



Extending the usability of the force-flow based topology optimization to the process of generative design

Márton Tamás Birosz · János György Bátorfi · Mátyás Andó

Received: 4 October 2022 / Accepted: 22 January 2023 / Published online: 14 February 2023
© The Author(s) 2023

Abstract Topology optimization and Generative Design are two methods to create volume or stiffness optimized parts and they are used more frequently nowadays. However, the specific methods are often not well described and the connection between these two is not clearly explained. In this article, a force-flow based topology optimization process has been explained in detail and extended with a function to be able to use as a Generative Design tool. The proposed algorithm has been tested on three 2D shapes and the effectiveness was evaluated. This work clarifies the vague description of theoretical solutions by presenting in detail the operation of the algorithm and bridging the lack of information that exists between the shape or topology optimization procedure and the generative design solution.

Keywords Optimization algorithm · Topology optimization · Generative design · Force-flow · Load-path

1 Introduction

As modern design software and optimization techniques make it possible, a completely new and

extremely user-friendly solution for the designers of components is made. Typically, the previous practice was that to fulfill a given function, the engineers created a simplified geometry that met the desired goal based on strength, thermal and other calculations, and then this created form was modified if needed e.g. it could be manufactured with a specific technology. Partly because the traditional manufacturing processes were limited in terms of the geometry that could be created, and partly because once the first version of the product that meets the criteria was designed, no further changes were usually made. However, the result was often an over-defined robust piece that locally contained redundant material, thus having a disproportionately large mass and volume. In some cases, it could be explained that the extra manufacturing process itself for separating the unnecessary parts from the raw material would be a higher cost than the material itself. However, this problem has also been solved with modern multi-axis cutting machines, electric discharge machining, or even 3D printing (working on the additive principle).

Another problem was that although the boundary conditions were known at the time of planning and where the significant stresses arise, they could change in an unknown way by removing the less load-bearing areas. There was no suitable method to find out where and how much material had to be removed, so it was more or less random, and in every case, checking calculations were needed. In other words, initially, these optimization methods were mainly about

M. T. Birosz (✉) · J. G. Bátorfi · M. Andó
Savaria Institute of Technology, Faculty of Informatics,
Eötvös Loránd University, Budapest, Hungary
e-mail: birosz.marton@gmail.com

size optimization, where the conditions were known and there was a starting geometry that did not change during the optimization, only the parameters of the geometry (size of the trusses or thickness of a plate). However, what was needed was shape or topology optimization (TO). In the case of shape optimization, the initial geometry is also relevant, but here the location and shape of the nodes, trusses, and holes can change. In the case of topology optimization, the main question is how many holes or material gaps are needed, where they are located, and how the parts of the component are connected [1].

Over the years, several TO procedures have been developed [2, 3], the two most popular are the Solid Isotropic Material with Penalization (SIMP) and the Evolutionary Structural Optimization (ESO). However, almost all procedures can be summed up by the fact that the material must remain where it is needed and must be removed from where it is not needed. Especially for structural optimization, the techniques based on the Force-flow are more prominent [4]. One method that belongs to this is the determination of Principal Stress Lines (PSL) [5–7]. These lines are pairs of orthogonal directions on each node, that are connected to define a trajectory where the material should be placed to get better mechanical resistance. For example, Li et al. [8] used PSL to create the mesoscale inside structure of 3D printed parts. This led to higher mechanical resistance compared to the conventional infill patterns. Also, PSL is optimal for generating truss structures but has a lack of standardization and parameterization. Another method is load path-based optimization. Load paths are defined as curves those transfer a constant load and can be calculated by the sequential connection of the stress pointing vector [9–11]. The representation of these trajectories was presented by Kelly et al. [12]. In their work, they stated that load paths can be defined by plotting contours aligned with total stress “pointing” vectors given by the columns of the stress matrix.

Complementing the use of TO, a new design solution called Generative Design (GD) has been developed. GD uses a presented TO method to create a structure that has a minimum weight or maximal stiffness set by the user as a preference while meeting some other restrictions. Compared to a “traditional” method here no initial geometry is imported. GD creates the most appropriate geometry in the available area in the space. As Fisher and Herr defined

it: “during the design process the designer does not interact with materials and products in a direct way, but via a generative system of some sort.” [13]. The input parameters are the objectives (mass/stiffness), geometry (preserve regions and obstacle regions), load cases (constraints and loads), material, and manufacturing constraints [14]. GD has got a small relevance for many years, but recently it became a popular topic in both the industry and academia. However, almost the only commercially widely available software is the Autodesk generative design tool. The methods by which GD creates the geometry can be various, for example, stochastic search, L-system, cellular automata, genetic algorithms, and evolutionary design [15–17].

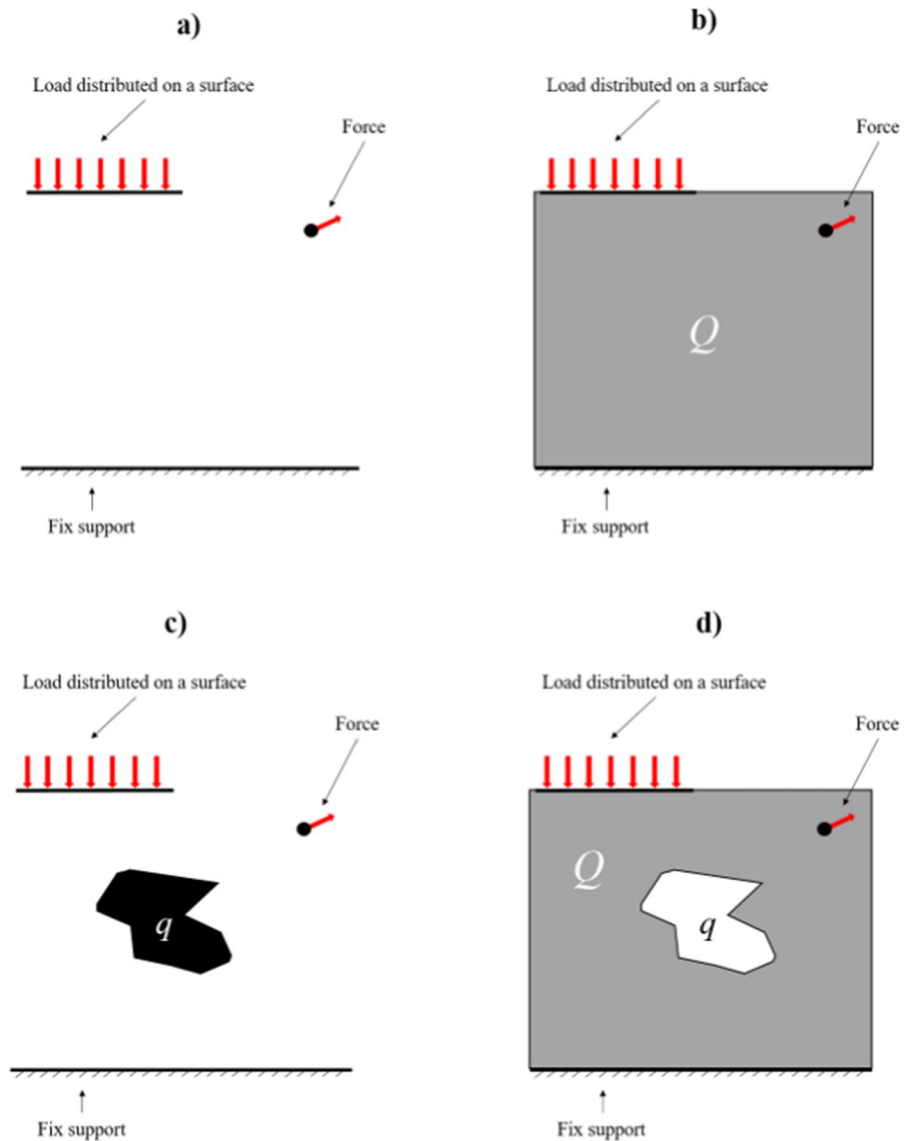
Since both topics, TO and GD are widely known, the detailed description of their operation is often vague. In this paper, we present a method that extends a TO process (load-path method) usability with in a GD process. As a set of goals, we created three guidance: (1) Describe the process of load-path based TO highlighting all the necessary steps and its difficulties and limitations, (2) Connecting TO with GD, and describing its theorem, (3) Prove the effectiveness of the results.

2 Methodology

The concept of the proposed solution has two main parts; the first one is to create the initial structure (GD responsibility) for topology optimization, and the second one is the topology optimization itself. The initial condition of the process is to start with an initial geometry that satisfies the boundary conditions but clearly has the lowest efficiency. In this case, the essence of the whole process is to use the smallest possible amount of material, while the stiffness of the part is still acceptable. Therefore, the original geometry should have the largest possible volume.

The representation of the geometry creation can be seen in Fig. 1. Typically, when there is a need to design a new part, the first known condition is the locations of the connecting surfaces, edges, or points and the locations and magnitude of the estimated loads that will be applied (Fig. 1a, b). Thus, the original geometry must have an area large enough to include all such locations. The secondary, but equivalent, the requirement is that there should

Fig. 1 Initial/original geometry determination; **a** constraints, **b** the biggest simplified design domain which includes all the constraints, **c** constraints and a restricted area and **d** the biggest simplified design domain which includes all the constraints and excludes the restricted area



be voids in the structure in specific areas where material cannot be due to some other function or component (Fig. 1c). These voids must be preserved later during the optimization, so as a principle it is always removed from the previously defined inclusion area. The boundary conditions can be summarised by Eqs. 1 and 2. Equation 1 prescribes the displacements values for a surface of the investigated geometry as fixed support. It means that all the directional displacements are equal to zero. Equation 2. describes the force boundary condition for a surface area as prescribed forces acting on specified nodes.

$$u = 0 : S \subset Q_{Bu} \tag{1}$$

$$F = F_B : S \subset Q_{Bp} \tag{2}$$

The second part is the topology optimization itself. As mentioned earlier, this part of the process logically follows the principle developed by Kelly et al. [12] with some changes so that generative design can be implemented. The theory presented in this article is that non-significant regions of material should be truncated from the generated initial

geometry. Thus, after the initial geometry creation, the procedure is simplified to a topology optimization task. Therefore, compared to other conventional GD solutions, there will be only one result at the end of the iteration optimization algorithm. After it, a Finite Element Analysis (FEA) has been done, so the stress tensor of each node is known (3).

$$[\sigma] = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \quad (3)$$

where σ_{ij} is the shear on the plane of an element unit and σ_{ii} is the normal stress. This 3×3 matrix must be simplified into total stress “pointing” vectors from the columns of the stress matrix. With that, only three vectors will remain pointing in the adequate coordinate directions. These pointing vectors are defined as (4):

$$\begin{aligned} V_x &= \sigma_{xx}i + \sigma_{yx}j + \sigma_{zx}k \\ V_y &= \sigma_{xy}i + \sigma_{yy}j + \sigma_{zy}k \\ V_z &= \sigma_{xz}i + \sigma_{yz}j + \sigma_{zz}k \end{aligned} \quad (4)$$

The forces acting on the arbitrary plane with normal, as (5):

$$n = n_x i + n_y j + n_z k \quad (5)$$

And the magnitudes, as (6):

$$\begin{aligned} F_x &= \int V_x \cdot n dA \\ F_y &= \int V_y \cdot n dA \\ F_z &= \int V_z \cdot n dA \end{aligned} \quad (6)$$

After getting these vectors and the corresponding stress values, the load paths can be determined by selecting the pointing vector which has the most relevance. In other words, the load path for a force in a given direction is a region in which the force in that direction remains constant. By knowing the values of stress tensor elements for each mesh point, the equivalent stress can be calculated by various methods (von Mises and Mohr). In this study, we have applied the Mohr Theory, which is based on a geometrical-graphical model. After this calculation the characteristic directional force values are determined by Eq. 4., and the maximum of them is used for the following calculations. Based

on this maximal characteristic force, the critical or not-needed parts can be recognized, extended, or removed.

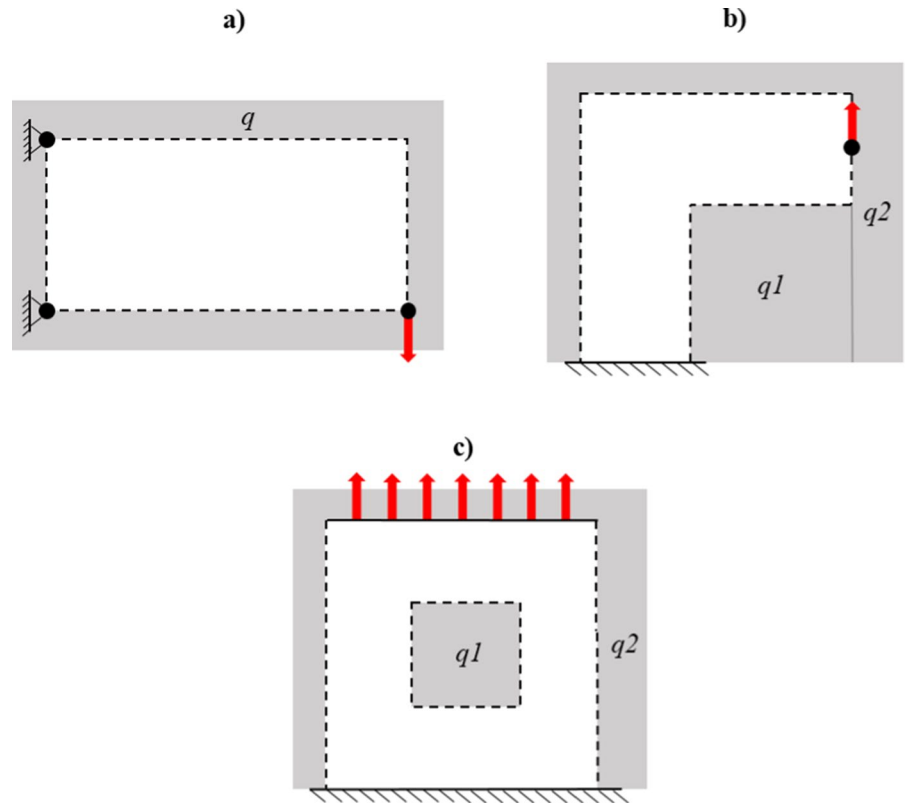
2.1 Test cases

To test the algorithm three 2D cases were investigated. Two of them are simplified geometries which frequently used by others [1] in order to compare the results. The first one is a beam with two fixed supports on the top and bottom left corners, and an applied load on the bottom right corner. In this case, there are no predefined voids, therefore, the initial geometry doesn't have any exclusion area. The second example is the most frequently tested “L shape” with fixed support on the edge of one of its truss's ends, and an applied load on a point at the other truss's end. The third one was a simple square with a cut out in the middle, the lower edge was a fixed support and a distributed load on the upper edge was applied. A more complex geometry has not been tested since the aim was to present the effectiveness of the proposed solution, and since in practice the designer also doesn't want to spend too much time on drawing the exclusion areas, they would simply put a square or circle at those regions where it is not allowed to have material. Basically, since the first consideration in the generated initial geometry is to include the specified boundary conditions locations, this only determines the minimum size. Without setting a maximum limit, an infinitely large area would result. Therefore, additional exclusion areas must be added as conditions that limit the maximum size. For all cases, it was set to get back the desired original geometries. In Fig. 2. the boundary conditions and exclusion areas are shown including the desired forms presented with dashed lines. As can be seen in Fig. 2b, c there are two exclusion areas, one belonging to geometry restriction ($q1$) and one for the external boundary ($q2$).

2.2 Proposed algorithm

The presented algorithm was created with MATLAB software, and with its partial differential equation (PDE) function, the input finite element simulation can be performed iteratively, so there is no need for a manual import-export process while the program is

Fig. 2 Test cases **a** beam, **b** L-shape, and **c** square with cut out



running. The optimization process can be divided into the following steps:

- (1) Based on the predefined boundary conditions and exclusion area(s) the starting geometry is created.
- (2) Perform a structural FEA.
- (3) From the simulation results calculate the pointing stress vectors, for later use, at each node only the vector with the highest associated magnitude was retained out of the three vectors.
- (4) Determine the V_{cut} point, and the elements with a value smaller than V_{cut} are deleted.
- (5) Scale the modulus of the elements according to (7).

$$E_i|_{new} = E_i|_{old} \frac{|V_i|}{|V|_{cut}} \tag{7}$$

if $E_i|_{new} < E_{min}$ then $E_i|_{new} = E_{min}$
 if $E_i|_{new} > E_{max}$ then $E_i|_{new} = E_{max}$

- (6) Smooth the geometry to remove the undesirable sharp edges and material defects.

- (7) With the obtained new geometry start a new FEA, and iteratively repeat the steps until the desired volume fraction is achieved (Fig. 3).

2.3 Explanation of the steps

- (1) Create the starting geometry: to simplify later calculations, the initial geometry is always rectangular (2D). To determine the load paths, interpolation of the pointing vectors to new nodes according to the elements of an $n \times m$ matrix is needed. Therefore, the inclusion area of the design domain is a rectangle that is big enough to contain the supports and loads, smaller than the external boundary, and the necessary exclusion areas are removed. The external boundary is defined as maximum and minimum extremums on each coordinate direction, and an exclusion area is a region where the nodes of the mesh are not evaluated.

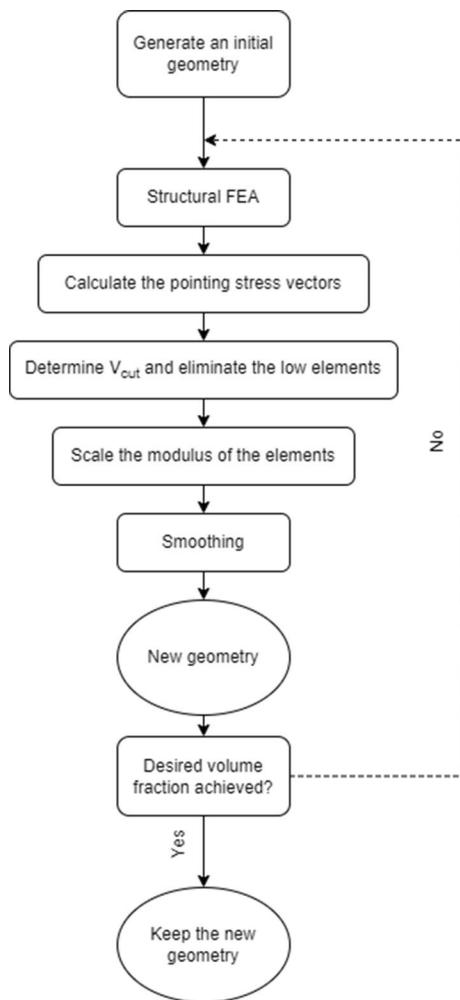


Fig. 3 Flowchart of the algorithm

- (2) During the FEA simulation the boundary conditions and exclusion areas are saved for the iteration process. Therefore, with the refreshed geometry they are automatically added to the simulation. Also, it helps to ensure, that these locations will not be eliminated or filled.
- (3) MATLAB uses tetrahedral linear mesh as default, and it cannot be changed to quadrilateral elements. Thus, the results and the calculated pointing vectors must be interpolated into the element locations of a square matrix. It makes the optimization process independent of the mesh shape and size. Also, every element will have the same size, which is necessary for optimization. Since the results are calculated for every interpolated

element of the generated matrix, those points where the elements were restricted by the exclusion or during the previous iteration were deleted must be deleted from the matrix as well.

- (4) At first all the pointing vectors V_i must be sorted in an ordered descending array. To obtain the value V_{cut} , an elimination factor Δ (0–1) must be determined before the optimization process, which reduces the volume of the geometry during each iteration. The value Δ cannot be too big, since it would lead to compliance problems or the dismemberment of the geometry. V_{cut} is the value of a member which is $\Delta \cdot \text{sum}(V)$, and the member with a smaller V is eliminated.
- (5) The scaling process goes according to the relations. After each iteration, the modulus matrix is refreshed. The modulus of the material was specified for E_{max} and zero for E_{min} .
- (6) The smoothing is necessary to remove any checkerboard pattern, smooth the sharp edges, and overcome the problems arising from the numerical solution. The smoothing is applied on each node, the algorithm checks whether the average of neighboring nodes is greater or equal to $E_{max}/2$. If it is greater then modify the node's modulus to E_{max} , if smaller then modify to E_{min} .
- (7) From the matrix with the modified modulus values for each node respectively the algorithm creates a new geometry, where the material is only added at the locations where the modulus equals E_{max} . Since the elimination factor is set relatively low to aid the better calculation, another factor β was set at the beginning of the iteration process. This β factor specifies the ratio of how much volume reduction to the starting geometry the algorithm will continue.

3 Results and discussion

In Fig. 4 the first cycle FEA results can be seen. It can be stated that as was expected based on the boundary condition and the set exclusion area the generated initial geometries are the beam, L-shape, and square with cut out. From the point of view of the subsequent optimization process, the values of the emerged stresses are not important, but rather their location and current ratio within the geometry.

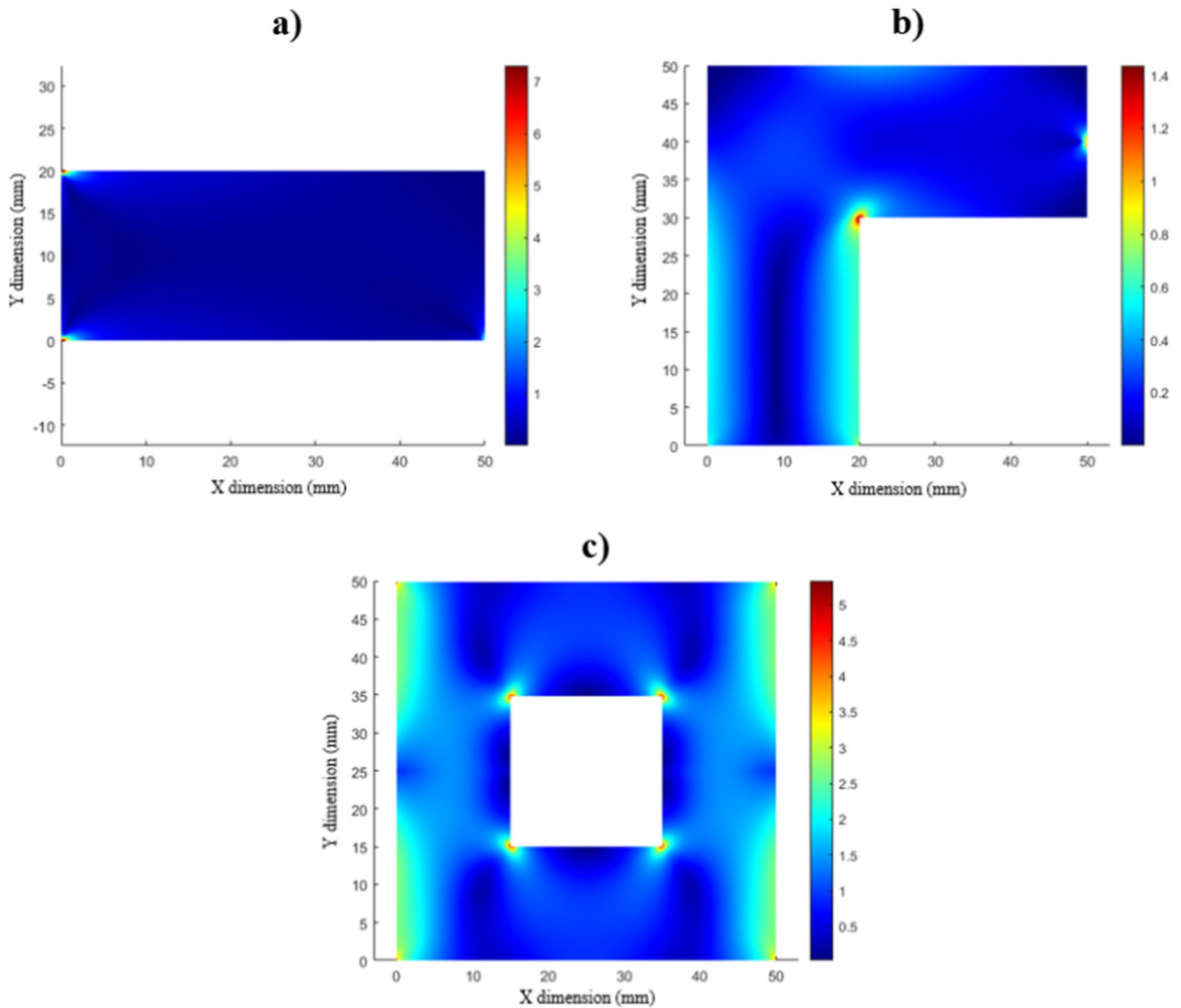


Fig. 4 FEA results of **a** beam, **b** L-shape, and **c** square with cut out

Figure 5 shows the plot of the pointing stress vectors, calculated from the first iteration’s results. It can be seen, that where the stress values are high, the curves become denser. The difference between V_x and V_y is most visible in the L-shape’s results. In Fig. 5c the darker/denser area shows the region where the pointing vectors aim in the X direction and has a higher significance in load-bearing, Fig. 5d show that the vertical strut of the L-shape mainly has pointing vectors aiming in the Y direction. Another observation is that there is a singularity point in the “sharp” corner of the L shape. After each iteration of the algorithm, this vector field would change to some extent, according to the new initial geometries. However, this

clearly represents those regions inside the domain which are mainly unnecessary for load bearing and those locations where the material must be preserved for a good mechanical performance. Even this first iteration result can be good starting point for the designers if they want to optimize the geometry based on their own considerations without the algorithm.

Figure 6 contains the results of the optimization algorithm. For each investigated cases β was set to 0.4, which means the algorithm was running until the volume decreased to 40% of the initial volume. The elimination factor Δ was set to 0.05, thus in each iteration, the volume was decreased by 5%. We have to note, that due to the smoothing process, some

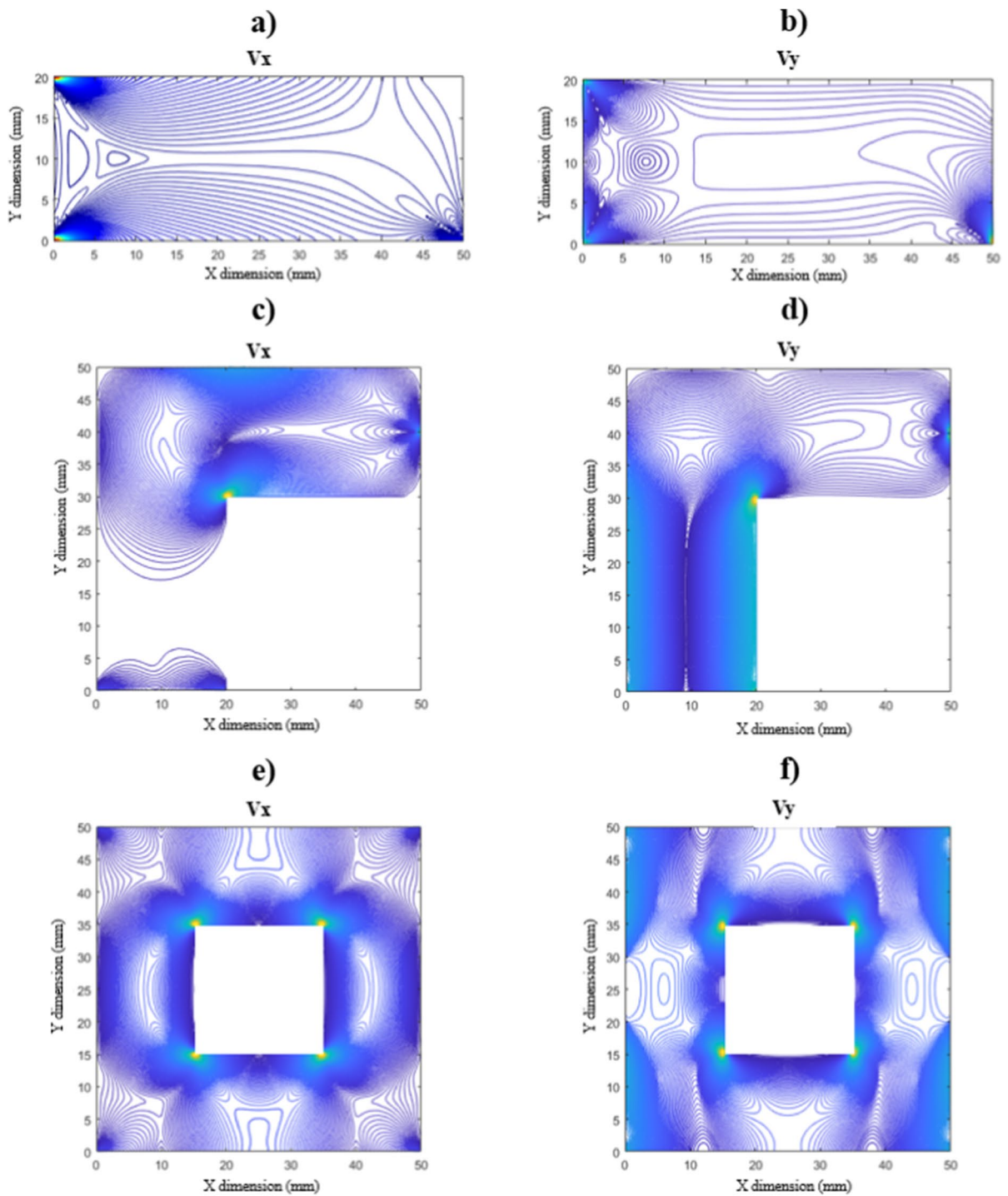


Fig. 5 Pointing stress vector plot **a** V_x of beam, **b** V_y of beam, **c** V_x of L-shape, **d** V_y of L-shape, **e** V_x of square with cut out, and **f** V_y of square with cut out

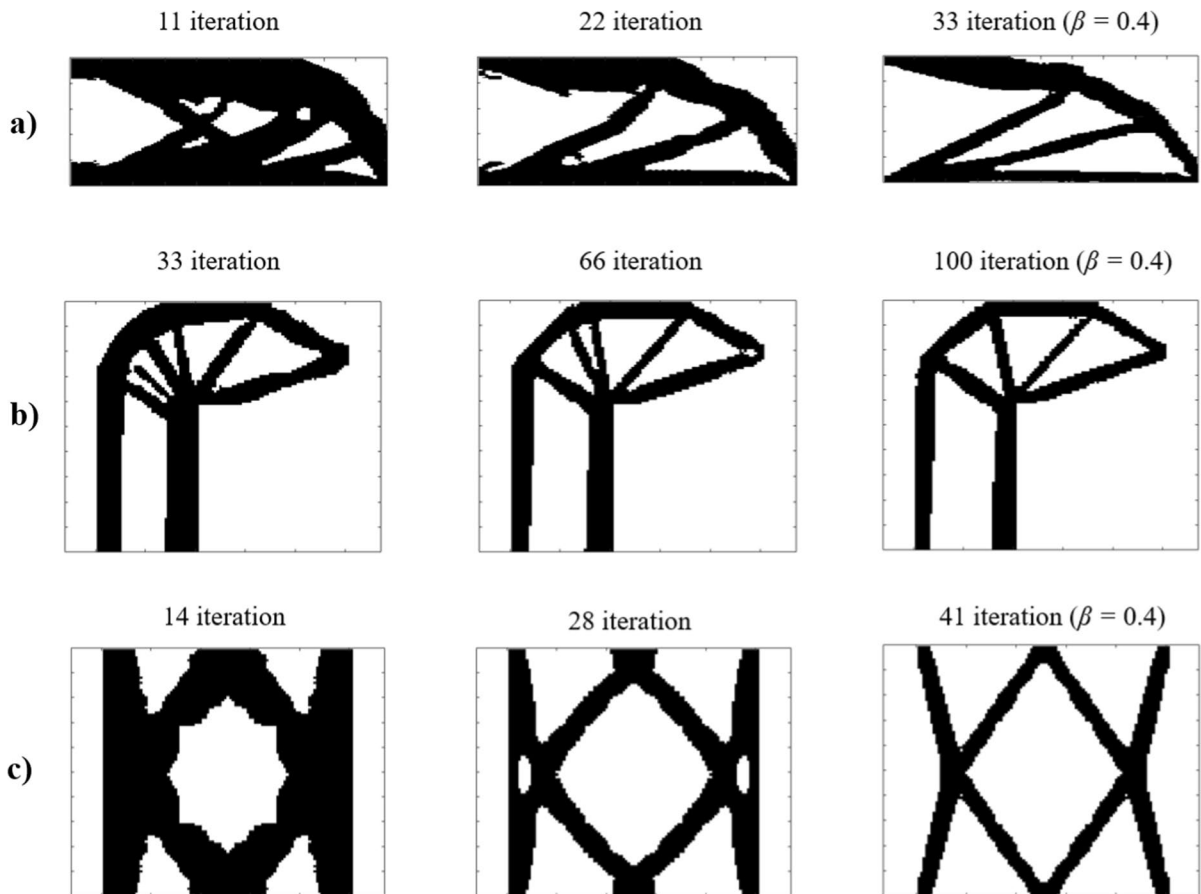


Fig. 6 Results of the load-path based optimization of **a** beam, **b** L-shape, and **c** square with cut out

eliminated nodes were re-added, and this explains why the two cases reached the finished state after a different number of iterations.

In addition, it can be seen from the Figure that the resulting final geometry follows the pointing vectors calculated at the initial state. The load-carrying trusses were formed where the load path requires it, and the regions where the presence of the material is of negligible importance were gradually eliminated.

A peculiarity during the operation of the algorithm can also be noticed, most visually in the geometry of the beam. The elements with a size smaller than the V_{cut} value are deleted, however, in the case that the load-path is not clearly defined, i.e., there are no paths with a value that is even an order of magnitude larger, then it happens that not only the existing trusses become thinner, but they can even break at some point. In this case, blind bars may form, for

example in Fig. 6b a strut at the inclination at the 33rd iteration, which blind bars will only be completely deleted in the next iteration. Therefore, in such cases, it may happen that even after making a small change on Δ , we will get a slightly different structure after the optimization has finished. From the figures for the different iterations, it can be concluded, that the model is usable, because adding elements was done for parts, where the stress was too high; and removing was done, where the stress was significantly lower than the limit value.

3.1 Remarks for the results

An explanatory figure has been made (Fig. 7). Here the density/volume decrease and the average stress in the geometry for each iteration have been plotted. As it can be seen, in each cases the algorithm

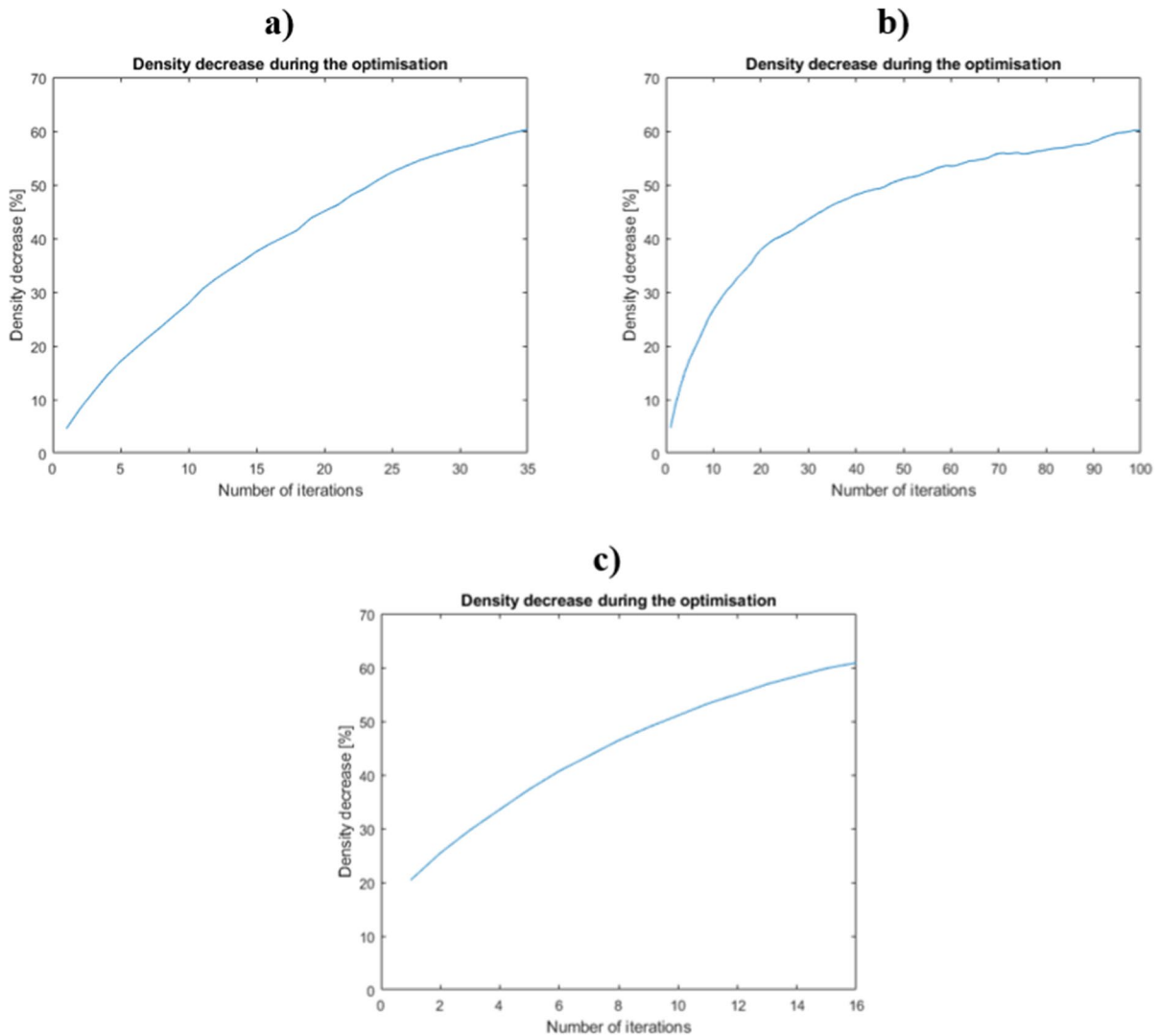


Fig. 7 **a** Density decrease of the beam, **b** density decrease of L-shape, and **c** density decrease of square with cut out

ran until the $\beta=0.4$ reductions were reached. However, the change in the density isn't linear. This is due to the smoothing cycle, as the program re-adds some elements after each element execution cycle. Because of the imperfect FEA solution and the discrete definition of V_{cut} sometimes some nodes are removed which should not. This could result in simulation error for the next iteration or misleading stress fields. Therefore, with the smoothing, these locations and their surroundings are filled back by the algorithm. The main advantage of this method is that it can straighten the load paths. As can be seen in the final shapes obtained, the bars that make

up the geometries are mostly straight and have a constant cross-section. This results in only tensile or compressive stress in the trusses, thereby reducing the number of local stress peaks, as the trusses bear the load in their entire cross-section along their length. Furthermore, it can be stated that the algorithm can even run to convergence, as it is most outstanding in Fig. 7b. In this case, at the 100th iteration, all the bars have already straightened, so when the nodes are eliminated, only their cross-section would change. However, after a certain point, if the geometry as whole experiences stresses of the same magnitude, the elements deleted at the V_{cut} value

can result in the geometry falling apart. Once a converged solution has been reached its worth to stop the optimization, since it means that the best structure has been established.

As it was mentioned before the value of applied loads and forces were insignificant for the research, since only the shape generation was investigated. The volume fraction was selected to measure the generation of topology structures which is frequently used in most TO processes. However, in the case of optimization based on force-flow, it is advised to run the algorithm until a converged solution is reached. Then the cross-section value of the generated prismatic beams of the geometries can be increased or decreased to reach the desired density/volume, or it can be assigned based on the amount of applied loads.

The resulting geometries are complex in some cases, thus it would be challenging to manufacture with traditional manufacturing technologies. In this paper, only 2D geometries were created, which could be produced by EDM, but generally saying the results are best manufacturable with 3D printing processes.

4 Conclusion

An algorithm has been created which uses boundary conditions to perform generative design. After the initial geometry was created the topology optimization based on the load paths was applied, and the usability of the proposed solution was tested on two frequently investigated 2D shapes. Based on the results it can be stated that the previously described method is valid. The numerical method was realized by numerical software (MATLAB) and applied to different geometries for presenting the usability of the model. It has been proven that the TO process can be further developed in the direction of GD, i.e., without a known initial geometry, a finished shape can be generated just from the boundary conditions, which does not require external intervention from the designer. Also, with the smoothing method, straightened strut-based geometries have been formed, which are beneficial for load bearing. Later the 2D isotropic model can be extended to a 3D, anisotropic one. After this extension, the model is going to be applicable for real manufacturing technology, for example, a typically anisotropic 3D printing process. Also, the cross-section assignment can be interpreted in later work to

make a closer connection with design for mechanical resistance.

Acknowledgements Project no. TKP2020-NVA-29 has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2020-NVA funding scheme.

Funding Open access funding provided by Eötvös Loránd University.

Declarations

Conflict of interest The authors declare that they do not have any Conflict of Interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Bendsoe M, Sigmund O (2004) Topology optimization theory, methods, and applications. Springer. <http://www.springer.de/engine/>
2. Sigmund O, Maute K (2013) Topology optimization approaches: a comparative review. *Struct Multi-discip Optim* 48:1031–1055. <https://doi.org/10.1007/s00158-013-0978-6>
3. Plocher J, Panesar A (2019) Review on design and structural optimisation in additive manufacturing: towards next-generation lightweight structures. *Mater Des*. <https://doi.org/10.1016/j.matdes.2019.108164>
4. Li S, Xin Y, Yu Y, Wang Y (2021) Design for additive manufacturing from a force-flow perspective. *Mater Des* 204:109664. <https://doi.org/10.1016/j.matdes.2021.109664>
5. Kwok TH, Li Y, Chen Y (2016) A structural topology design method based on principal stress line. *CAD Comput Aided Des* 80:19–31. <https://doi.org/10.1016/j.cad.2016.07.005>
6. Tam KMM, Mueller CT (2015) Stress line generation for structurally performative architectural design, ACADIA 2015 - Comput. Ecol. Des. Anthr. Proc. 35th Annu. Conf. Assoc. Comput. Aided Des. Archit. 2015-October

7. Xu G, Dai N, Tian S (2021) Principal stress lines based design method of lightweight and low vibration amplitude gear web. *Math Biosci Eng* 18:7060–7075. <https://doi.org/10.3934/mbe.2021351>
8. Li S, Wang S, Yu Y, Zhang X, Wang Y (2021) Design of heterogeneous mesoscale structure for high mechanical properties based on force-flow: 2D geometries, addit. *Manuf* 46:102063. <https://doi.org/10.1016/j.addma.2021.102063>
9. Kelly DW, Hsu P, Asudullah M (2001) Load paths and load flow in finite element analysis. *Eng Comput (Swansea Wales)* 18:304–313. <https://doi.org/10.1108/02644400110365923>
10. Marhadi K, Venkataraman S (2009) Comparison of quantitative and qualitative information provided by different structural load path definitions. *Int J Simul Multidiscip Des Optim* 3:384–400. <https://doi.org/10.1051/ijsmdo/2009014>
11. He Z-Q, Liu Z, Wang J, Ma ZJ (2020) Development of strut-and-tie models using load path in structural concrete. *J Struct Eng* 146:1–6. [https://doi.org/10.1061/\(asce\)st.1943-541x.0002631](https://doi.org/10.1061/(asce)st.1943-541x.0002631)
12. Kelly D, Reidsema C, Bassandeh A, Pearce G, Lee M (2011) On interpreting load paths and identifying a load bearing topology from finite element analysis, *Finite Elem. Anal Des* 47:867–876. <https://doi.org/10.1016/j.finel.2011.03.007>
13. Caetano I, Santos L, Leitão A (2020) Computational design in architecture: defining parametric, generative, and algorithmic design. *Front Archit Res* 9:287–300. <https://doi.org/10.1016/j.foar.2019.12.008>
14. Buonamici F, Carfagni M, Furferi R, Volpe Y, Governi L (2020) Generative design: an explorative study. *Comput Aided Des Appl* 18:144–155. <https://doi.org/10.14733/cadaps.2021.144-155>
15. Bukhari F (2011) A hierarchical evolutionary algorithmic design (HEAD) system for generating and evolving building design models, Ph.D Thesis. 362
16. Terzidis K (2003) Expressive form: a conceptual Approach to Computational Design. Spon Press
17. Terzidis K (2004) Algorithmic Design: A Paradigm Shift in Architecture?, *Proc. Int. Conf. Educ. Res. Comput. Aided Archit. Des. Eur.* 201–207. <https://doi.org/10.52842/conf.ecaade.2004.201>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.