

Article

TABASCO—Topology Algorithm That Benefits from Adaptation of Sorted Compliances Optimization

Bogdan Bochenek  and Katarzyna Tajs-Zielińska * Faculty of Mechanical Engineering, Cracow University of Technology, 31-155 Krakow, Poland;
bogdan.bochenek@pk.edu.pl

* Correspondence: katarzyna.tajs-zielinska@pk.edu.pl

Abstract: Although structural topology optimization has been developing for decades, it still plays a leading role within the area of engineering design. Solving contemporary design problems coming from industry requires the implementation of efficient methods and approaches. This stimulates research progress in the development of novel and versatile topology optimization algorithms. To follow these modern trends, an original topology generator has been elaborated and finally built as a Cellular Automaton with original update rules. The motivation for building the algorithm in this way came from the idea of utilizing the benefits of local compliances sorting. This is conducted on two levels: on the global level, the monotonic function mapping local compliances distribution is defined based on their sorted values; on the local level, for each cell, the compliances are sorted within the cell neighborhood. The three largest absolute values are selected, and these are the basis from which to formulate Cellular Automata update rules. These original rules can efficiently control the generation of structural topologies. This technique is somewhat inspired by the grey wolf optimizer strategy, wherein the process of updating design variables refers to the positions of the three best fitted wolves. It is proposed that we refer to the topology algorithm that benefits from the adaptation of sorted compliances optimization as TABASCO. The developed algorithm is a modified version of the flexible Cellular Automata one presented previously. The implemented extension, regarding the local level cell sorting, allows us to improve the resulting compliance values. The advantages of the algorithm, both from numerical and practical engineering points of view, as compared to the others developed within the field, may be gathered as follows: the algorithm works based on simple update rules, i.e., its numerical implementation is not complicated; it does not require gradient computations; filtering techniques are not needed; and it can easily be combined with professional structural analysis programs which allow engineering applications. The developed topology generator has been linked with ANSYS to show that it can be incorporated into a commercial structural analysis package. This is especially important with respect to the engineering implementations.



Citation: Bochenek, B.; Tajs-Zielińska, K. TABASCO—Topology Algorithm That Benefits from Adaptation of Sorted Compliances Optimization. *Appl. Sci.* **2023**, *13*, 10595. <https://doi.org/10.3390/app131910595>

Academic Editors: Stavroulakis Georgios, Loucas Papadakis and Ioannis Ntintakis

Received: 4 August 2023

Revised: 12 September 2023

Accepted: 21 September 2023

Published: 22 September 2023

Keywords: topology optimization; heuristic methods; compliance minimization; cellular automata method; grey wolf optimizer; hybrid methods



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The development of structural topology optimization has already been observed for decades; nevertheless, this approach still plays a leading role in the area of engineering design. Along with the original novel concepts applied in many research and engineering fields, a variety of approaches to topology generation have been proposed over the years. Many survey papers, e.g., refs. [1–4], supported by the recently published papers of Ribeiro et al. [5] and Logo and Ismail [6], provided a broad discussion on various issues and aspects of topology optimization.

Moreover, solving contemporary design problems coming from industry requires the elaboration of novel methods and approaches, which stimulates continuous research progress in the development of efficient topology optimization algorithms. The natural

consequence of this progress is that nowadays, commercial structural analysis software often also offers performing topology optimization. This easy access to topological design is very helpful for engineers but, on the other hand, the results produced by the black box systems are not necessarily the best to be found. It is definitely a step towards the engineering implementation of topological designs but working on efficient topology generators is still welcome. The present paper follows this trend, showing how to build an original efficient generator of structural topologies. Hence, it is built as a Cellular Automaton with original update rules. The rules utilize information gathered within cells' neighborhoods and are applied to all cells/elements. The motivation for building the algorithm came from the idea of utilizing the benefits of local compliances sorting. This is conducted on two levels: first, the monotonic function mapping local compliances distribution is defined on the global level based on their sorted values; next, on the local level, for each cell, the compliances are sorted within the cell neighborhood. Among them, the three largest absolute values are selected, and this is the basis on which to formulate Cellular Automata update rules. This technique is somewhat inspired by the grey wolf optimizer [7] strategy, wherein the process of updating design variables refers to the positions of the three best fitted wolves, namely alpha, beta and delta, wolves.

It is proposed that we refer to the topology algorithm that benefits from the adaptation of sorted compliances optimization as TABASCO. The developed algorithm is a modified version of the flexible Cellular Automata model presented previously. The implemented extension, regarding local-level cell sorting, allows us to improve the resulting compliance values. The advantages of the algorithm, both from numerical and practical engineering points of view, as compared to the others developed within the field, are as follows: The algorithm works based on simple update rules; its numerical implementation is not complicated; it does not require gradient computations; it can easily be combined with professional structural analysis programs which allows engineering applications. Moreover, generated topologies are free from so-called grey elements, i.e., ones of intermediate densities, and due to the grey elements elimination, it is easier to transform obtained numerical solution to a CAD model and to prepare elements for manufacturing. One of the most important advantages of using the algorithm proposed in this paper is that the implementation of additional filtering techniques is not necessary. However, it is worth stressing that, in favor of the CA approach, the neighborhood remains the same independent of the mesh size, whereas in the case of the filtering technique, the neighborhood size has to be selected depending on the mesh size. The problem related to the above is the generation of topologies for a large number of elements. The OC method requires adjusting filtering parameters in order to obtain a solid topology. In the case of the CA-based algorithm, it is not necessary.

The formulated original rules can efficiently control the generation of structural topologies, and the topology generator built on this basis has been linked with ANSYS to show that it can be incorporated into a commercial structural analysis package. The performance of the algorithm has been illustrated by the generation of selected plane and spatial topologies using the TABASCO rules. Exemplary engineering implementation has been included.

The content of the paper is structured as follows. The structural topology optimization via heuristic methods is reviewed in Section 2. Within this subject, a Cellular Automata approach and its implementation into topology optimization are described. The TABASCO algorithm concept is introduced in Section 3, and its performance is first illustrated in Section 4 based on the generation of plane structures topologies. Next, in Section 5, the ability of the algorithm to cope with 3D structures is presented. Finally, some concluding remarks are formulated in Section 6, which ends the paper.

2. Structural Topology Optimization and Heuristic Methods

Performing topology optimization (TO) means looking for the distribution of material within a specified domain in order to minimize a chosen objective function and to satisfy imposed constraints. It is assumed that topology optimization started with Michell's paper,

published in 1904. However, intensive studies started in the 1980s, and since then, TO has become the most popular engineering tool. The growing demands drive the development of techniques and methods of TO, which can be helpful and effectively used in engineering implementations and practical problems.

Heuristic methods are becoming more popular and useful in engineering applications of topology optimization. Their simplicity of implementation and fast convergence, which does not require a large number of finite element analyses, make heuristics a useful basis for different approaches applied to the topology optimization of realistic engineering problems. Among very powerful heuristic methods are those based on physical and chemical phenomena or population intelligence. The latter are very effective, but their greatest weakness is the need to refer to computation of the objective many times, i.e., for each individual at each iteration step. This means the necessity of multiple finite element analyses of designed structures, which may be very time consuming for coarse mesh. Solving this problem efficiently is an important task for designers. Nevertheless, heuristic techniques are widely discussed and continuously applied in engineering topology optimization. As an example, they may serve topology optimization with the use of an ant colony approach [8] or a genetic algorithm [9], simulated annealing [10], or binary particle swarm optimization [11]. Less popular examples may serve a meta-heuristic using probabilistic learning [12], a heuristic algorithm mimicking a self-evolution of a structure [13], the big bang–big Crunch algorithm [14], and the recently published hunter–prey optimization algorithm [15].

As already mentioned in the introduction, an approach based on the concept of Cellular Automata (CA) is applied in this paper. Cellular Automata mimic the behavior of complex systems in a relatively easy way. Hence, a sequence of simple decision making rules is implemented. These rules control the performance of a considered system. The engineering implementation requires the decomposition of the considered domain into a lattice of cells which usually form a uniform lattice. The examples of plane and spatial neighborhoods for regular cells are presented in Figures 1 and 2, respectively:

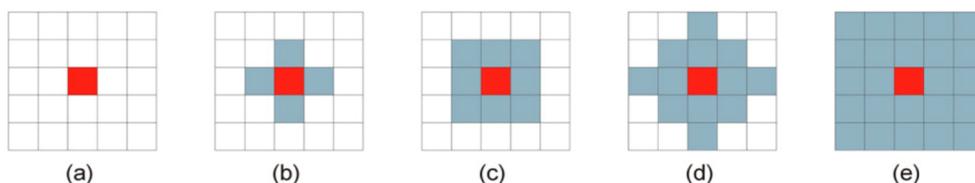


Figure 1. Neighborhoods for plane problems: (a) empty; (b) von Neumann; (c) Moore; (d) MvonN; (e) extended Moore.

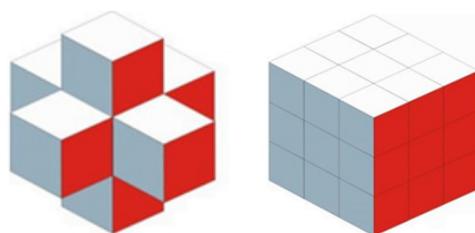


Figure 2. The von Neumann (left) and the Moore (right) neighborhoods for spatial problems.

Each cell, together with cells it is connected to, forms a neighborhood. The cells carry only local information. It is assumed that the interaction between cells is limited to their neighborhood. Local homogenous rules govern the evolution of cells state. The rules are applied simultaneously to each cell, and they are identical for all neighborhoods. The physical quantities are also updated repetitively and as a result, the process covers the global behavior of the system. Since the late 1940s when von Neumann and Ulam proposed

the concept of Cellular Automata, this idea has been found interesting by researchers representing various research fields.

Cellular Automata have proved to be an effective tool in engineering optimization for both size optimization (e.g., [16,17]) and topology optimization, as started by Inou et al. [18]. The effectiveness of the CA approach in the generation of structural topologies and its versatility can be observed in various engineering problems starting from compliance minimization problems (e.g., [19]), including problems with stress and displacement constraints (e.g., [20]), through reliability-based topology optimization (e.g., [21]) or dynamic–static coupling topology optimization ([22]) to the topology optimization of energy absorbers ([23,24]) or the optimization of multi-materials structures ([25] or [26]), composites ([27]), and design of materials (e.g., [28,29]). The mentioned papers may serve only as examples of numerous applications of the Cellular Automata method. Additionally, a new trend in developing this method can be observed, namely, a hybridization of CA with other methods, pertaining to both classical methods ([19,30,31] for size optimization) and heuristics. Among hybridization CA with heuristics applied to topology optimization, the following can be mentioned: a Cellular Automaton mimicking colliding bodies [32], a hybrid evolutionary bi-directional Cellular Automaton [33], and the cuttlefish algorithm (CFA) for truss topology optimization utilizing the Cellular Automata approach [34]. The survey of combinations of CA with heuristics in broad fields of engineering computations is presented in [35,36], where, among others, the cellular grey wolf optimizer (CGWO) is proposed. In that paper, utilizing the hunting mechanism of grey wolves, the analysis steps need to be performed for each individual at each iteration, which may cause a significant increase in computation time for large engineering problems. It is worth stressing that in the present paper, which also refers to GWO, the idea is different—the analysis is performed once for each iteration and the derived rules of Cellular Automata govern the behavior of the system.

In general, the effectiveness of the population-based algorithms strongly depends on the population size. An extensive discussion of this aspect of topology optimization problems can be found, e.g., in [37]. In the optimization process at each iteration the fitness function must be calculated for each individual. As a result, the population-based methods usually require a larger number of functional calls than other non-population-based heuristics.

When considering the topology optimization problems using Cellular Automata, it is necessary to mention the valuable and groundbreaking papers of Tovar et. al, e.g., [38], where the hybrid Cellular Automata (HCA) is introduced. It must be underlined, however, that in those papers, hybridization refers to the interaction of analyses based on finite elements with the Cellular Automaton approach.

3. TABASCO Algorithm—Two-Level Sorting and Update Rules Inspired by the Grey Wolf Optimizer

Formulating the topology optimization problem, in the present study, the attention is focused on the minimization of structural compliance c (see Equation (1)). This is the classical approach (a discussion of the problem can be found in the early paper [39]). In this paper, the approach based on finite elements has been applied and the structural topology optimization problem formulation takes the form known from the well-recognized paper by Sigmund [40]. Hence, it is assumed that only a specified volume fraction κ of a material is distributed within the final structure. This results in the constrained volume V , Equation (2):

$$\text{minimize } c(\mathbf{d}) = \mathbf{u}^T \mathbf{k} \mathbf{u} = \sum_{i=1}^N d_i^p \mathbf{u}_i^T \mathbf{k}_i \mathbf{u}_i, \quad (1)$$

$$\text{with respect to } V(\mathbf{d}) = \kappa V_0, \quad (2)$$

$$\mathbf{k} \mathbf{u} = \mathbf{f} \quad (3)$$

$$0 < d_{min} \leq d_i \leq 1. \tag{4}$$

The following quantities have been introduced at the element level: displacement vector u_i , stiffness matrix k_i , design variable d_i , defined as the relative material density, and number of elements N , as well as the global quantities: stiffness matrix \mathbf{k} , displacement vector \mathbf{u} , and force vector \mathbf{f} from Equation (3). In order to avoid singularity, the simple bounds related to the design variables with minimal non-zero density d_{min} (e.g., 10^{-9}) have been introduced in Equation (4). In the above formulation, the convenient finite element approach has been adopted.

The broadly accepted material model SIMP—Solid Isotropic Material with Penalization [35]—has been adopted, meaning that the elasticity modulus E_i for each cell/element is the function of the design variable d_i as shown in Equation (5):

$$E_i = d_i^p E_0. \tag{5}$$

The quantity E_0 is the elasticity modulus of a solid material, whereas p , usually equal to 3, plays the role of penalization power. The SIMP approach is the most common material interpolation scheme, and it is widely discussed and exploited in literature. The majority of papers on topology optimization use SIMP (or SIMP modifications) while performing calculations, using at the same time new techniques for updating the design variables—i.e., those conventional or newly developed heuristics. It is worth mentioning here the various modifications of the SIMP method (e.g., [41–43] and many more) created to prevent some difficulties in the application of SIMP in selected complex problems such as multi-material topology optimization, dynamic problems, design-dependent loads, thermodynamic problems.

While performing the optimization process, the values of the design variables associated with cells are updated based on the local rules.

In order to solve the problem, a heuristic local CA update rule is proposed, utilizing the information gathered from a selected neighborhood. New values of design variables can be calculated based on the Gauss–Seidel iteration mode, i.e., a cell updates its state based on updated values found for cell neighbors in the current and previous iteration, or on the Jacobi iteration mode, i.e., a cell updates its state based on the states of the surrounding cells obtained in the previous iteration only. This scheme was chosen for this paper.

The Jacobi scheme of updating is utilized, which means that the states of neighboring cells known from the previous iteration are the basis for the update process, as in Equation (6):

$$d_i^{new} = d_i + \Delta d_i. \tag{6}$$

The quantity Δd_i represents the update component implemented in each iteration. Its selection decides the effectiveness of the updating scheme; therefore, this is the important research issue. The original proposal is included in this paper. It can be treated as an extension and modification of the rule discussed in the paper [44].

The updating process consists of a few steps. First, structural analysis is performed and the compliances values for N cells are computed. Next, the compliances are sorted in ascending order and divided into three subsets: N_1 cells having the lowest values, $N-N_2$ cells with the largest values, and the remaining cells having intermediate values of compliances. The cells from the first subset have $-C$ value assigned, the ones from the second subset have a value of C , whereas for cells from the intermediate interval, a function $f(i)$ of local compliances is defined. Finally, the function mapping local compliances, which utilize results of the **global sorting**, take the form of Equation (7), where i denotes a cell index:

$$F(i) = \begin{cases} -C & \text{if } i < N_1 \\ f(i) & \text{if } N_1 \leq i \leq N_2 \\ C & \text{if } i > N_2 \end{cases} \tag{7}$$

Having finished with the analysis at the global level, the local rules of design variables updating can be established. Therefore, for each cell, the neighboring cells are taken into consideration. In the former papers, in dealing with the implementation of Cellular Automata rules, in the process of topology generation, information gathered within all cells forming neighborhood has been considered.

The novel proposal, inspired by the grey wolf optimizer strategy, where the process of updating design variables refers to the positions of the three best fitted wolves, assumes that information from only three of the most influential cells is the basis for forming the update rule. Hence, on the local level, for each cell, the compliances are sorted as absolute values within the cell neighborhood. As the result of **local sorting**, the three values F_1, F_2, F_3 are chosen, and these are the basis from which to build the CA rule for the updating the design variables. The rule then takes the following final form:

$$d_i^{new} = d_i + \frac{F_1 + F_2 + F_3}{3} m. \quad (8)$$

Updating design variables results in either adding material to or removing it from a cell i . The quantity m in Equation (8) limits the possible change of the design variables' values. The function $f(i)$, introduced in Equation (7), is adopted here in the form of Equation (9):

$$f(i) = C \frac{\tanh\left[\beta\left(\frac{i-N_1}{N_2-N_1} - \frac{1}{2}\right)\right]}{\tanh\left(\frac{1}{2}\beta\right)}. \quad (9)$$

The above form enables flexibility of the function, which is controlled by the quantity β . One can observe that β tending to zero $f(i)$ approaches linear function, while for large values of $\beta f(i)$, these are close to the step function. As mentioned above, $f(i)$ is defined for cells from the interval $[N_1, N_2]$. Its width can be altered in the iteration process influencing the elimination of void cells if it is wide and forcing the elimination of cells of intermediate densities if it is narrow (for details, see [44]).

As to the move limit m , it plays in (8) the role of a multiplier. It becomes active if the updated design variable value is outside the admissible interval:

$$\max(d_{min}, d_i - m) \leq d_i^{new} \leq \min(1, d_i + m). \quad (10)$$

The m value is kept constant in the iteration process. The value of 0.2 is a common choice in the literature, e.g., [40,45]. The already-gathered experience from dealing with setting the move limit is that for its small values, the topology generation process slows down and the tendency of creating thin lines of generated topology can be observed. On the other hand, large values of the move limit change the structure topology and the compliance value, which may also cause the topology generation process to diverge.

4. The TABASCO Algorithm in Use

To illustrate how the TABASCO algorithm works, introductory examples have been proposed and detailed calculations are presented in this section. The Cellular Automata method allows for two approaches: simultaneous analysis and design, or the sequential approach chosen in this paper. So, the analysis using the finite element method is performed first and the design updating scheme is applied separately at each iteration. The updated value of design variable for each cell is based on the states of the neighboring cells determined in the preceding iteration, according to the Jacobi iteration scheme. The Moore-type neighborhood has been implemented. Each application of local update rules is preceded by finite element analysis performed using Matlab R2020b or Ansys 14 software. The diagram presenting the TABASCO workflow is given in Figure 3.

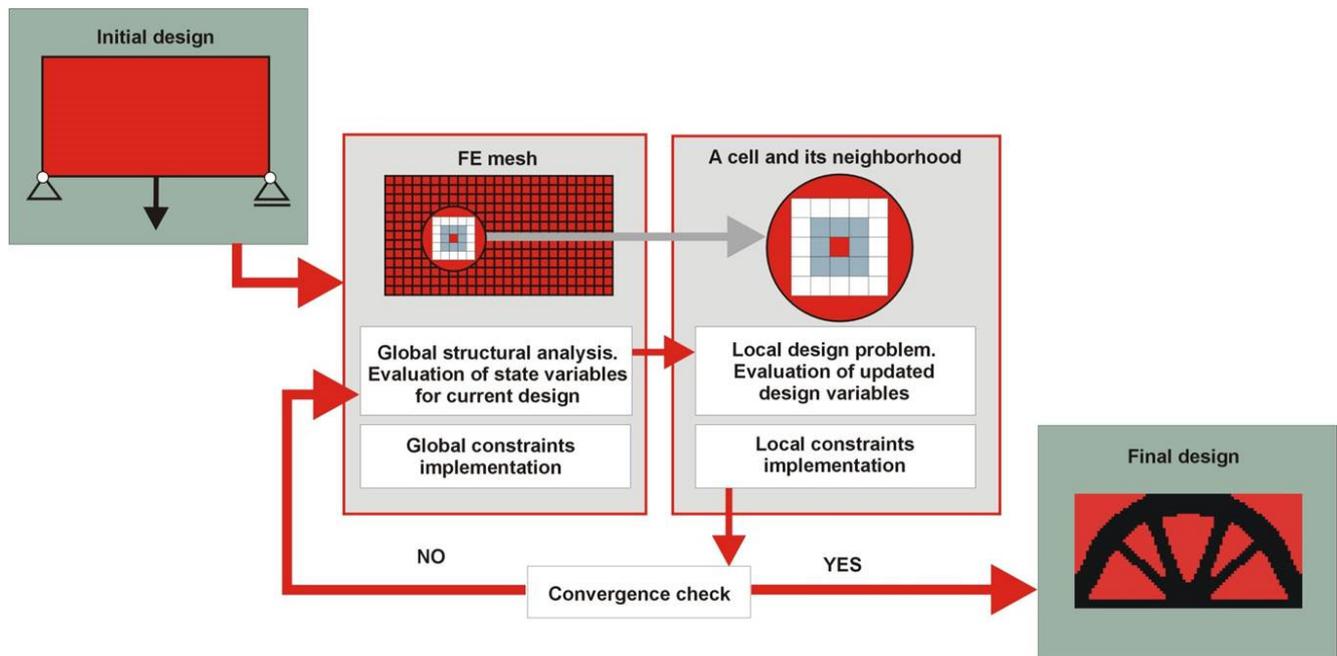


Figure 3. Diagram presenting the TABASCO workflow.

The stopping criterion for iteration process can be selected according to the user assumptions. The defined number of iterations was chosen in these investigations, but in general, it can be specified, for example, based on the objective function convergence rate.

The present paper proposes the new, original concept of a novel topology generator. The TABASCO algorithm utilizes some flexible local rules, discussed in earlier papers; therefore, the results of computations refer to some earlier concepts. To test the versatility of the new idea, selected 2D and 3D structures were examined.

The rectangular, plain structure has been selected as the first test example (see Figure 4).

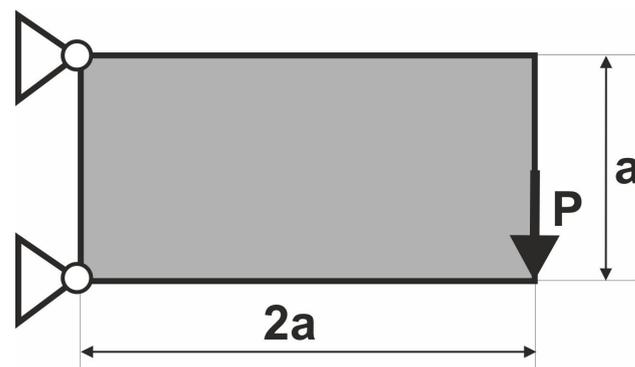


Figure 4. Short cantilever: computational model including supports and applied loads.

The regular mesh of 20,000, 200×100 4-node elements of edge length 1 mm has been implemented to perform the FEM analysis and the process of topology generation. The value of applied load has been set to $P = 100$ N, while the Young modulus and Poisson ratio have been set to values of $E = 10$ GPa and $\nu = 0.3$, respectively. The value of global constraint, i.e., the volume fraction 0.3, has been assumed.

The resulting topology and the overview of the topology generation process are presented in Figures 5 and 6, respectively.

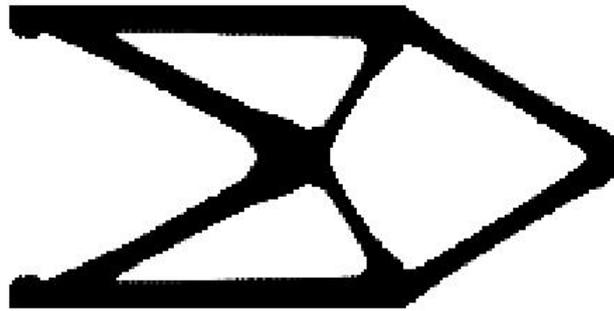


Figure 5. Short cantilever results: the final topology.

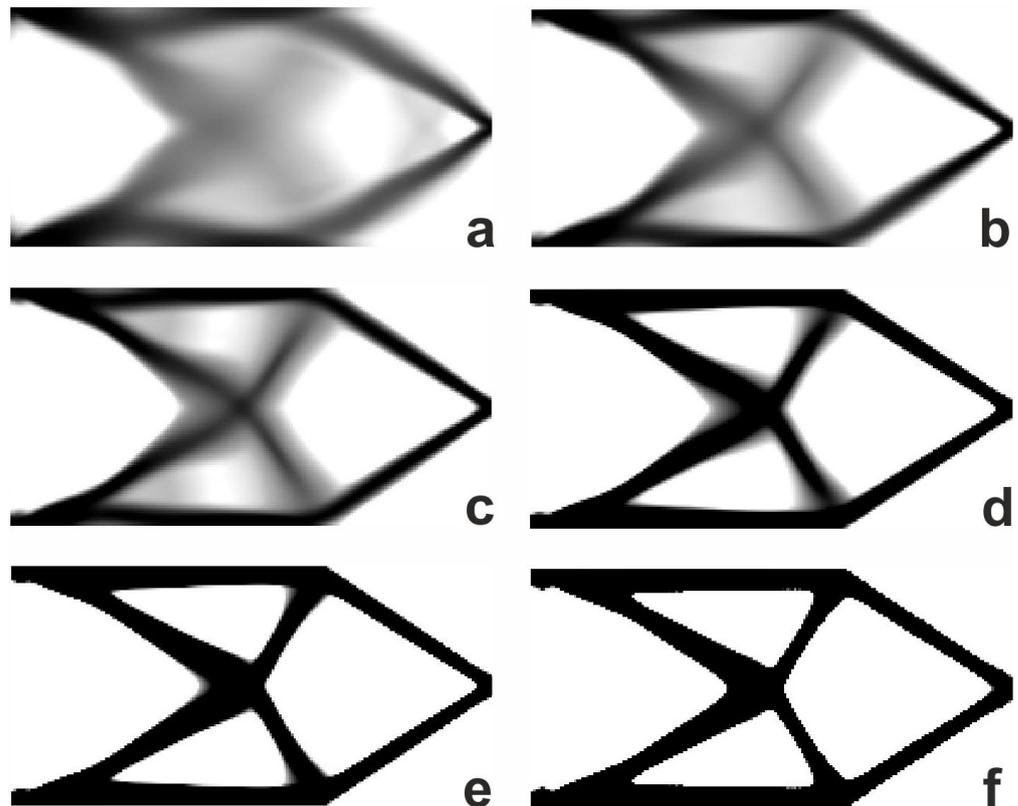


Figure 6. A look inside the topology generation process: (a) Iteration 5; (b) Iteration 10; (c) Iteration 20; (d) Iteration 30; (e) Iteration 40; (f) Iteration 80.

The iteration history, illustrating a good convergence to the optimized solution, is given in Figure 7.

A large number of iterations has been performed for all examples to allow comparisons of its performance for all considered problems. It is also intended to underline a fast decrease in the objective and a stable behavior when approaching minimal value. In Figure 7, presenting compliance history, the additional figure for the narrow interval of objective function values has been added, which gives us a closer look at the compliance behavior.

The resulting compliance value equals 133.22 Nmm for $\beta = 0.1$. The original FCA [44] allows us to obtain the value of 134.14 Nmm for optimized topology. It is also observed that the number of iterations needed to find the solution was reduced, i.e., it is 25 iterations, whereas FCA requires 46.

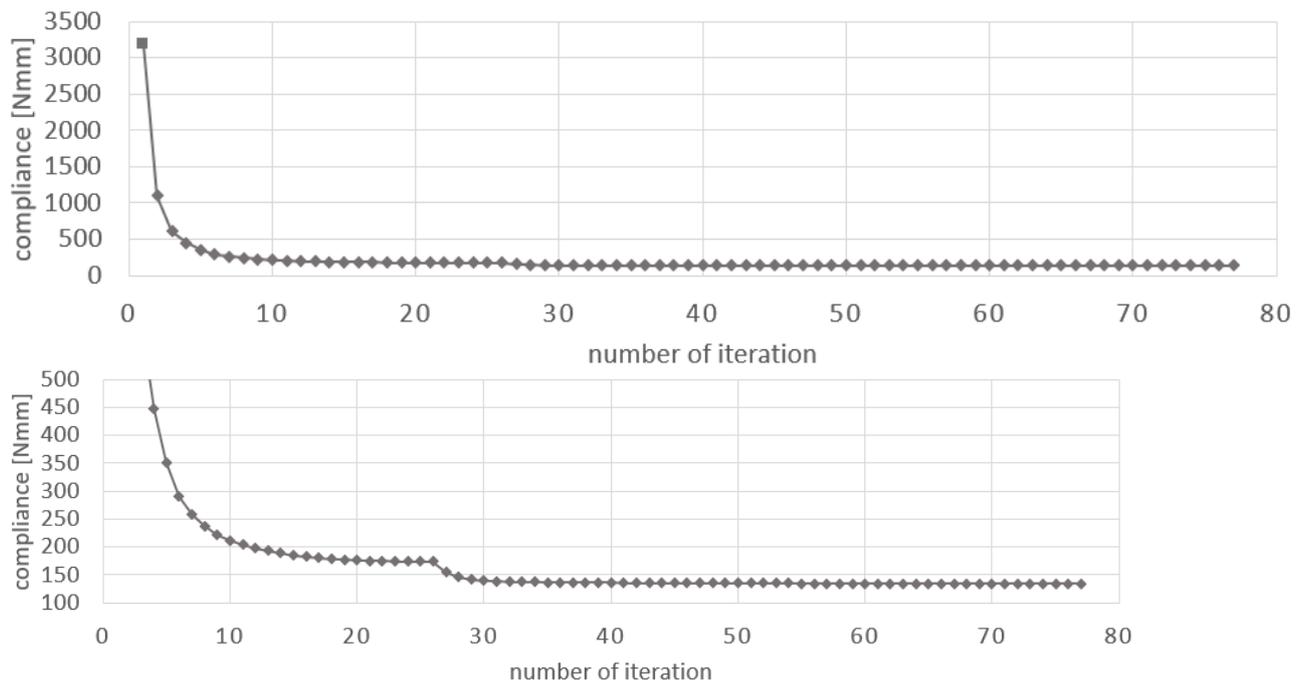


Figure 7. Compliance history for the short cantilever. The figure below provides a closer look, referring to the narrowed interval of objective function values.

It can be noticed that the solution for this example is sensitive to the value of β , but to avoid the necessity of adjusting this control parameter, the self-adaptive technique can be implemented [46], which, in each step of the iteration process, can track the objective function's lowest value. This technique can be realized by updating the values of design variables using, simultaneously, three values of β at each iteration, i.e., it can propose small, middle, and large ones. The algorithm selects the set of design variables created by this scheme for which the compliance has been found to be the smallest, and this set is the starting point for the next iteration.

The effectiveness of the proposed sorted compliances optimization concept (see [44]) can be controlled not only by the β parameter; the adaptive technique for the width of the interval $[N_1, N_2]$ can be suggested. In the first exploration phase of the calculations, when the preliminary layout of the structure is being sought, the interval $[N_1, N_2]$ is proposed to be wide. Then, in the exploitation phase, in order to guide the calculations to a black/white (solid/void) structure, the interval is successively reduced. Based on the numerous tests performed [44], the following unified strategy can be proposed: if the iteration number is lower than 25, then $N_1 = 0.02 N$ and $N_2 = 0.98 N$ for the second phase; while, if the iteration number is greater than 25, $N_1 = 0.6 N$ and $N_2 = 0.98 N$, where N is a number of elements (cells).

As one can notice, the solution is free from the checkerboard effect, without any of the additional filtering or techniques reported in, e.g., [47]. The gray elements are absent in the solutions obtained with the proposed method, which is a great advantage compared with the OC method, for which the scale of occurrence of gray elements in the solution is dictated by additional parameters.

To complete the discussion of generating the optimized topology conducted for the test structure, the comparison with the results found using the well-known optimality criteria method [40] was prepared. The final compliance for the mentioned code was 136.55 Nmm.

The long cantilever structure shown in Figure 8 has been chosen as the next test example.

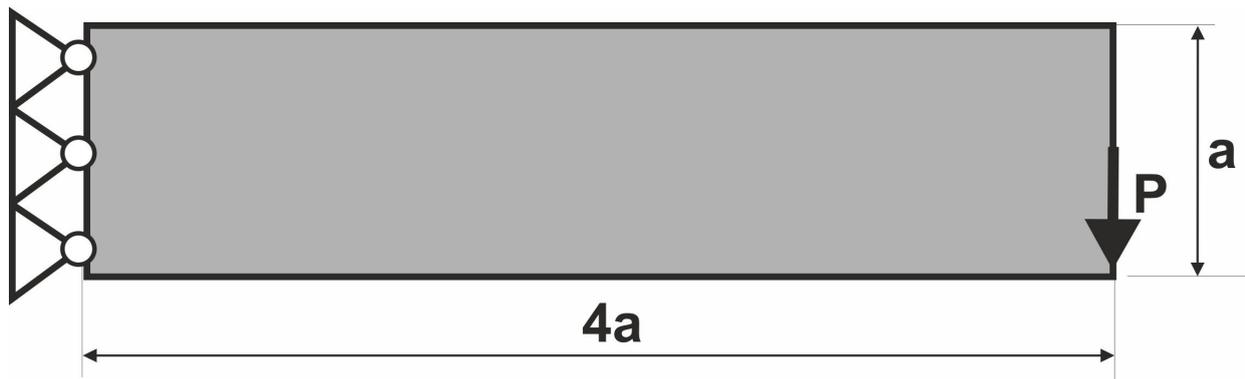


Figure 8. Long cantilever: computational model including supports and applied loads.

In this example, the value of load equals $P = 100 \text{ N}$, the parameter a is equal to $a = 10 \text{ mm}$, while for the Young modulus and Poisson ratio, $E = 10 \text{ GPa}$ and $\nu = 0.3$, respectively. The mesh of finite elements consists of 14,400 square elements, 240×60 , regular mesh, 4-node elements, an element edge length 1 mm. The calculations were performed for the volume fraction 0.5, and the resulting topology is presented in Figure 9, while the iteration history is presented in Figure 10. For the generated topology of the second test structure, the value of the compliance equals 365.69 Nmm, and for original, former FCA [44], the value of the compliance equals 371.97 Nmm.

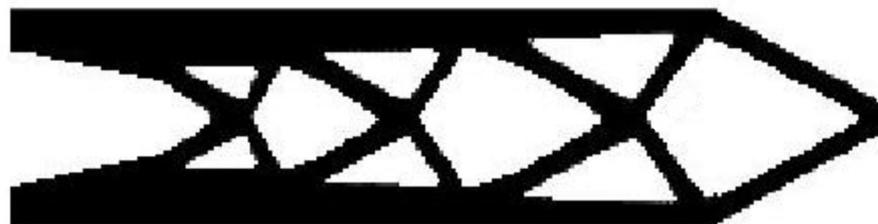


Figure 9. Long cantilever results: the final topology.

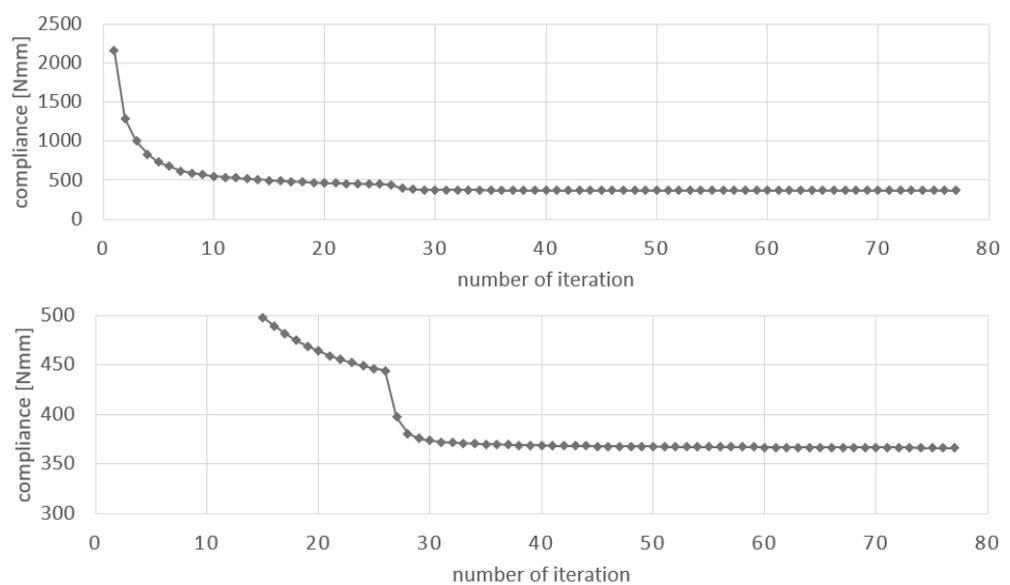


Figure 10. Compliance history for the long cantilever. The figure below gives a closer look, referring to the narrowed interval of objective function values.

To sum up the investigation, the snail-like structure has been selected (see Figure 11) as the third test example.

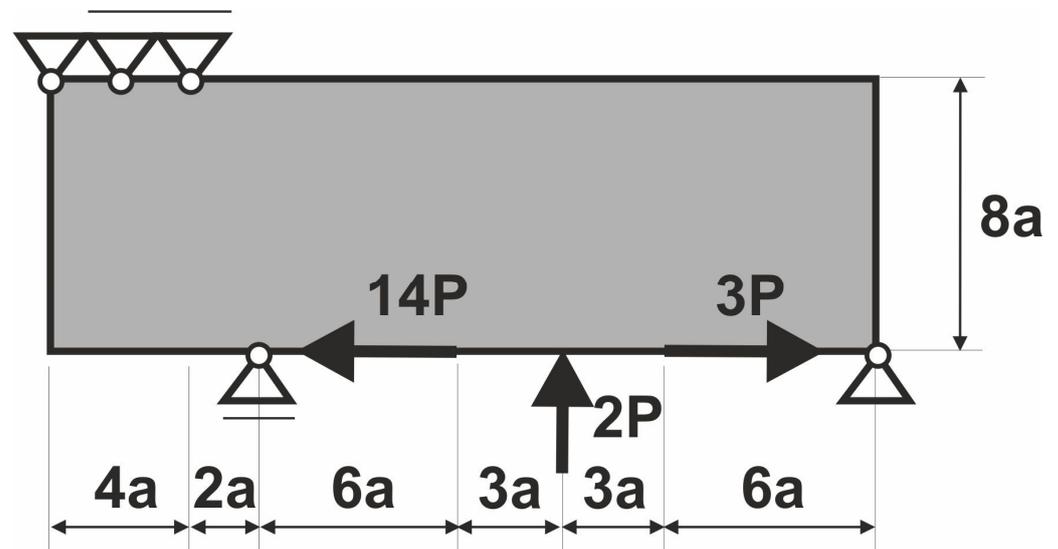


Figure 11. Snail-like structure: computational model including supports and applied loads.

The values of parameters P and a equal $P = 100$ N and $a = 10$ mm, while the Young modulus and Poisson ratio equal $E = 10$ GPa and $\nu = 0.3$, respectively. The assumed volume fraction for this example is 0.5. The finite element mesh consists of 19,200 square elements (240×80 , regular mesh, 4-node elements, element edge length 1 mm).

The final topology and the iteration history illustrating the fast convergence to the optimal solution are presented in Figures 12 and 13, respectively.



Figure 12. Snail-like structure results: the final topology.

The compliance value for the final topology of the third test structure equals 12.659 Nmm, and for original, former FCA [44], the value is 12.662 Nmm. This result has been obtained for $\beta = 0.1$.

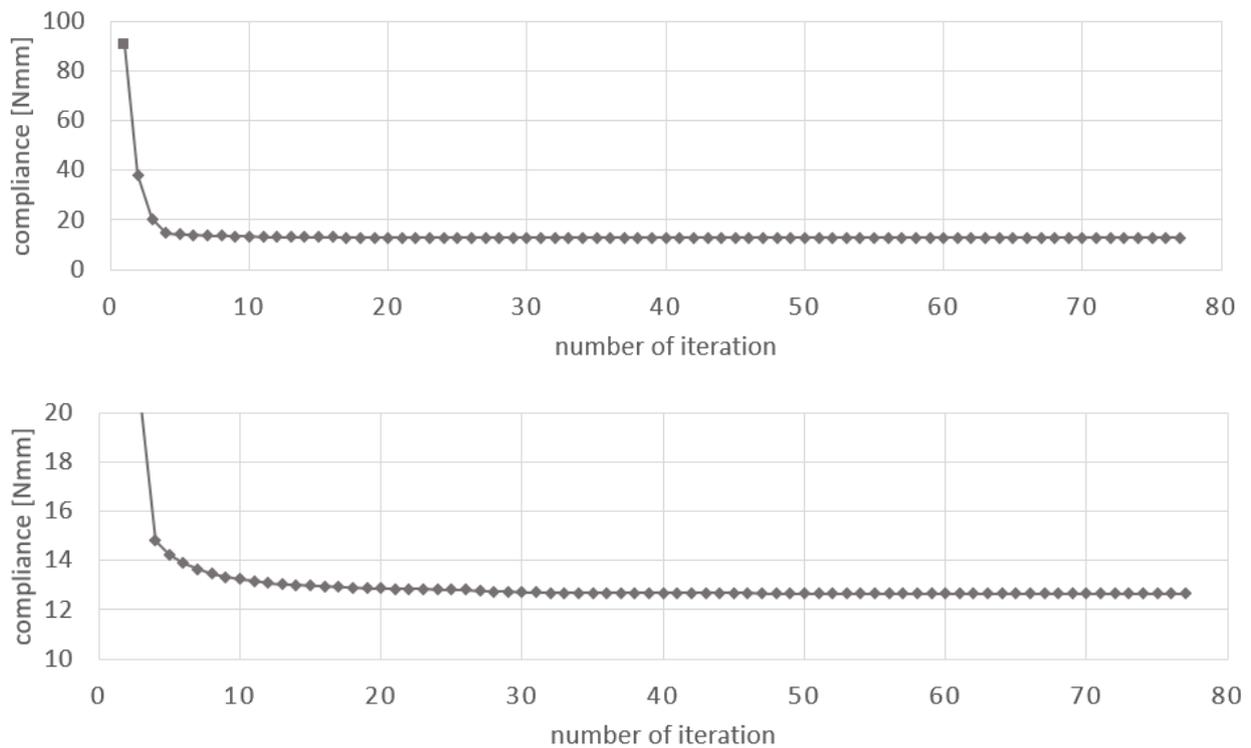


Figure 13. Compliance history for the snail-like structure. The figure below gives a closer look, referring to the narrowed interval of objective function values.

5. Generation of 3D Topologies with TABASCO

As the first example of the design of a spatial structure using the proposed method, the tower under four concentrated loads shown in Figures 14 and 15 has been selected.

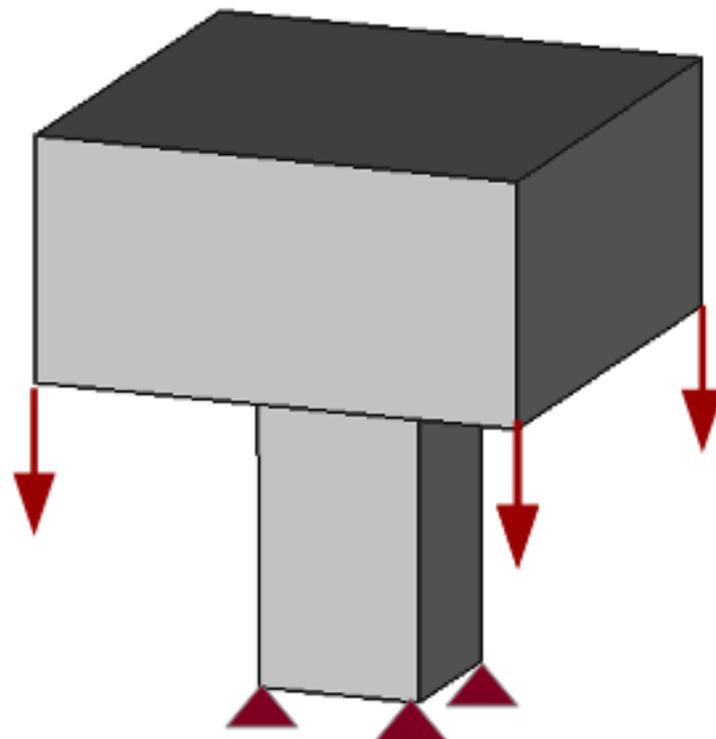


Figure 14. Tower structure: computational model including supports and applied loads.

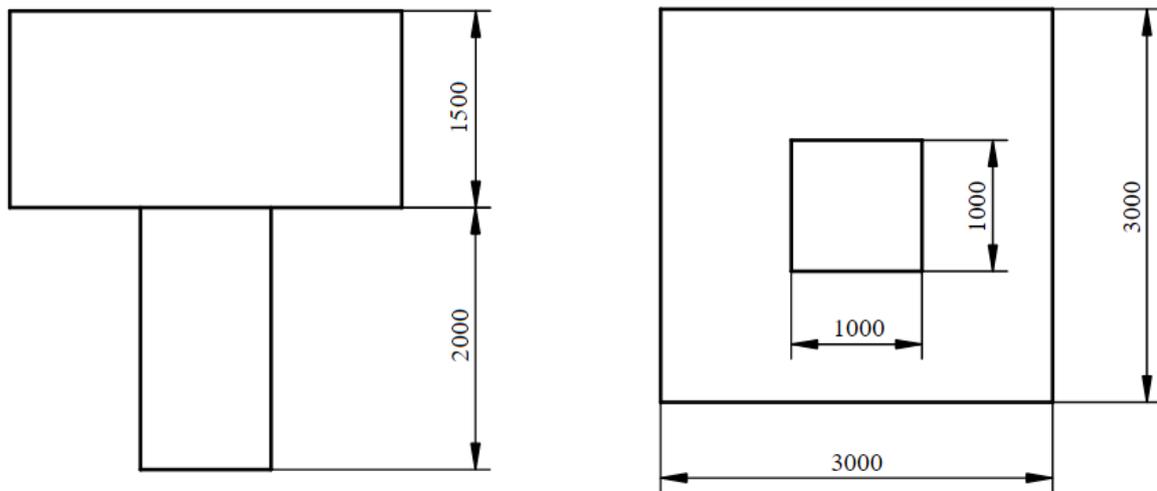


Figure 15. Tower structure: dimensions [mm].

The structure is discretized using 31,000 8-node elements. The quarter of a structure is considered according to symmetry. For this example, the data are selected as $E = 200$ GPa and $\nu = 0.3$. Four concentrated loads values of 1000 N each are applied. The assumed volume fraction equals 0.1. The calculation was performed for $\beta = 0.1$. The final layout of the structure with the resulting compliance equal to 2.46 Nmm is illustrated in Figure 16, while the overview of the topology generation process is presented in Figure 17. The compliance history is presented in Figure 18. To make a comparison with the method reported in the literature, the optimality criteria method was employed, and the final compliance obtained in this way equals 2.76 Nmm.

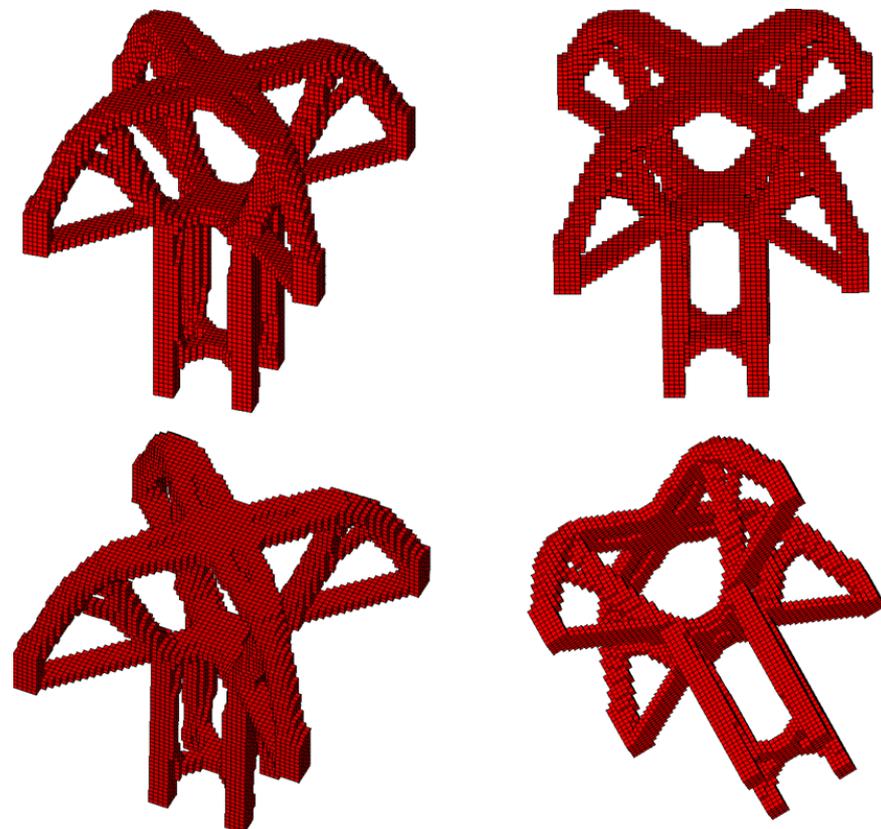


Figure 16. Tower structure results: the final topology with mesh grid. Views from different angles.

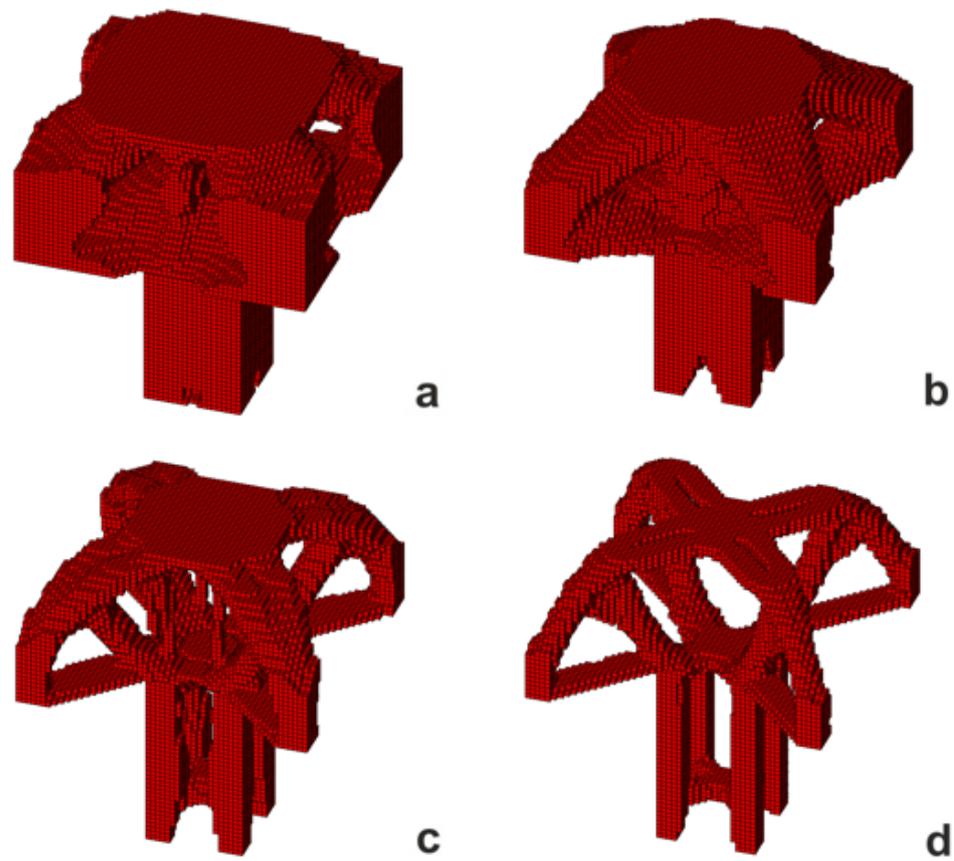


Figure 17. TAhe look inside the topology generation process: (a) Iteration 3; (b) Iteration 10; (c) Iteration 30; (d) Iteration 70.

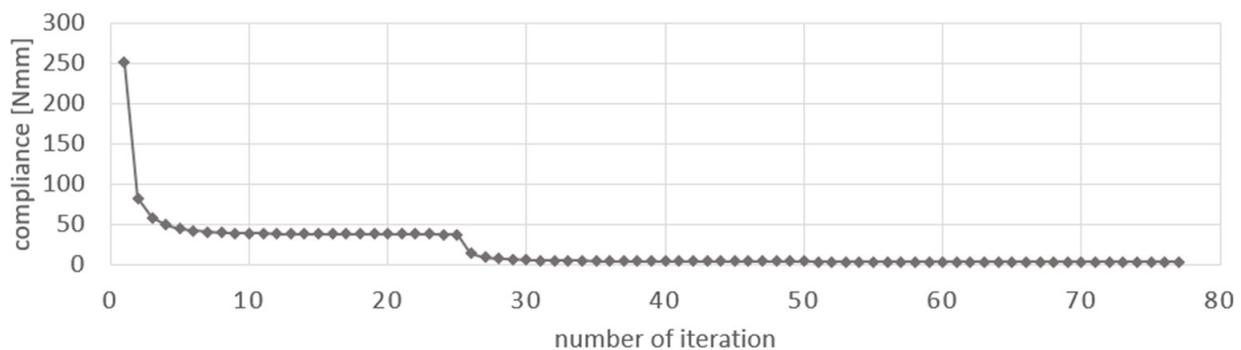


Figure 18. Compliance history for the tower structure.

One can notice a sudden decrease in the objective function values after 25 and 50 iterations; this is related to the implemented adaptive technique mentioned in Sections 3 and 4. Due to the update rule flexibility, the interval $[N_1; N_2]$ was reduced. The unified strategy of adjustment for implementation is as follows: if the iteration number is lower than 25, then $N_1 = 0.02 N$ and $N_2 = 0.98 N$; for the second phase (iteration number greater than 25), $N_1 = 0.6 N$ and $N_2 = 0.98 N$; and for the third phase (iteration number greater than 50), $N_1 = 0.8 N$ and $N_2 = 0.98 N$, where N is a number of elements (cells). This extension for the third phase is applied according to the bigger number of iterations needed for 3D examples (compared to 2D examples).

The gray elements are absent in the solution obtained with the proposed method, which is illustrated in Figure 19 by the change in the percentage of gray elements at each iteration step.

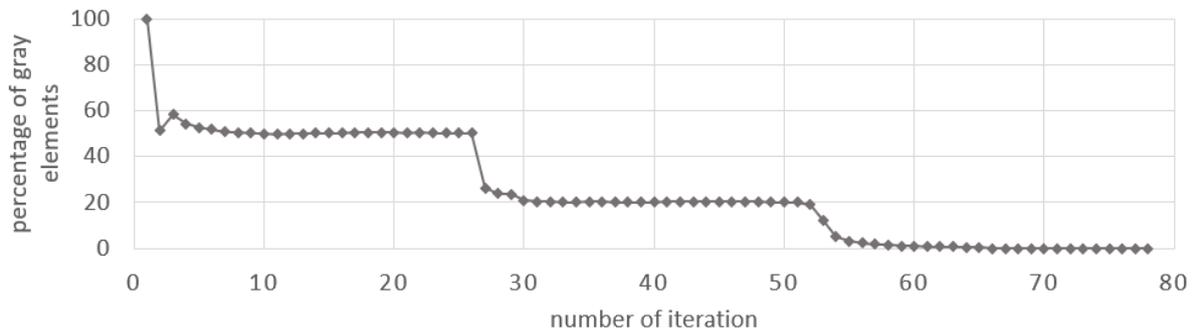


Figure 19. Percentage of gray elements at each iteration step for the tower structure.

As the second example, the spatial rocker arm under bending loads shown in Figures 20 and 21 has been selected.

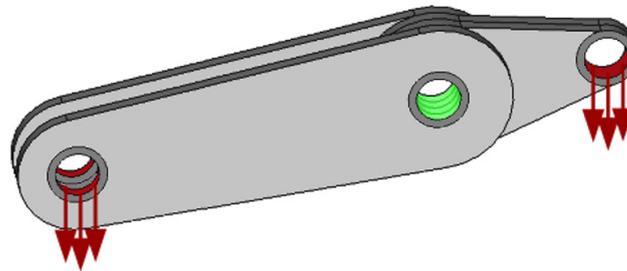


Figure 20. The rocker arm: computational model including supports (green) and applied loads (red).

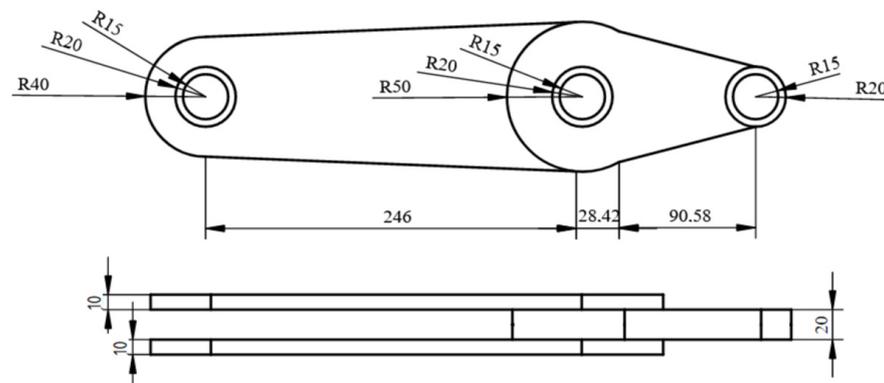


Figure 21. The rocker arm: dimensions [mm].

The structure is discretized using 164,964 8-node elements. Within the structure, the non-optimized volumes are defined and marked in Figure 20 as the dark gray regions. The design domain is marked as the light gray region. The vertical load of value of 4060 N is distributed at selected nodes across the bottom edge of left cylinders (see Figure 20) and 8120 N across the bottom edge of right cylinder. The inner wall of the middle cylinder is fixed (see Figure 20, green color). The values of material properties are selected as $E = 210$ GPa and $\nu = 0.3$. The calculations were performed for $\beta = 8$. The resulting topology is given in Figure 22. A final compliance for the volume fraction 0.2 equals 5212 Nmm, and the compliance history is presented in Figure 23. To attain a comparison with a method reported in the literature, the optimality criteria method was employed, and the final compliance obtained this way equals 5756 Nmm.

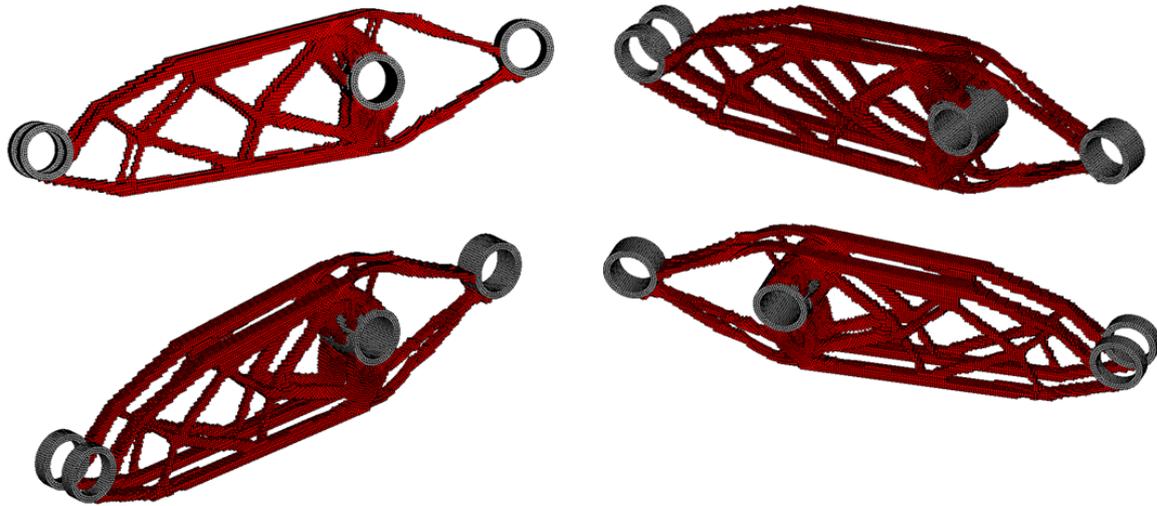


Figure 22. The rocker arm results: the final topology with mesh grid.

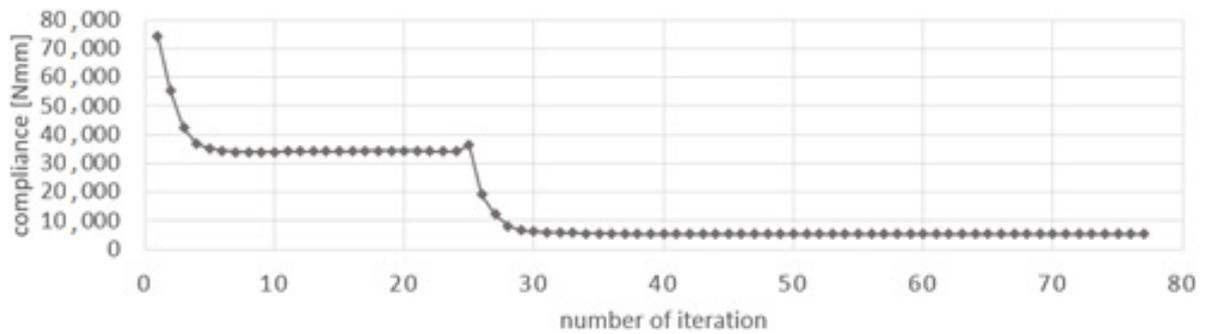


Figure 23. Compliance history for the rocker arm structure.

The adaptive technique mentioned in Sections 3 and 4 was adapted. The gray elements are absent in the solution obtained with the proposed method, which is illustrated in Figure 24 by the change in the percentage of gray elements at each iteration step.

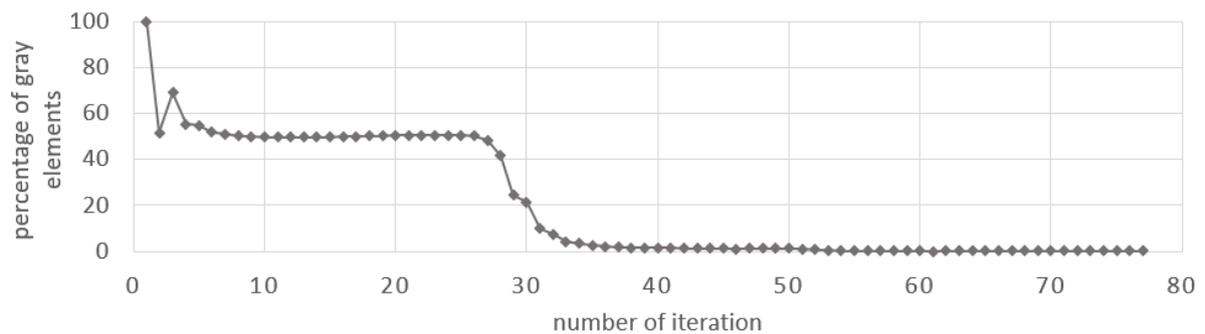


Figure 24. Percentage of gray elements at each iteration step for the rocker arm structure.

6. Concluding Remarks

The topology algorithm that benefits from the adaptation of sorted compliances optimization (TABASCO) has been developed. It is built as a Cellular Automaton with original update rules. Building the update rules, the function mapping local compliances distribution has been defined based on their sorted values on the global level, whereas on the local level, for each cell, the compliances have been additionally sorted within the cell neighborhood. The derived algorithm has successfully coped with the generation of

both 2D and 3D topologies. In the latter case, a step has been made towards engineering applications by linking TABASCO with ANSYS software. The implementation of local sorting improved algorithm performance as compared with former approaches, providing results with better objective function values. Moreover, the algorithm is versatile and easy to implement. It can also be underlined that it generates distinct solid–void-resulting topologies without intermediate gray elements. This is thanks to the implementation of the CA approach with local information exchanging among cells forming a neighborhood. This approach practically eliminates the necessity of filtering techniques' implementation. The CA approach also offers the possibility of adaptation to problems with irregular meshes (see, e.g., [48]). To conclude, the above advantages hopefully place TABASCO among other topology generators which are worthy of the attention of designers in the field of structural optimization.

Author Contributions: As far as conceptualization, methodology, software, investigation, and writing the paper are concerned, all authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by the Faculty of Mechanical Engineering, Cracow University of Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Deaton, J.D.; Grandhi, R.V. A survey of structural and multidisciplinary continuum topology optimization: Post 2000. *Struct. Multidisc. Optim.* **2014**, *49*, 1–38. [[CrossRef](#)]
2. Sigmund, O.; Maute, K. Topology optimization approaches. *Struct. Multidisc. Optim.* **2013**, *48*, 1031–1055. [[CrossRef](#)]
3. Liu, J.; Gaynor, A.T.; Chen, S.; Kang, S.; Suresh, K.; Takezawa, A.; Li, L.; Kato, J.; Tang, J.; Wang, C.C.L.; et al. Current and future trends in topology optimization for additive manufacturing. *Struct. Multidisc. Optim.* **2018**, *57*, 2457–2483. [[CrossRef](#)]
4. Kentli, A. Topology optimization applications on engineering structures. In *Truss and Frames-Recent Advances and New Perspectives*; IntechOpen: London, UK, 2020. [[CrossRef](#)]
5. Ribeiro, T.P.; Bernardo, L.F.A.; Andrade, J.M.A. Topology optimisation in structural steel design for additive manufacturing. *Appl. Sci.* **2021**, *11*, 2112. [[CrossRef](#)]
6. Logo, J.; Ismail, H. Milestones in the 150-Year History of Topology Optimization: A Review. *Comput. Assist. Methods Eng. Sci.* **2020**, *27*, 97–132.
7. Mirjalili, S.; Lewis, A.; Mirjalili, S.M. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
8. Kaveh, A.; Hassani, B.; Shojaei, S.; Tavakkoli, S.M. Structural topology optimization using ant colony methodology. *Eng. Struct.* **2008**, *30*, 2559–2565. [[CrossRef](#)]
9. Kane, C.; Schoenauer, M. Topological optimum design using genetic algorithms. *Control Cybern.* **1996**, *25*, 1059–1088.
10. Najafabadi, R.H.; Goto, G.T.; Falheiro, M.S.; Martins, C.T.; Barari, A.; Tsuzuki, M.S.G. Smart Topology Optimization Using Adaptive Neighborhood Simulated Annealing. *Appl. Sci.* **2021**, *11*, 5257. [[CrossRef](#)]
11. Luh, G.-C.; Lin, C.-Y.; Lin, Y.-S. A binary particle swarm optimization for continuum structural topology optimization. *Appl. Soft Comput.* **2011**, *11*, 2833–2844. [[CrossRef](#)]
12. Valdez, S.I.; Marroquín, J.L.; Botello, S.; Faurrieta, N. A meta-heuristic for topology optimization using probabilistic learning. *Appl. Intell.* **2018**, *48*, 4267–4287. [[CrossRef](#)]
13. Chen, R. Heuristic algorithm for topology optimization of continuum structure. *J. Discret. Math. Sci. Cryptogr.* **2018**, *6*, 1379–1384. [[CrossRef](#)]
14. Ahn, H.K.; Han, D.S.; Han, S.Y. A Modified Big Bang–Big Crunch Algorithm for Structural Topology Optimization. *Int. J. Precis. Eng. Manuf.* **2019**, *20*, 2193–2203. [[CrossRef](#)]
15. Zhao, Z.; Rui, Y.; Liu, Y.; Liu, Z.; Tu, Z. Topology Optimization of Continuum Structures Based on Binary Hunter-Prey Optimization Algorithm. *Symmetry* **2023**, *15*, 1118. [[CrossRef](#)]
16. Afshar, M.H.; Faramarzi, A. Size Optimization of Truss Structures by Cellular Automata. *Comput. Syst. Sci. Eng.* **2010**, *3*, 1–9.

17. Cortes, H.; Tovar, A.; Munoz, J.D.; Patel, N.M.; Renaud, J.E. Topology Optimization of Truss Structures using Cellular Automata with Accelerated Simultaneous Analysis and Design. In Proceedings of the 6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro, Brazil, 30 May–3 June 2005.
18. Inou, N.; Shimotai, N.; Uesugi, T. A cellular automaton generating topological structures. In Proceedings of the 2nd European Conference on Smart Structures and Materials, Glasgow, Scotland, 12–14 October 1994.
19. Bochenek, B.; Tajs-Zielińska, K. GHOST—Gate to Hybrid Optimization of Structural Topologies. *Materials* **2019**, *12*, 1152. [[CrossRef](#)] [[PubMed](#)]
20. Motosoares, C.A.; Martins, J.A.C.; Rodrigues, H.C.; Ambrósio, J.A.C.; Pina, C.A.B.; Motosoares, C.M.; Pereira, E.B.R.; Folgado, J. Topology Optimization with Stress and Displacement Constraints Using the Hybrid Cellular Automaton Method. In *III European Conference on Computational Mechanics*; Springer: Dordrecht, The Netherlands, 2006.
21. Patel, N.; Renaud, J.; Agarwal, H.; Tovar, A. Reliability Based Topology Optimization Using the Hybrid Cellular Automaton Method. In Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Austin, TX, USA, 18–21 April 2005.
22. Zhang, X.; Wang, D.; Huang, B.; Wang, S.; Zhang, Z.; Li, S.; Xie, C.; Kong, D. A dynamic-static coupling topology optimization method based on hybrid cellular automata. *Structures* **2023**, *50*, 1573–1583. [[CrossRef](#)]
23. Zeng, D.; Duddeck, F. Improved hybrid cellular automata for crashworthiness optimization of thin-walled structures. *SMO* **2017**, *56*, 101–115. [[CrossRef](#)]
24. Afrousheh, M.; Marzbanrad, J.; Göhlich, D. Topology optimization of energy absorbers under crashworthiness using modified hybrid cellular automata (MHCA) algorithm. *SMO* **2019**, *60*, 1021–1034. [[CrossRef](#)]
25. Tajs-Zielińska, K.; Bochenek, B. Multi-Domain and Multi-Material Topology Optimization in Design and Strengthening of Innovative Sustainable Structures. *Sustainability* **2021**, *13*, 3435. [[CrossRef](#)]
26. Deng, X.; Chen, H.; Xu, Q.; Feng, F.; Chen, X.; Lv, X.; Lin, X.; Fu, T. Topology optimization design of three-dimensional multi-material and multi-body structure based on irregular cellular hybrid cellular automata method. *Sci. Rep.* **2022**, *12*, 5602. [[CrossRef](#)] [[PubMed](#)]
27. Setoodeh, S.; Abdalla, M.; Gürdal, Z. Combined topology and fiber path design of composite layers using cellular automata. *Struct. Multidisc. Optim.* **2005**, *30*, 413–421. [[CrossRef](#)]
28. Jia, J.; Da, D.; Loh, C.L.; Zhao, H.; Yin, S.; Xu, J. Multiscale topology optimization for non-uniform microstructures with hybrid cellular automata. *Struct. Multidisc. Optim.* **2020**, *62*, 757–770. [[CrossRef](#)]
29. Da, D.C.; Chen, J.H.; Cui, X.Y.; Li, G.Y. Design of materials using hybrid cellular automata. *Struct. Multidisc. Optim.* **2017**, *56*, 131–137. [[CrossRef](#)]
30. Hassani, B.; Tavakkoli, M. A multi-objective structural optimization using optimality criteria and cellular automata. *Asian J. Civ. Eng.* **2007**, *8*, 77–88.
31. Faramazzi, A.; Afshar, M.H. Application of cellular automata to size and topology optimization of truss structures. *ScientiaIranica* **2012**, *19*, 373–380.
32. Bochenek, B.; Tajs-Zielińska, K. Cellular Automaton Mimicking Colliding Bodies for Topology Optimization. *Materials* **2022**, *15*, 8057. [[CrossRef](#)]
33. Gan, N.; Yao, S.; Xiong, Y.; Hong, X. A hybrid cellular automaton–bi-directional evolutionary optimization algorithm for topological optimization of crashworthiness. *Eng. Opt.* **2018**, *12*, 2054–2070. [[CrossRef](#)]
34. Mashayekhi, M.; Mehdizade, F. Size and topology optimization of truss structures using an enhanced cuttlefish method. *Modares Civ. Eng. J.* **2020**, *20*, 133–144.
35. Evsutin, O.; Shelupanov, A.; Meshcheryakov, R.; Bondarenko, D.; Rashchupkina, A. The Algorithm of Continuous Optimization Based on the Modified Cellular Automaton. *Symmetry* **2016**, *8*, 84. [[CrossRef](#)]
36. Chao, L.; Liang, G.; Jin, Y. Grey wolf optimizer with cellular topological structure. *Expert Syst. Appl.* **2018**, *107*, 89–114.
37. Bureerat, S.; Kunakote, T. Topological design of structures using population-based optimization methods. *Inverse Probl. Sci. Eng.* **2006**, *14*, 589–607. [[CrossRef](#)]
38. Tovar, A. Bone Remodeling as a Hybrid Cellular Automaton Process. Ph.D. Thesis, University of Notre Dame, Notre Dame, IN, USA, 2004.
39. Bendsoe, M.P.; Kikuchi, N. Generating Optimal Topologies in Structural Design Using a Homogenization Method. *Comput. Methods Appl. Mech. Eng.* **1988**, *71*, 197–224. [[CrossRef](#)]
40. Sigmund, O. A 99 line topology optimization code written in MATLAB. *Struct. Multidisc. Optim.* **2001**, *21*, 120–127. [[CrossRef](#)]
41. Bendsoe, M.; Sigmund, O. Material interpolation schemes in topology optimization. *Arch. Appl. Mech.* **1999**, *69*, 635–654. [[CrossRef](#)]
42. Yi, B.; Yoon, G.H.; Zheng, R.; Liu, L.; Li, D.; Peng, X. A unified material interpolation for topology optimization of multi-materials. *Comput. Struct.* **2023**, *282*, 107041. [[CrossRef](#)]
43. Stople, M.; Svanberg, K. An alternative interpolation scheme for minimum compliance topology optimization. *Struct. Multidisc. Optim.* **2014**, *22*, 116–124.
44. Tajs-Zielinska, K.; Bochenek, B. Topology algorithm built as an automaton with flexible rules. *Bull. Pol. Acad. Sci.* **2021**, *69*, e138813.

45. Andreassen, E.; Clausen, A.; Schvenels, M.; Lazarov, B.S.; Sigmund, O. Efficient topology optimization in Matlab using 88 lines of code. *Struct. Multidisc. Optim.* **2011**, *4*, 1–16. [[CrossRef](#)]
46. Tajs-Zielińska, K.; Bochenek, B. Cellular Automata Approach to Topology Optimization of Graded Multi-Material Structures. *Appl. Sci.* **2023**, *13*, 2929. [[CrossRef](#)]
47. Sigmund, O.; Petersson, J. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Struct. Optim.* **1998**, *16*, 68–75. [[CrossRef](#)]
48. Tajs-Zielińska, K.; Bochenek, B. CARMA—Cellular Automata with Refined Mesh Adaptation—The easy way of generation of structural topologies. *Appl. Sci.* **2020**, *10*, 3691. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.