

Solving topology optimization problems using cellular automata and mortar finite element method

Yashchuk Yu. O.¹, Tajs-Zielinska K.²

¹*Ivan Franko National University of Lviv,
1 Universytetska Str., 79000, Lviv, Ukraine*

²*Cracow University of Technology,
al. Jana Pawla II 37, 31-864, Cracow, Poland*

(Received 24 March 2020; Revised 12 May 2020; Accepted 15 May 2020)

Currently topology optimization is widely used by engineers for different practical problems. Researches by different authors offer algorithms of using cellular automata in these problems, and most recent publications introduce a mesh-refinement procedure to decrease numerical efforts. In this article, we propose to apply a mortar finite element method for solving topology optimization problems using cellular automata. This methodology enables us to handle the non-conforming meshes, which can arise in the refinement process. We present a formulation of the algorithm and analyse its computational complexity by applying to a test problem.

Keywords: *topology optimization, finite element method, cellular automata, adaptive mesh refinement, mortar functions.*

2010 MSC: 74P05, 65N30, 65N50, 68Q80

DOI: 10.23939/mmc2020.02.239

1. Introduction

In different engineering problems one can run into a task to choose a design of an element or a part. Sometimes it is not a difficult task, but if, for example, the mechanical behaviour is not obvious, or the material is expensive, then a slight difference in design may result in a significant amount of money, or even leads to a failure of a construction. Then one needs to find a structural design that is optimal for a certain design criterion while satisfying other constraints, i. e. to solve a structural optimization problem [3]. Based on the geometrical feature that is parametrized, the structural optimization problem can be classified into [2]:

- size optimization: parametrizing variable is one-dimensional, e. g. thickness of a plate;
- shape optimization: parametrizing variable represents the boundary of the domain;
- topology optimization: parametrizing variable represents connectivity of a domain.

The last class of problems is the most general and, therefore, the most complicated to solve. Moreover, perfect mathematical solutions sometimes are too difficult to implement in real life. But recent rapid development of 3-D printing technologies provides more possibilities to transform optimal solutions into real objects.

Topology optimization is usually formalized in one of the following two formulations [4]:

- stress constrained problem: minimize volume fraction while satisfying stress constraints;
- minimum compliance problem: minimize compliance for a given volume fraction.

Most of the solving methods are applicable to both formulations, so usually only the minimum compliance formulation is discussed. Current research also considers the minimum compliance formulation.

Since the problem formulation implies void and non-void regions in an initial domain, the problem is discrete by its nature. To transform a discrete problem into a continuous one, Bendsoe [7] offered to introduce some virtual density of a material, which can be in the interval between 0 (no material, i.e. void) to 1 (material is present). Penalization was introduced to minimize regions with undesired values

in the middle of this interval. Nowadays this method is known as SIMP method (Solid Isotropic Microstructure (or Material) with Penalization for intermediate densities) and is widely used for topology optimization.

Inou et al. [8] offered to use the cellular automata (CA) concept for the optimal structural design. CA principle is to consider local neighbourhood to change cell's state. This easily can be used in SIMP procedures. On the other hand, CA discretization naturally can be considered as finite element mesh, and the finite element method (FEM) is the most widely used technique for solving elasticity problems. These benefits led to a variety of topology optimization algorithms, which use FEM, CA and SIMP concepts. Bochenek and Tajs-Zielinska developed this technique to be applied to irregular grids [9], and later introduced an automatic mesh refinement algorithm [10]. In the current article, we extend this methodology by introducing mortar finite elements [14], which allow solving elasticity problems with non-conforming meshes. We present the formulation of the algorithms and the analysis of computational complexity.

2. Topology optimization problem formulation

Let us consider an elasticity problem in a plane strain formulation. An elastic body is defined by domain Ω with boundary Γ . Mass forces f are applied in Ω , and the boundary force densities t are applied at $\Gamma_t \subset \Gamma$. Let us define u to be a displacement vector; $\varepsilon = (\varepsilon_{11}, \varepsilon_{22}, \frac{1}{2}\varepsilon_{12})^T$ is a vector, composed of the constitutive components of the strain tensor; $\sigma = (\sigma_{11}, \sigma_{22}, \frac{1}{2}\sigma_{12})^T$ is a vector, composed of the constitutive components of the stress tensor.

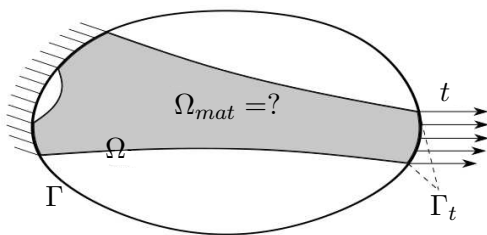


Fig. 1. Topology optimization problem.

The topology optimization problem in the minimum compliance formulation is to find subdomain $\Omega_{mat} \subset \Omega$ of a predefined (normalized) volume $0 < M < 1$ with minimal total compliance under the predefined load. We derive the topology optimization formulation from the standard elasticity problem for the domain Ω . The easy and effective way to define subdomain is to introduce the design variable $\bar{d}(x)$, $x \in \Omega$:

$$\bar{d}(x) = \begin{cases} 1, & x \in \Omega_{mat}, \\ 0, & x \notin \Omega_{mat}. \end{cases} \tag{1}$$

For the proper elastic behaviour of the body we assume that the design variable influences the initial problem through the variable stiffness tensor:

$$\bar{K}(x) = \bar{d}(x) K^*, \tag{2}$$

where K^* is a real, constant stiffness of the material. Then compliance (i.e. strain energy) can be defined as:

$$\bar{C}(\Omega) = \int_{\Omega} u^T \bar{K} u d\Omega. \tag{3}$$

Now we formalize the equilibrium state of the object. Let us denote by B the matrix, which connects the strain tensor components with the displacement [12]

$$\varepsilon = B u. \tag{4}$$

Then we can introduce traditional energy bilinear form and a load linear form:

$$\bar{a}(u, v) = \frac{1}{2} \int_{\Omega} \varepsilon^T(u) \sigma(v) d\Omega = \frac{1}{2} \int_{\Omega} u^T B^T \bar{K} B v d\Omega, \quad l(u) = \int_{\Omega} f u d\Omega + \int_{\Gamma_t} t u d\Gamma_t.$$

A weak form of the equilibrium equation will be the following:

$$\bar{a}(u, v) = l(v), \quad v \in V, \tag{5}$$

where $V = H^1(\Omega)$.

The topology optimization problem in the minimum compliance formulation can be formulated as:

$$\begin{aligned} &\text{find } \bar{d}_0 \in \bar{L}_M(\Omega), \\ &\text{such that } \bar{d}_0 = \arg \min \bar{C}(\Omega), \\ &\text{where } u \text{ satisfies } \bar{a}(u, v) = l(v), \quad v \in V, \end{aligned} \tag{6}$$

where $\bar{L}_M(\Omega)$ is defined as:

$$\bar{L}_M(\Omega) = \left\{ \bar{d} \in L^1(\Omega) : \int_{\Omega} \bar{d}(x) d\Omega = M; \bar{d}(x) \in \{0, 1\}, x \in \Omega \right\}. \tag{7}$$

Since \bar{d} can have only 0 or 1 values, this problem is discrete. As we mentioned in previous section, to transform it into a continuous form, one can use the SIMP. For this aim, instead of $\bar{d}(x)$, we introduce $d(x)$, which can have real values in the interval $[0, 1]$, but also we add penalization for non 0 or 1 values. Similar to (5), linear space for $d(x)$ we give as follows:

$$L_M(\Omega) = \left\{ d \in L^1(\Omega) : \int_{\Omega} d(x) d\Omega = M; 0 < d_{min} \leq d(x) \leq 1, x \in \Omega \right\}, \tag{8}$$

where d_{min} (small floating point number, usually around 0.001) is introduced to preclude stiffness singularities [7].

Penalty is introduced in a natural way, as it is described, e.g., in [2]:

$$K(x) = d^p(x) K^*. \tag{9}$$

Here, p is a penalty parameter. In real problems calculations, it was investigated that $p \geq 3$ provides penalization behaviour of the calculation scheme.

Let us also denote by $a(\cdot, \cdot)$ and C the bilinear form and compliance, derived from K instead of \bar{K} :

$$a(u, v) = \frac{K^*}{2} \int_{\Omega} d^p u^T B^T B v d\Omega, \tag{10}$$

$$C(\Omega) = K^* \int_{\Omega} d^p u^T u d\Omega. \tag{11}$$

In the end, we obtain the following formulation of the topology optimization problem:

$$\begin{aligned} &\text{find } d_0 \in L_M(\Omega), \\ &\text{such that } d_0 = \arg \min C(\Omega), \\ &\text{where } u \text{ satisfies } a(u, v) = l(v), \quad v \in V. \end{aligned} \tag{12}$$

3. Cellular automaton and topology optimization algorithm

As we mentioned in the previous sections, cellular automata are already well-known tools for solving topology optimization problems. Similarly, the finite element method is widely used to solve elasticity problems. To be more precise, considering topology optimization problem in the form (12), CA are applied to line 2 (the minimization part), and FEM is applied to line 3. Let us formalize the CA methodology first.

A continuous-value 2D cellular automaton can be described as (see e.g. [11]):

- a lattice of N cells in 2D space;
- a set of N real values that represent the local state of each cell at each time step;
- a transition rule, that describes evolution of the states in time, based on the states of the neighbour cells.

In the case of application to the topology optimization problem, the whole domain Ω is divided into cells, and the (approximated) design variable is used as the local state value of a cell. The transition rule is the most flexible part here, and the great number of modifications can be created by tuning the rule.

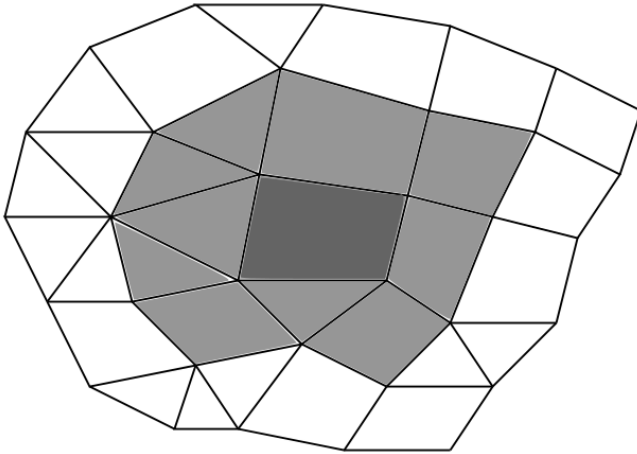


Fig. 2. Neighbourhood illustration.

Let Ω be divided into N cells $\omega_1, \dots, \omega_N$. We denote the set of all the cells as $T = \{\omega_i\}_{i=1, \dots, N}$. Let us define x_1, \dots, x_n to be the nodes of this lattice. Every cell ω_i is accompanied with the neighbourhood (see Fig. 2):

$$T(\omega_i) = \{\omega_j \in T : \omega_j \cap \omega_i \neq \emptyset\}. \quad (13)$$

As one can see, this definition is a form of the Moore neighbourhood [11].

To apply CA methodology, we introduce piecewise constant approximation of the design variable:

$$d(x)|_{\omega_i} \approx d_i. \quad (14)$$

In other words, d_i is an unknown constant design variable value at the cell ω_i . This value will

be subject to CA cell update rule. The rule usually is derived from some global considerations. In our case this consideration is: eliminate local peak compliances in the body. From the nature of the problem we know, that local increasing of stiffness (or, in our case, of design variable) leads to local decreasing of compliance. Thus, our rule is: if compliance in the neighbourhood of the cell is greater than average in the body, then increase the design variable in the cell; otherwise, decrease it. This rule is applied iteratively. Formally, for every cell ω_i the rule is as follows:

$$d_i^{(k+1)} = d_i^{(k)} + \frac{\alpha}{|T(\omega_i)|} \sum_{\omega_j \in T(\omega_i)} \text{sign} \left(\frac{C^{(k)}(\omega_j)}{|\omega_j|} - \frac{C^{(k)}(\Omega)}{|\Omega|} \right). \quad (15)$$

Here $|\cdot|$ denotes the number of cells in a set, or the area of a domain or a subdomain; $\alpha \in (0; 1)$ is some constant step of a change; $\text{sign}(\cdot)$ is a signum function; $\cdot^{(k)}$ denotes the value of the function at iteration number k .

The initial distribution of $d(x)$ we take as uniform: $\forall x \in \Omega : d^{(0)}(x) = M$, thus $d_i^{(0)} = M$, $i = 1, \dots, N$.

The update rule (15) may cause the total volume of the material to deviate from M . To compensate this, after updating all the cells, we normalize the design variable in the domain Ω :

$$d_i^{(k+1)} := \frac{M |\Omega|}{\sum_{j=1}^N |\omega_j| d_j^{(k+1)}} d_i^{(k+1)}. \quad (16)$$

Thus, we arrive at the topology optimization algorithm in the following form.

Algorithm 1 Topology optimization algorithm.

- Require:** M, α, E ;
 1: initialization: uniform distribution of $d(x)$ in Ω ;
 2: **repeat**
 3: solve FEM (i.e. find $u(x)$);
 4: update CA cells (i.e. calculate $d_i^{(k+1)}$ according to (15));
 5: normalize design according to (16)
 6: **until** $\sum_{j=1}^N |\omega_j| (0.5 - |0.5 - d_j^{(k)}|) < E |\Omega|$.

As one can see, the algorithm stops, when design variable is close enough to 0 or 1 almost everywhere in Ω . Constant E corresponds to the level of precision.

4. Finite Element Method for non-conforming meshes

Both CA and FEM methods prescribe discretization of the domain. It is natural to consider finite elements as cells of the automaton and vice versa. We support this concept in an extended form — considering that elements-cells can be of any polygonal form, not only rectangular. Thus in the following we will use the name “finite element” (or just “element”) for the discretization parts, keeping in mind that they are cells of the automaton at the same time. Consequently, lattice of cells and mesh of finite elements denote the same discretization of the domain.

The elasticity problem (line 3 in (12)) is continuous in regard to space variable x . Following the finite element method (see e.g. [13]), to make it discrete, dependent on the values in nodes x_i , we introduce approximation of the functions based on the mesh. Classically it is the locally polynomial continuous approximation. But our aim is to introduce an adaptive topology optimization algorithm, and adaptation prescribes, that different parts of the domain may have different diameter of the mesh. The researcher then has to choose: either to introduce specific procedures to keep the conformity of the mesh, or to use specific modifications of the FEM to be able to utilize non-conforming mesh. We choose the second solution, namely the mortar FEM, described e.g. in [14]. Thus, we assume, that Ω is divided into subdomains $\Omega_i, i = (1, \dots, N^*)$ and inside of each subdomain the mesh is conforming (see Fig. 3). Let us denote by N^i the number of nodes in Ω_i . We denote by Γ_{ij} — interfaces between Ω_i and Ω_j . Within every subdomain Ω_i we associate with every node $x_{ij} \in \Omega_i$ of the mesh the isoparametric FEM basic function ϕ_{ij} [13]. Similarly, within every interface Γ_{ij} with every node $x_{kl} \in \Gamma_{ij}$ we construct one-dimensional polynomial FEM basic functions ψ_{kl} . For exact form of the ϕ_{ij} and ψ_{kl} functions we refer to [15]. Let us also introduce finite-dimensional approximation spaces $\hat{V} = \prod_{i=1}^{N^*} \text{span} \{ \phi_{ij} \}$ and $\hat{\Lambda} = \prod_{\Gamma_{ij} \neq \emptyset} \text{span} \{ \psi_{kl} \}$. Thus, we consider approximation of u in the form:

$$u(x) \approx \tilde{u}(x) = \sum_{i=1}^{N^*} \sum_{j=1}^{N^i} \phi_{ij}(x) u_{ij}, \tag{17}$$

where $u_{ij} = u(x_{ij})$. Similarly, every function $\lambda \in \hat{\Lambda}$ can be given as:

$$\lambda(t) = \sum_{\Gamma_{ij} \neq \emptyset} \sum_{kl: x_{kl} \in \Gamma_{ij}} \phi_{kl}(t) \lambda_{kl}. \tag{18}$$

This approximation may be discontinuous across Γ_{ij} . To ensure optimal approximation properties, according to the mortar FEM, we add a weak continuity condition, which in a discrete form will be:

$$\int_{\Gamma_{ij}} (u^i(t) - u^j(t)) \lambda(t) dt = 0, \quad \forall \lambda \in \hat{\Lambda}. \tag{19}$$

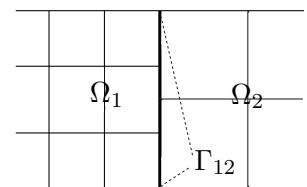


Fig. 3. Interface between two subdomains with internally conforming mesh.

Here $u^i = u|_{\Omega^i}$. We use Lagrange multipliers to implement this condition (see [14] for the details). For this aim we introduce bilinear form:

$$b(\mu, u) = \sum_{\Gamma_{ij} \neq \emptyset} \int_{\Gamma_{ij}} (u^i(t) - u^j(t)) \mu(t) dt, \quad \mu \in \hat{\Lambda}. \quad (20)$$

Since across the interfaces there can arise the discontinuity of the displacements, we need to modify the strain energy bilinear form:

$$\hat{a}(u, v) = \frac{1}{2} \sum_{i=1}^{N^*} \int_{\Omega_i} u^T B^T K B v d\Omega_i. \quad (21)$$

Thus, solving the elasticity problem with mortar FEM is formalized as follows:

$$\begin{aligned} &\text{find } (\hat{u}, \hat{\lambda}) \in (\hat{V}, \hat{\Lambda}) \text{ such that:} \\ &\hat{a}(\hat{u}, \phi_{ij}) + b(\hat{\lambda}, \phi_{ij}) = l(\phi_{ij}); \quad i = (1, \dots, N^*), j = (1, \dots, N^i), \\ &b(\psi_{kl}, \hat{u}) = l(v); \quad \Gamma_{ij} \neq \emptyset, x_{kl} \in \Gamma_{ij}. \end{aligned} \quad (22)$$

Taking into account (17) and (18), problem (22) leads to the system of the linear algebraic equations on u_{ij} and λ_{kl} :

$$\begin{aligned} &\sum_{i=1}^{N^*} \sum_{j=1}^{N^i} u_{ij}(\phi_{ij}, \phi_{rs}) + \sum_{\Gamma_{ij} \neq \emptyset} \sum_{kl} \lambda_{kl}(\psi_{kl}, \phi_{rs}) = l(\phi_{rs}); \quad r = (1, \dots, N^*), s = (1, \dots, N^r), \\ &\sum_{i=1}^{N^*} \sum_{j=1}^{N^i} u_{ij}(\psi_{rs}, \phi_{ij}) = 0; \quad rs: x_{rs} \in \Gamma_{kl}, \Gamma_{kl} \neq \emptyset. \end{aligned} \quad (23)$$

Therefore, considering Algorithm 1, line 3 “solve FEM” means solving system (23), and use (17) to find the approximated value of the displacements function.

5. Adaptive strategy

Since the Algorithm 1 is iterative, the efficient implementation of the FEM is strongly affecting the performance of the Algorithm. One of the methodologies to increase efficiency of the FEM is the mesh adaptation. Generally, there exist three types of the mesh adaptation:

- h-adaptation is local change of the size of the finite element;
- p-adaptation is local change of the order of the approximation functions;
- r-adaptation is local change of the positions of the nodes of the mesh.

H-adaptation proved to be the most effective and is the most widely used. In this paper we investigate only this type, and, therefore, refer to it as just “adaptation”.

Let us recap, that the SIMP procedure prescribes iterative solving of the elasticity problem in the domain, which is a priori larger than the final optimal form. In the void regions there still exists the mesh, and the nodes there increase the size of the system (23). It is natural to aim to keep the mesh in the void regions coarse. The same logic (though to a lesser extent, since we want the approximation to be fine enough) can be used to the non-void regions. Thus, we aim to carry out the mesh refinement in the “border” regions, which are neighbouring purely void and purely non-void regions. Thus we apply the following refinement rule:

$$\begin{aligned} &\text{refine } \omega_i, \text{ if:} \\ &\forall \omega_j \in T(\omega_i): |d_i - d_j| > \eta. \end{aligned} \quad (24)$$

where η is a positive constant.

Thus we arrive to the following adaptive topology optimization algorithm:

Algorithm 2 Adaptive topology optimization algorithm.

Require: M, α, E, η ;

- 1: initialization: uniform distribution of $d(x)$ in Ω ;
 - 2: initialization: coarse mesh in Ω ;
 - 3: **repeat**
 - 4: define subdomains Ω_i with locally uniform mesh;
 - 5: solve FEM (i.e. calculate u_{ij} by solving (23));
 - 6: update CA cells (i.e. calculate $d_i^{(k+1)}$ according to (15));
 - 7: normalize design according to (16)
 - 8: adapt mesh according to (24);
 - 9: **until** $\sum_{j=1}^N |\omega_j| (0.5 - |0.5 - d_j^{(k)}|) < E |\Omega|$.
-

6. Numerical investigation

Numerical investigation of the algorithms was made using the test problem, illustrated in Fig. 4. Material parameters are: Young modulus equals 21000, Poisson coefficient equals 0.3. We use rectangular finite elements with bilinear approximations. The mortar FEM allows using any method of mesh refinement, so we choose the uniform dividing of the rectangular element into four new elements, as shown at Fig. 5. Empirically we take the following values of the parameters: $M = 0.5$, $\alpha = \eta = 0.1$, $E = 0.05$. We solve the problem with both Algorithm 1 and Algorithm 2. For the Algorithm 1, which is non-adaptive, we choose the mesh size empirically. To compare the results, we take this mesh size as the minimal size of the element in Algorithm 2. Thus, if the size of the element is less or equal that this minimal value, no refinement will be made regardless to (24). Convergence of the topologies are illustrated at Fig. 6 and Fig. 7. We see, that both algorithms converge to almost same topology.

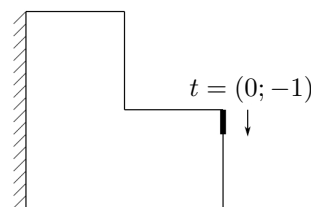


Fig. 4. L-form test problem.

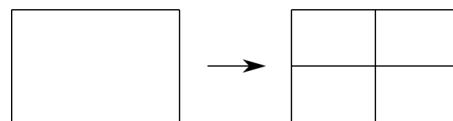


Fig. 5. Mesh refinement principle.

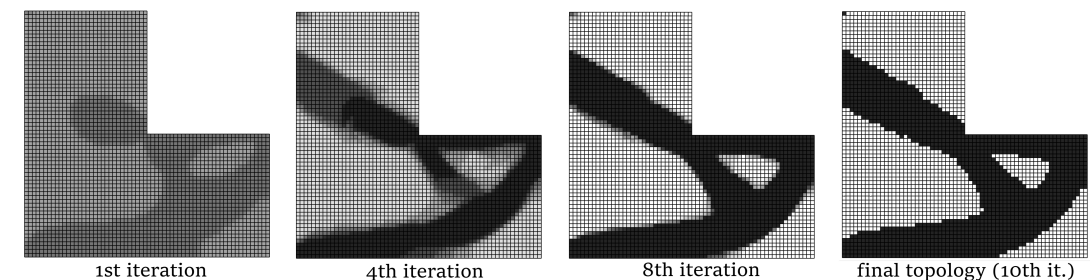


Fig. 6. Topology convergence for the test problem solved with Algorithm 1.

We see, that adaptive algorithm requires more iterations, than the non-adaptive. This is explained by the fact, that on first iterations with coarse mesh the material is redistributed on relatively large areas. Meanwhile in the non-adaptive algorithm the redistribution is relatively precise from the very first iterations.

Total compliance of the object is minimized throughout the process, which one can see from Fig. 8. The curves are very close for the two algorithms. In the end, both algorithms produce topologies with almost same total compliance.

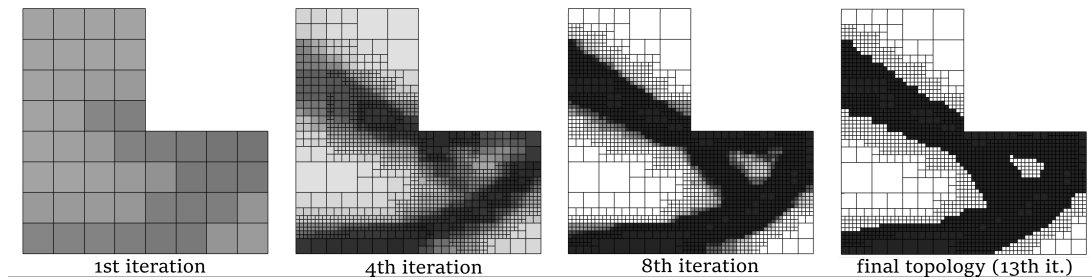


Fig. 7. Topology convergence for the test problem solved with Algorithm 2.

For the computational efficiency investigation we compare total number of degrees of freedom (d.o.f.) in the discretized problems according to both algorithms (Fig. 9). For the adaptive algorithm both FEM basic functions and mortar functions are taken into account. To reach the optimal topology, Algorithm 1 utilizes 62720 d.o.f. in total, whereas Algorithm 2 utilizes 52000 d.o.f. The benefit of adaptivity is not as great as one can expect. The reason is, that, starting from the 4th iteration, both algorithms solve the problem with same order of d.o.f. number. In this situation, the bigger number of iterations for the Algorithm 2 influences greatly the total number of d.o.f. To overcome this problem, the coarsening procedure should be introduced for the mesh neighbourhood with same 0 or 1 value of the design variable. In other words, if in some region all the cells are void, or all are non-void, there is no need for large cells, and the neighbour cells can be united into one bigger cell.

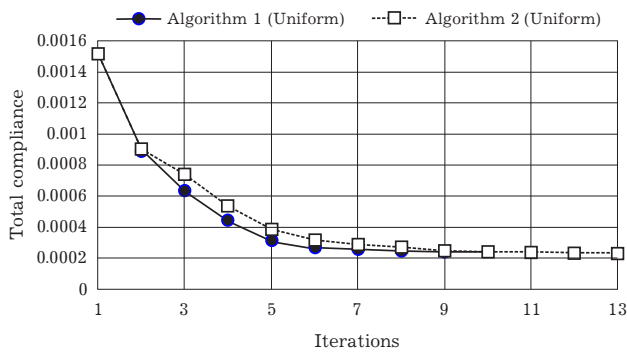


Fig. 8. Total compliance of the L-form domain.

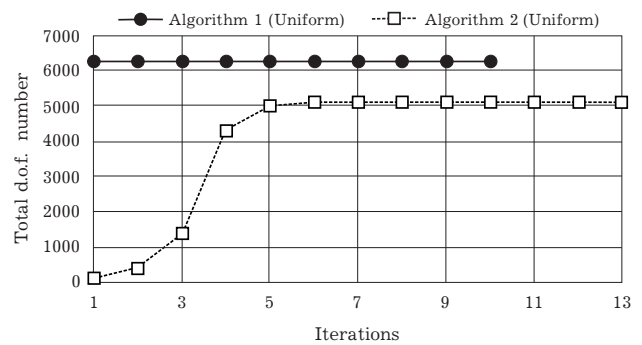


Fig. 9. Total amount of degrees of freedom in the FEM system.

7. Conclusion

The SIMP is an effective in solving topology optimization problems, and can be easily implemented with cellular automata. Naturally combining this methodology with finite element method allows us to use other procedures, like mesh adaptation. The mortar finite element method helps to solve elasticity problems on non-conforming meshes. As we had shown, utilizing the mesh adaptation strategy decreases slightly the total number of degrees of freedom in the problem. However, the effect can be increased if the coarsening procedure is introduced for fully void or non-void regions. The algorithm described in this article can be easily extended to other types of cell neighbourhoods, FEM approximation types, cell update rules, and mesh refinement rules. There is no restriction on the form of the cells (finite elements).

-
- [1] Rozvany G. I. N. Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics. *Structural and Multidisciplinary Optimization*. **21**, 90–108 (2001).
 - [2] Bendsoe M. P., Sigmund O. *Topology Optimization: Theory, Methods and Applications*. Springer, Berlin (2003).

- [3] Leng J. 6 – Optimization techniques for structural design of cold-formed steel structures. *Recent Trends in Cold-Formed Steel Construction*. 129–151 (2016).
- [4] Biyikli E., Albert C. Proportional Topology Optimization: A new non-gradient method for solving stress constrained and minimum compliance problems and its implementation in MATLAB. *PLOS ONE*. **10** (12), e0145041 (2015).
- [5] Xie Y. M., Steven G. P. *Evolutionary Structural Optimization*. Springer (1997).
- [6] Zhou M., Rozvany G. I. N. On the validity of ESO type methods in topology optimization. *Structural and Multidisciplinary Optimization*. **21**, 80–83 (2001).
- [7] Bendsoe M. P. Optimal shape design as a material distribution problem. *Structural Optimization*. **1**, 193–202 (1989).
- [8] Inou N., Shimotai N., Uesugi T. Cellular automaton generating topological structures. *Proceedings of Second European Conference on Smart Structures and Materials*. Vol. 2361, 47–50 (1994).
- [9] Bochenek B., Tajs-Zielinska K. GOTICA – generation of optimal topologies by irregular cellular automata. *Structural and Multidisciplinary Optimization*. **55**, 1989–2001 (2017).
- [10] Tajs-Zielinska K., Bochenek B., Yashchuk Yu. On implementation of adaptive cellular automata based on irregular grids to generation of structural topologies. *Proc. 2nd Int. Conf. Comp. Meth. in Mech.* 80 (2019).
- [11] Chopard B., Droz M. *Cellular Automata Modeling of Physical Systems*. Cambridge University Press (2005).
- [12] Barber J. R. *Elasticity*. Kluwer Academic Publishers (2004).
- [13] Zienkiewicz O. C., Taylor R. L., Zhu J. Z. *The Finite Element Method: Its Basis and Fundamentals*. Butterworth–Heinemann (2013).
- [14] Wohlmuth B. A Mortar Finite Element Method Using Dual Spaces For The Lagrange Multiplier. *SIAM Journal on Numerical Analysis*. **38** (3), 989–1012 (2000).
- [15] Dyyak I., Matysiak S., Yashchuk Yu. FEM-BEM based error estimator in elasticity problems. *Proc. of the 19th Internat. Conf. on Comput. Meth. in Mech. Short Papers*. 175–176 (2011).

Розв’язування задач топологічної оптимізації за допомогою клітинних автоматів та мортарного методу скінченних елементів

Ящук Ю. О.¹, Тайс-Зелінська К.²

¹Львівський національний університет імені Івана Франка,
вул. Університетська, 1, 79000, Львів, Україна

²Краківська Політехніка,
ал. Івана Павла II, 37, 31-864, Краків, Польща

Топологічна оптимізація широко використовується інженерами для розв’язування різноманітних практичних задач. Дослідження різних авторів пропонують застосувати клітинні автомати до цих задач, а у більшості останніх публікаціях додають також процедури згущення сітки для зменшення обчислювальних затрат. У цій статті запропоновано використати мортарний метод скінченних елементів для розв’язування задачі топологічної оптимізації з використанням механізму клітинних автоматів. Це дозволяє розв’язувати задачу на нерівномірній сітці, яка може виникнути в процесі згущення. Подано формулювання алгоритму та проаналізовано його обчислювальну складність на тестовій задачі.

Ключові слова: топологічна оптимізація, метод скінченних елементів, клітинні автомати, адаптація сітки, мортарні функції.