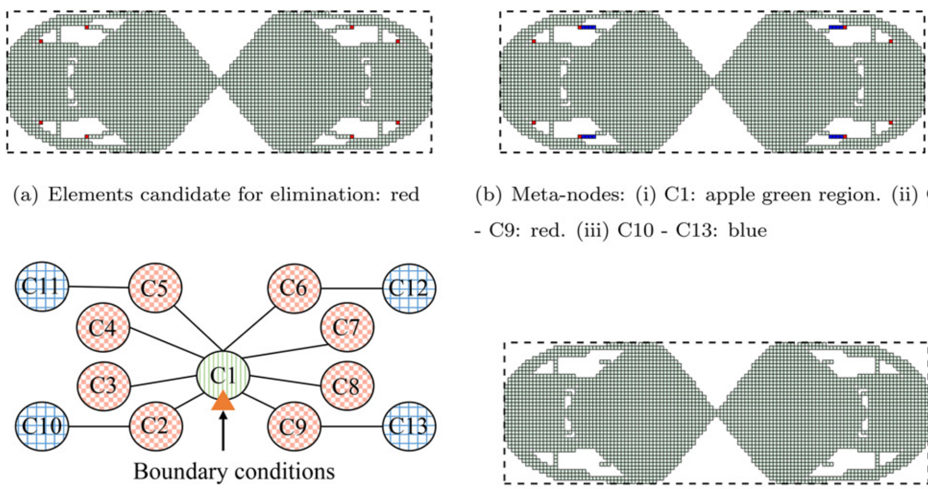
Figures 19 and 21 show how the meta-graph abstraction was used and the resultant domain for a particular iteration in each simulation. Figures 20 and 22 show the evolution of the shape throughout the optimization. It can be seen that the obtained designs are similar to the solutions given by ANSYS®.



(a) Problem 1. Boundary conditions (b) Problem 1. ANSYS® solution (c) Problem 1. Solution using the meta-graph procedure

(d) Problem 2. Boundary conditions (e) Problem 2. ANSYS® solution (f) Problem 2. Solution using the meta-graph procedure

Figure 18. Viaduct simulations. Design domains and ANSYS® solutions.

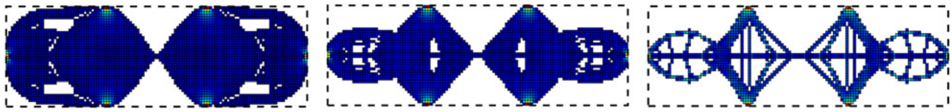


(a) Elements candidate for elimination: red (b) Meta-nodes: (i) C1: apple green region. (ii) C2 - C9: red. (iii) C10 - C13: blue

(c) Meta-graph generated from Fig. 20(b) (d) Surviving meta-node C1 after elimination of meta-nodes C2 - C13

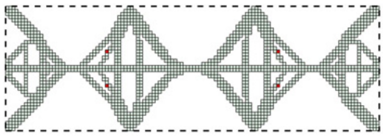
Figure 19. Bridge 1. Intermediate iteration of the problem in Figure 18a. Elimination of under-stressed meta-nodes lead to disconnection-based elimination (Case 2).

The obtained designs resemble to some extent the shape of a viaduct of the type of the Millau Viaduct, France (Figure 23). Notice that the links under the actual bridge are missing (compared with the evolution shown in Figures 20c and 22c), due to the fact that such links, in the engineering reality, do not stand compression. They only stand tension. A future necessary improvement for the method presented in this work would include the consideration of such anisotropic effects.

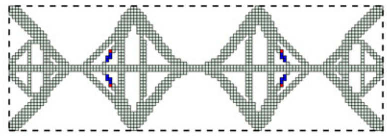


(a) Iteration 30 (b) Iteration 60 (c) Iteration 129

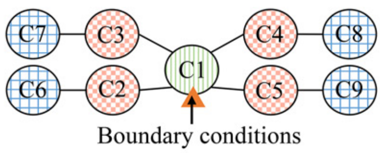
Figure 20. Bridge 2. Intermediate iteration of the problem in Figure 18d. Elimination of under-stressed meta-nodes lead to disconnection-based elimination (Case 2).



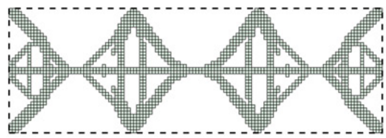
(a) Elements candidate for elimination: red



(b) Meta-nodes: (i) C1: apple green region. (ii) C2 - C5: red. (iii) C6 - C9: blue



(c) Meta-graph generated from Fig. 22(b)



(d) Surviving meta-node C1 after elimination of meta-nodes C2 - C9

Figure 21. Bridge 1. Evolution dictated by the meta-graph strategy of the problem in Figure 18a. Dotted line denotes the initial design domain.

Computational Demands of the Proposed Algorithm

Due to the material removal procedure associated to the presented algorithm, given the mesh $M_i = (N_i, E_i)$ at iteration i can be stated: $|N_0| \geq |N_i|$ and $|E_0| \geq |E_i|$, where $M_0 = (N_0, E_0)$ is the initial mesh. In addition, $|N_0| > |E_0|$. Therefore, the computational demands of one iteration of the algorithm can be expressed as a function of N_0 and the bandwidth W of the stiffness matrix calculated during the FEA simulation (Farmaga et al. 2011).

Table 1 presents the computational expenses of the implemented algorithm. Notice that the time complexity and memory complexity of an iteration of the algorithm is dictated by the term $O(N_0^2)$. This term corresponds to the dominant generation of the graph and the meta-graph associated to the FEA mesh.

A comparison of the computational resources used vs. the efficiency of evolution is beyond the capabilities of the present work. One reason for this limitation is that the measure of the quality or efficiency of an evolution is itself an open research question at this time.

Boundary Synthesis

The simulations carried out with the meta-graph-based algorithm showed that the resultant shape is very rough. Following the procedure described

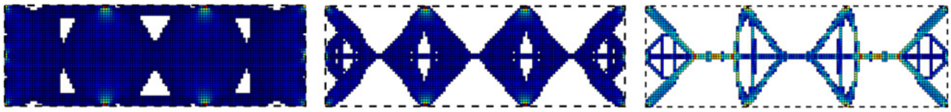


Figure 22. Bridge 2. Evolution dictated by the meta-graph strategy of the problem in Figure 18d. Dotted line denotes the initial design domain.

Figure 23. Millau Viaduct, France. Image by MIEL 1971(2015). Licensed by CC0 (Creative Commons Zero).

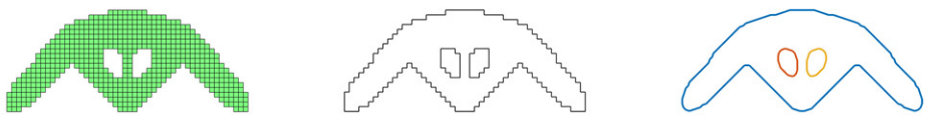
Table 1. Analysis of the computational costs of an iteration of the proposed algorithm.

Process	Time expenses	Memory expenses
FEA Simulation (Farmaga et al. 2011)	$O(N_0W^2)$	$O(N_0W)$
Mesh Partition	$O(N_0)$	$O(N_0)$
Element Deletion	$O(N_0^2)$	$O(N_0^2)$
Graph Generation	$O(N_0^2)$	$O(N_0^2)$
Graph Components Calculation	$O(N_0)$	$O(N_0)$
Meta-graph Generation	$O(N_0^2)$	$O(N_0^2)$
Meta-graph Pruning	$O(N_0)$	$O(N_0)$

in Section “Methodology”, one of the designs obtained with the algorithm was smoothed. The results of the application of the smoothing algorithm using a mean filter of order 4 are shown in Figure 24. Notice that the boundary of the shape was corrected without losing sensitive information of the design.

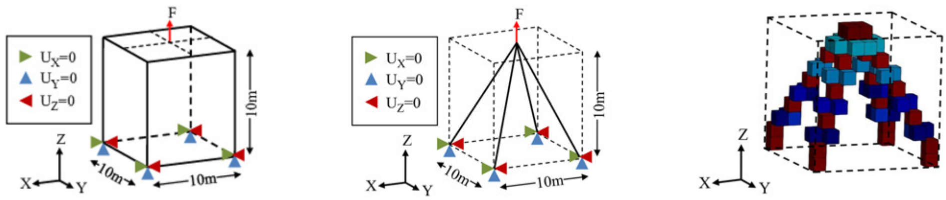
Extension to 3 D Domains

In this section, this article aims to show that the algorithm presented in this work can be applied in 3 D domains. Since the generation of the meta-graph is not constrained to any special characteristic of the 2 D domains,



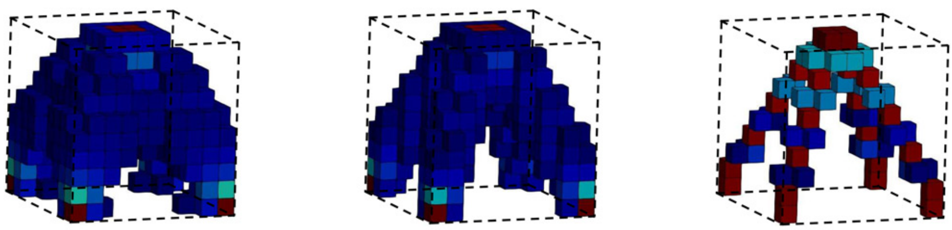
(a) Result of meta-graph pruning (b) Structure boundary (c) Smoothed boundary

Figure 24. Michell structure. Post-processing for border smoothing.



(a) Boundary conditions (b) Theoretical solution (Abolbashari and Keshavarzmanesh 2006) (c) Solution using the meta-graph procedure

Figure 25. 3D benchmarking test (Abolbashari and Keshavarzmanesh 2006). Design domain and solution.



(a) Iteration 15 (b) Iteration 30 (c) Iteration 47

Figure 26. 3D benchmarking test (Abolbashari and Keshavarzmanesh 2006). Evolution dictated by the meta-graph strategy of the problem in Figure 25a. Dotted line denotes the initial design domain.

the most sensitive change when working in 3 D is the generation of the graph associated to the 3 D FEA mesh.

For 3 D FEA meshes, multiple adjacency relations can be defined between the elements. The following simulation is based on a face adjacency relation between the FEA elements. Figure 25a shows the initial 3 D domain and boundary conditions. Figure 25b depicts the solution proposed by Abolbashari and Keshavarzmanesh (2006) and Figure 25c exhibit the solution obtained using the meta-graph strategy. Figures 26a–c present the evolution of the domain using the meta-graph strategy. It can be seen that the shape of the domain evolves toward the optimal shape in Figure 25b.

Conclusions

This article presents the implementation of a novel methodology for topology optimization. This methodology joints the concepts of ESO (a well-known optimization algorithm) with mathematical graph modeling of the FEA mesh. The implemented methodology is based on a meta-graph connectivity abstraction of the FEA mesh. This method progressively removes the under-demanded material of the structure while maintaining a feasible solution at every iteration.

The meta-graph approach allows a unified treatment of neighborhood and static connectedness of the evolving shape. Seven different examples, with both 2 D and 3 D domains, are presented to clearly illustrate the meta-graph method.

The examples discussed in this manuscript are in the field of linear solid mechanics. However, the implemented algorithm can be adapted to manage other stimuli sources, other than forces, pressures or torques. The current work requires that the effects of the stimuli be described as a scalar field $f()$ acting on the finite elements. Research opportunities are open for (e.g.) friction or abrasive stimuli and for domains that could grow (not only retract as in the present approach).

Limitations and Shortcomings

In the case in which elements with boundary conditions do not lie in the same meta-node, some meta-nodes of degree 2 (or more) could be deleted and the optimization process must be divided in multiple branches. However, since only meta-nodes of degree 2 are deleted, all the other branches are not considered.

Future Work

Future work should address the utilization of stimuli other than stress/strain: friction, abrasion, heat, humidity, etc. Future research should also address the integration of the algorithm with other evolutionary based techniques (e.g., mutation) to allow the generation of multiple feasible solutions that explore a wider region of the solution space.

Glossary

AM	Additive manufacturing
BESO	Bidirectional evolutionary structural optimization
ESO	Evolutionary structural optimization
FEA	Finite element analysis
GA	Genetic algorithms
	Ω_0 Compact and bounded subset of $\mathbb{R}^2$, that represents an initial material stock from which to carve the part
Ω_i	The part after the i-th step of the evolution

f Scalar function $f : \Omega_i \rightarrow \mathbb{R}$ that expresses how much is the neighborhood of a point $x \in \Omega_i$ being demanded by the stimuli (e.g. stress) being considered

g Scalar function $g : \Omega_i \rightarrow \mathbb{R}$ that expresses the permissible level of demand f that the neighborhood of a point $x \in \Omega_i$ may stand (e.g. permissible stress allowable). In mechanical design, g is usually a constant for the whole domain $\Omega i G = (V, A)$ the Finite Element- based graph in which a vertex $v_i \in V$ is a finite element. An arc $(v_i, v_j) \in A$ means that finite elements vi and vj are neighbors $G_M = (V_M, A_M)$A meta-graph built on G, in which a meta-vertex $V_i \in V_M$ is a connected set of finite elements of V. A meta - arc $(V_i, V_j) \in A_M$ means that meta-vertices Vi and Vj are neighbors

ORCID

Diego Montoya-Zapata http://orcid.org/0000-0001-5521-1794
Diego A. Acosta http://orcid.org/0000-0001-8654-6116
Oscar Ruiz-Salguero http://orcid.org/0000-0002-9674-8974
Jorge Posada http://orcid.org/0000-0001-7985-9915

References

Abolbashari, M. H., and S. Keshavarzmanesh. 2006. On various aspects of application of the evolutionary structural optimization method for 2D and 3D continuum structures. *Finite Elements in Analysis and Design* 42 (6):478–91. doi: 10.1016/j.finel.2005.09.004.

Chen, J., R. Ahmad, H. Suenaga, W. Li, K. Sasaki, M. Swain, and Q. Li. 2015. Shape optimization for additive manufacturing of removable partial dentures – A new paradigm for prosthetic CAD/CAM. *PLos One* 10 (7):1–17. doi:10.1371/journal.pone.0132552.

Chen, Y., M. Schellekens, S. Zhou, J. Cadman, W. Li, R. Appleyard, and Q. Li. 2011. Design optimization of Scaffold microstructures using wall shear stress criterion towards regulated flow-induced erosion. *Journal of Biomechanical Engineering* 133 (8): 081008–448. doi: 10.1115/1.4004918.

Da, D., L. Xia, G. Li, and X. Huang. 2018. Evolutionary topology optimization of continuum structures with smooth boundary representation. *Structural and Multidisciplinary Optimization* 57 (6):2143–59. doi: 10.1007/s00158-017-1846-6.

Das, R., and R. Jones. 2011. Topology optimisation of a bulkhead component used in aircrafts using an evolutionary algorithm. 11th International Conference on the Mechanical Behavior of Materials (ICM11). *Procedia Engineering* 10:2867–72. doi: 10.1016/j.proeng.2011.04.476.

Deaton, J. D., and R. V. Grandhi. 2014. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization* 49 (1):1–38. doi: 10.1007/s00158-013-0956-z.

Farmaga, I., P. Shmigelskyi, P. Spiewak, and L. Ciupinski. 2011. Evaluation of computational complexity of finite element analysis. In *2011 11th International Conference the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, 213–214. February. [CrossRef]

Ghabraie, K. 2015. An improved soft-kill BESO algorithm for optimal distribution of single or multiple material phases. *Structural and Multidisciplinary Optimization* 52 (4):773–90. doi: 10.1007/s00158-015-1268-2.

Giger, M., and P. Ermanni. 2006. Evolutionary truss topology optimization using a graph-based parameterization concept. *Structural and Multidisciplinary Optimization* 32 (4): 313–26. doi: 10.1007/s00158-006-0028-8.

Hemp, W. S. 1973. *Optimum structures (Oxford Engineering Science Series)*. 1st ed. Oxford: Oxford University Press.

Huang, X., Y. M. Xie, B. Jia, Q. Li, and S. W. Zhou. 2012. Evolutionary topology optimization of periodic composites for extremal magnetic permeability and electrical permittivity. *Structural and Multidisciplinary Optimization* 46 (3):385–98. doi: 10.1007/s00158-012-0766-8.

Madeira, J. F. A., H. L. Pina, and H. C. Rodrigues. 2010. GA topology optimization using random keys for tree encoding of structures. *Structural and Multidisciplinary Optimization* 40 (1–6):227. doi: 10.1007/s00158-008-0353-1.

Montoya-Zapata, D., D. A. Acosta, O. Ruiz-Salguero, and D. Sanchez-Londono. 2019. "FEA structural optimization based on metagraphs. In *International Joint Conference SOCO'18-CISIS'18-ICEUTE'18*, ed. M. Graña, J. M. López-Guede, O. Etxaniz, Á. Herrero, J. A. Sáez, H. Quintián, and E. Corchado, 209–220. Cham: Springer International Publishing.

Munk, D. J., G. A. Vio, and G. P. Steven. 2017. A Bi-directional evolutionary structural optimisation algorithm with an added connectivity constraint. *Finite Elements in Analysis and Design* 131:25–42. doi: 10.1016/j.finel.2017.03.005.

Munk, D. J., G. A. Vio, and G. P. Steven. 2015. Topology and shape optimization methods using evolutionary algorithms: A review. *Structural and Multidisciplinary Optimization* 52 (3): 613–31. doi:10.1007/s00158-015-1261-9.

Querin, O. M., G. P. Steven, and Y. M. Xie. 1998. Evolutionary structural optimisation (ESO) using a bidirectional algorithm. *Engineering Computations* 15 (8):1031–48. doi: 10.1108/0264440 9810244129. doi: 10.1108/02644409810244129.

Seifi, H., A. Rezaee Javan, S. Xu, Y. Zhao, and Y. M. Xie. 2018. Design optimization and additive manufacturing of nodes in gridshell structures. *Engineering Structures* 160: 161–70. doi: 10.1016/j.engstruct.2018.01.036.

Stojanov, D., B. G. Falzon, X. Wu, and W. Yan. 2016. Implementing a structural continuity constraint and a halting method for the topology optimization of energy absorbers. *Structural and Multidisciplinary Optimization* 54 (3):429–48. doi: 10.1007/s00158-016-1451-0.

Tang, Y., G. Dong, Q. Zhou, and Y. F. Zhao. 2018. Lattice structure design and optimization with additive manufacturing constraints. *IEEE Transactions on Automation Science and Engineering* 15 (4):1546–62. doi: 10.1109/TASE.2017.2685643.

MIEL1971. 2015. *Millau Viaduct*. Accessed August 30, 2018. https://pixabay.com/es/viaducto-millau-puente-francia-2002206/.

Xia, L., Q. Xia, X. Huang, and Y. M. Xie. 2018. Bi-directional evolutionary structural optimization on advanced structures and materials: A comprehensive review. *Archives of Computational Methods in Engineering* 25 (2):437–78. doi: 10.1007/s11831-016-9203-2.

Xie, Y. M., and G. P. Steven. 1993. A simple evolutionary procedure for structural optimization. *Computers & Structures* 49 (5):885–96. doi: 10.1016/0045-7949(93)90035-C.