Építés – Építészettudomány 51 (2023) 3–4, 195–213
DOI: 10.1556/096.2023.00094

AKADÉMIAI KIADÓ

Comparative Study on the Optimal Topologies

Bálint Tóth[1] – Matteo Bruggi[2] – János Lógó[3*]

[1] Faculty of Civil Engineering, Budapest University of Technology and Economics, Hungary / Politecnico di Milano, Italy. E-mail: stothbalint@edu.bme.hu

[2] Department of Civil and Environmental Engineering, Politecnico di Milano, Italy. E-mail: matteo.bruggi@polimi.it

[3] Department of Structural Mechanics, Budapest University of Technology and Economics, KM 63, Műegyetem rkp. 3, H-1111 Budapest, Hungary. E-mail: logo.janos@emk.bme.hu

ORIGINAL RESEARCH ARTICLE

Received: 19 March 2023 • Accepted: 20 July 2023
First published online: 20 September 2023

SUMMARY

The topology optimization is a leading tool in structural design. Due to the rapidly spreading need of the industry, commercial software are available in the market. Generally, these software are suitable for solving one subtask (preprocessing, postprocessing, stress calculation, etc.) but need some user manipulation to interconnect to one that is better for some other subproblem. This is the reason why we write a study on the available software and make suggestions on their usability. The purpose of this research is to briefly introduce selected software such as Rhino 3D, Grasshopper, Peregrine, Karamba, Galapagos, polyTop and PolyStress using topology optimization theory. Due to the demand to apply them for industrial applications, the additional goal is to make suggestions to make these software programs more user-friendly and to create algorithms to connect with software used in the industry, such as Consteel. This work also discusses the connected algorithms and optimization methods such as layout optimization by Peregrine, and topology optimization by polyTop and PolyStress. Several illustrative videos are provided as supplements. In addition to the text of this paper one can see demonstrations of the applications by the use of the provided YOUTUBE links.

KEYWORDS

topology optimization, parametric, comparative study, Peregrine, PolyStress, polyTop, application

INTRODUCTION

Topology optimization is a very rapidly developing research field. It can be originated in 1870, however, significant developments have elaborated in the last four decades (see the review papers of Lógó,[1] Lógó and Ismail[2]). It can be applied/used in all fields of life. There are applications in the building industry, biomechanics, and automobile/aerospace industry, and signifi-

[*] Corresponding author.

[1] Lógó 2014.

[2] Lógó–Ismail 2020.

cant activities can be found in the field of fundamental research. It needs significant knowledge in mathematics, mechanics, and informatics.

This research aims to briefly introduce the topology optimization theory and give an overview of selected software, such as Rhino 3D, Grasshopper, Peregrine, Karamba, Galapagos, polyTop, and PolyStress. Due to the demand to apply them for industrial applications, the additional goal is to make these software programs user-friendly and to create algorithms to connect with software used in the industry, such as Consteel. This work also discusses the connected algorithms and optimization methods such as evolutionary algorithms, layout optimization by Peregrine, and topology optimization by polyTop and PolyStress.

Topology optimization in a "continuous" design domain seems to be the most sophisticated method, but in both later presented cases it will yield results that are not trivial to implement in real-world problems given today's tools in the Civil Engineering industry. These methods can be extended also in 3D domains and with this kind of extension actual 3D models can be created which with the development of 3D printing[3] (metal and composite printing) could be used almost directly from such programs with little effort. One can easily see that efficient printing of a 40 m span truss will not be an option soon, but we think that a big problem with optimized structures are the often-complex joints that can perhaps be resolved with 3D printed joints. Such an application was already carried out by the companies MX3D and Takenaka, but it is important to note that the focus there was more on the aesthetics instead of the strength and optimality of the structure *(Fig. 1).*

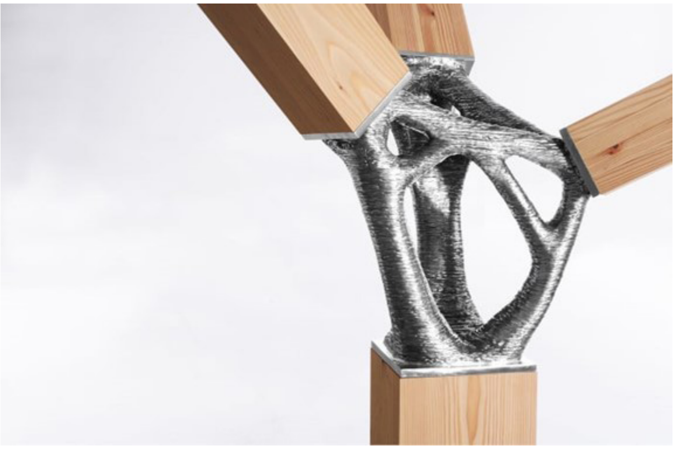


Figure 1. 3D printed steel connection

The structure of this study can be detailed as follows: the first chapter contains the main theory of topology optimization and some important milestones of its history. In chapter two the investigated software are presented. The necessary algorithms and the details of the layout optimization by Peregrine, and topology optimization by polyTop and PolyStress are overviewed in chapter three. Here the computer scripts and workflows developed by the authors of this work are presented. In addition, parametric calculations to get optimal topologies are also discussed

[3] Bruggi–Laghi–Trombetti 2023. 279.

in section four. Several illustrative videos are provided as supplements. In addition to the text of this paper one can see demonstrations of the applications by the use of the provided YOUTUBE links. The entire work is closed with a conclusion section which contains the general comments and the comparative studies.

MAIN THEORY OF TOPOLOGY OPTIMIZATION

Material and energy saving has always been and will be a decisive issue in design. This is especially true nowadays when more and more people become aware of the finiteness of our resources and the footprint of our actions. This is partly the reason why structural optimization is a trending and intensively investigated branch of the research area in engineering.

The topology (shape and material distribution), size of elements, material properties, and mechanical behavior of a support structure are determined by various parameters. Some of these are given, while others are determined by the designer. These design variables stretch the design space. Several requirements must be met during the design process. The geometric functions express the imposed limits on the shape and dimensions of the structure while the limits on the strength, stiffness, and stability are reflected by the behavioral conditions. These are called design conditions and they cut out the design domain from the design space. The mechanical behavior of the structure is characterized by state parameters (stresses, deformations, and displacements) which are required to satisfy the state equations. The goal of optimal design is to choose and define the design variables in such a way that satisfies the state equations and the design conditions while maximizing a favorable property (e.g., load capacity, stiffness, energy absorption capacity) or minimizing a disadvantageous property (e.g., cost, weight, displacement) of the loadbearing structure. All these properties can be specified depending on the design variables. This mathematical expression is called the objective function. The optimal design is a conditional extreme value problem, in which the maximum and minimum of the objective function are sought. The conditions are given in the form of equations and inequalities. In most cases, the objective function expresses the structure’s weight, volume, or cost.

For illustration purposes the fundamental formulation of topology optimization can be given by a constrained mathematical programming problem. By the use of the FEM, let us consider the simple case of a linearly elastic, 2D structure (disk) consisting of ground elements (g = 1, …, G) which has a constant thickness (t_g) of either $t_g = t_{min} = 0$ or $t_g = t_{max} = 1$, such that each ground element (g) contains one or several sub-elements ($e = 1, …, E_s$), whose stiffness coefficients are linear homogeneous functions of the ground element thickness t_g (practically it means that the meshing consists of two parts, a primary part and a secondary one), a single static loading, given boundary conditions and displacement constraints ($d = 1,…, D$).

Using the above-normalized formulation, the structure can easily be transformed to problems with a different prescribed maximum thickness $t_g = t_{max}$. Due to the linear relations this is done by multiplying all loads by t_{max}, whilst stresses, strains and displacements do not alter their value.

The weight (W) of the structure is given by

$$W = \sum_{g=1}^{G} \gamma_g * A_g * t_g \tag{1}$$

where γ_g is the specific weight and A_g the area of the ground element g.

The displacement constraint can be expressed as

$$\hat{\boldsymbol{u}}_d^T \boldsymbol{K}\boldsymbol{u} - \Delta_d \leq 0; \ (d = 1, \ldots, D) \tag{2}$$

where $\hat{u}_d$ is the virtual nodal displacement vector caused by the virtual loads (e.g. "unit dummy loads") associated with the displacement constraints, $\boldsymbol{K}$ is the system stiffness matrix, $\boldsymbol{u}$ is the nodal displacement vector associated with the real load $\boldsymbol{P}$, and Δ_d is the prescribed limit on the displacement value in constraint d. A displacement constraint may refer to some weighted combination of displacements when the virtual loads consist of the corresponding weighting factors. In the case of single displacement limit when the constraint point is at the location of a point load, it is easy to prove that the above inequality constraint (2) can be replaced by the following expression:

$$\boldsymbol{u}^T \boldsymbol{K}\boldsymbol{u} - C \leq 0 \tag{3}$$

where C is the compliance of the structure. In the following the compliance constraint is used for the optimal design.

The side constraints can be stated as

$$\begin{aligned} &-t_g + t_{min} \leq 0; \ (for\ g = 1, \ldots, G) \\ &t_g - t_{max} \leq 0; \ (for\ g = 1, \ldots, G). \end{aligned} \tag{4}$$

In order to *suppress* the intermediate thicknesses, the weight calculation formulation is replaced by $\tilde{W} = \sum_{g=1}^{G} \gamma_g A_g t_g^{\frac{1}{p}}$ (where p is the penalty parameter and $p \geq 1$).

This gives the correct weight value for $t_g = 0$ and $t_g = 1$. The use of the penalty parameter has similar effect in the later formulations as it was the role of the penalty parameter in the classical optimality criteria method.

The optimization problem is to minimize the penalized weight of the structure which is subjected to a given compliance and side constraints.

$$\tilde{W} = \sum_{g=1}^{G} \gamma_g * A_g * t_g^{\frac{1}{p}} = \text{min!}$$
$$subject\ to \begin{cases} \boldsymbol{u^T K u} - C & \leq 0; \\ -t_g + t_{min} & \leq 0; \quad (for\ g = 1, \ldots, G) \\ \ \ t_g - t_{max} & \leq 0; \quad (for\ g = 1, \ldots, G). \end{cases} \tag{5}$$

In eq. (5) the nodal displacement vector $\boldsymbol{u}$ associated with the real load $\boldsymbol{P}$ is calculated from $\boldsymbol{Ku} = \boldsymbol{P}$ linear system.

As it is seen in the problem above one needs an intensive preprocessing part (graphical presentation of the design domain), a suitable connection to FEM program as to a mechanical solver and finally a post processing part to present graphically the results. In the following the selected commercial and open programs are handled in this context.

MILESTONES OF STRUCTURAL OPTIMIZATION

Most papers refer to Michell's optimization paper from 1904 as the starting point but later it was shown that the first optimal solution was published by Maxwell in 1870,[4] which was later extended by Michell.[5] He obtained the optimal layout of a statically determinate truss in the case of a single load case when the absolute value of the axial stress in any bar was not to exceed a given limit. His approach to the solution and the design condition sparked an interest in the 20th century. Next important results in the topic were published by Kazinczy – the inventor of the plastic hinge theory[6] –, whose results unfortunately remained unexplored. He investigated the volume minimization of supporting structures as the subject of economic design and he was among the earliest researchers to consider the problem of statically indeterminate trusses in the case of multiple loading conditions.[7]

The middle of the 20th century was a prosperous period of structural optimization. During this time significant improvements were made in the power and efficiency of optimization techniques in the application of general structural design problems. Some important papers from this era without claiming to be complete are mentioned here. From this time Cox built a structure for the transmission of a bending moment,[8] which was not Michell's optimum structure since it failed to satisfy the orthogonality conditions for members with oppositely signed loads. His solution also indicated that optimal solutions are not necessarily unique in the case of Michell's structures. The derivation of Cox's optimal layout was presented by Chan in 1960.[9] In 1973[10] Shield declared appropriate necessary conditions for the structural volume to be stationary with the help of variational calculus. In 1970[11] Kozlowski and Mróz investigated the uniqueness theorem in topology design where they presented a general formulation of the optimal design problem with geometric constraints.

Achievements in the last three decades start with Bendsoe and Kikuchi's results, who created a new generation of topology optimization problem formulation by using homogenization in 1988.[12] After their publication new procedures and computational concepts started to emerge such as the density-based method, which is often labeled as SIMP (Solid Isotropic Material with Penalization)[13] or Evolutionary approaches.[14] In today's age some mentionable results have been published by Bruggi about the solution of the checkerboard problem[15] or in the field of stress constrained topology optimization.[16]

[4] Maxwell 1870. 1–40.

[5] Michell 1904. 589–597.

[6] Kazinczy 1914. 101–104.

[7] Kazinczy 1942.

[8] Cox 1958.

[9] Chan 1960. 1–40.

[10] Shield 1973. 13–37.

[11] Kozlowski–Mróz 1970. 1007–1021.

[12] Bendsøe–Kikuchi 1988. 197–224.

[13] Zhou–Rozvany 1991. 309–336.

[14] Zhou–Rozvany 1992. 250–262.

[15] Bruggi 2008. 1819–1829.

[16] Bruggi–Duysinx 2012. 369–384.

ADAPTED COMMERCIAL SOFTWARES

In this section, a short introduction is given about the software adapted in this paper, which are based on the achievements mentioned in the previous section. The selection criteria for this purpose were to be versatile in their application and to show a range of possibilities in structural optimization. This is not a thorough description of them, most of them are capable of more complex duties but the focus here is to show their use in a specific way.

Rhinoceros 3D is a commercial general 3D modelling and computer-aided design (CAD) software that was developed by Robert McNeel & Associates.

Geometry in the software is based on the NURBS mathematical model that focuses on a mathematically precise representation of curves and freeform surfaces in computer graphics, but it is capable of modelling polygon meshes too. Rhino is used in architecture, industrial design, and civil engineering, just to name a few. The software also supports two scripting languages, Rhinoscript (based on VBScript) and Python, the advantage of which we also used for tasks discussed later.

Grasshopper (GH) is a built-in (Rhino 5 and up) graphical algorithm editor for Rhinoceros. It is tightly integrated with Rhino's modelling tools. It requires no previous knowledge of programming, but it supports Python, VBScript, and C# for scripting if needed. GH allows users to create an algorithm-based generator of any geometry from the simple to the mind-boggling ones. Behind these graphical commands, a fair amount of code is pre-written for the program to execute and give an output in case it gets the required input.

Peregrine[17] is a robust structural layout optimization plugin for GH. It can be used to determine efficient (minimum volume) layouts of elements forming a truss, for a given design domain and set of loads, supports, and material properties. It also includes a variety of tools with which the efficiency and practicality of the generated designs can be enhanced.

Peregrine was created in association with the University of Sheffield, Arup, and other partners as part of the UK government-founded BUILD-OPT research project.

Karamba[18] is a finite element program embedded in the parametric environment of Grasshopper for analyzing behavior of structures under external loads. Although it was developed in the structural engineering field it was originally created to be used by architects. The goal of development was to have a program that is fast, interactive, user-friendly, and provides a seamless flow of data between the geometric and structural models. The code being the spine of Karamba through its smart solutions could reduce computation time by 2–3 orders of magnitudes compared to off-the-shelf FE software at that time it was first published, and it still has a significant advantage in this even today. The listed properties make it easy to combine parametric geometric models, finite element calculations, and optimization algorithms like Galapagos.

Galapagos is an evolutionary solver embedded in Grasshopper. In case of interest, one can read a detailed description on David Rutten's blog[19] about how genetic algorithms exactly work in Galapagos. Here is a short summary of their advantages and disadvantages:

[17] Preisinger 2013. 110–113.

[18] Bollinger–Hofman–Preisinger 2010.

[19] Rutten https://ieatbugsforbreakfast.wordpress.com/2011/03/04/epatps01/ (Accessed 14 October 2022).

The strong benefits of these algorithms are as follows:

- Genetic algorithms are exceptionally flexible and can be put to use for a large variety of problems.
- These algorithms are fairly indulgent and will take on poorly formulated, under or over-constrained problems too. They always provide a better solution as time passes which will most likely be better than the intuition.
- Generally they can offer a strong interaction with the user. Their process is transparent and browsable.

Known drawbacks are:

- They are time and resource-consuming. This is especially true in the case of complex set-ups that need a longer time to even do one iteration.
- These algorithms are highly dependent on tuning parameters and given that, they do not guarantee a global minimum or even a solution unless a predefined objective value is specified.
- Consteel is a commercial FEA software focused on analysis and design for steel, composite, and reinforced concrete structures.
- Pangolin is a plugin built for data exchange between Consteel and Grasshopper.

ADAPTED OPEN SOURCE SOFTWARES

From time to time open-source educational software are published in the journal *Structural and Multidisciplinary Optimization*. These software are intended for MSc and PhD level students and are mostly written in Matlab. They are carefully explained and easy to manipulate. Mentioned parameters align precisely with the goals of this paper and for this reason 3 of these educational codes are briefly introduced.

The first one is PolyMesher,[20] an open-source educational software for polygonal mesh generation *(Fig. 2)* that relies on an implicit description of the domain geometry. The main focus of the program is to provide the input needed for finite element and optimization codes that use linear convex polygons. It has already been shown that polygonal discretization is significantly less affected by numerical instabilities (e.g. Checkerboard patterns) in contrast to triangular and quadrilateral meshes. Also, the use of polygonal elements makes possible meshing of complicated geometries with a self-contained Matlab code. In the original paper one can find worked-out examples and a thorough description of the theory behind the undoubtedly compact code.

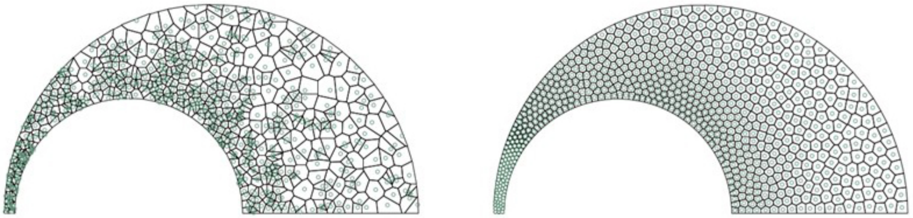


Figure 2. Voronoi tessellation with the original seed (left) and the mesh after Lloyd's iteration (right)

[20] Talischi et al. 2012a. 309–328.

The second code from the above-mentioned journal is polyTop.[21] This is an efficient Matlab code for topology optimization including a general FE routine that is based on isoparametric polygonal elements that can be viewed as the extension of linear triangles and bilinear quads.

The structure of the code is created with the intention to be easily manipulated, meaning in this case that the optimization algorithm and the finite element routine is separated from the choice of the formulation of the topology optimization.

The third code from the above-mentioned journal is PolyStress.[22] This is also a Matlab implementation of topology optimization. Specialties of this code are the use of local stress constraints considering linear and material nonlinear problems. In it a Newton–Raphson scheme is implemented to solve the nonlinear elastic problem. This can handle nonlinear material models with a given strain energy density function. We adopt a system for solving the stress-constrained problem based on the augmented Lagrangian method that treats the issue consistently with the local definitions of stress without employing traditional constraint aggregation techniques. The original publication discusses everything in more detail.

PARAMETRIC STUDY

In this section results of various methods and algorithms will be presented. One common initial geometry (IG) and load case is tested with all the methods/software used in this paper, which is a 40-by-5 rectangle with supports at the bottom edges and a "concentrated" load on the top middle of the geometry *(Fig. 3)*.

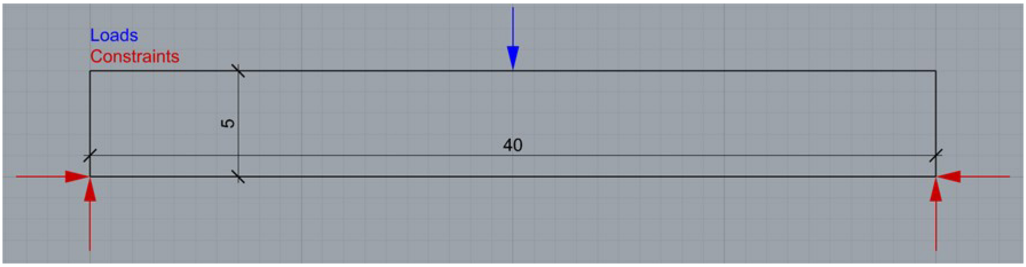


Figure 3. Initial geometry examined in every method below

In some cases, modifications to this geometry or even other shapes are also presented.

MATHEMATICAL CLASSIFICATION OF THE ADAPTED SOFTWARE

In this subsection, a brief mathematical aspect of the selected software is presented. Generally, there are gradient-based optimizers or heuristic based ones. The cited software in this section use a heuristic solver. The main problem of this algorithm is that the solution cannot be named

[21] Talischi et al. 2012b. 329–357.

[22] Giraldo–Paulino 2021. 2065–2097.

as a mathematically optimal one, however, it is called an optimized solution. For the industry it is fair enough, because this optimized solution is the best available solution of the given problem. The only shortcoming is the lack of proof of the mathematical optimality.

Applying this type of optimization method, a parametric model was prepared in Grasshopper which can create a geometry for a Warren or an x-truss with flexible node arrangements with which the evolutionary solver can play around. See the optimized geometries in *Figs 4–5*. A demonstrational video can be seen with the help of the QR code in *Fig. 6*.

Only the self-weight and one concentrated load in the middle of the truss was considered. The goal of the fitness functions is to minimize the maximal deflection of the truss. To create a finite element model and solve the analysis of the truss Karamba 3D was used with a student license.

The evolutionary solver used was Galapagos. It is notable how many different geometries it tries out in the beginning and one can see how it starts to converge and relaxes the rate of changing the variables. By the time as indicated before the solutions regarding the fitness function are becoming better and better but one can use any of the tested geometries and can stop the solver at any time.

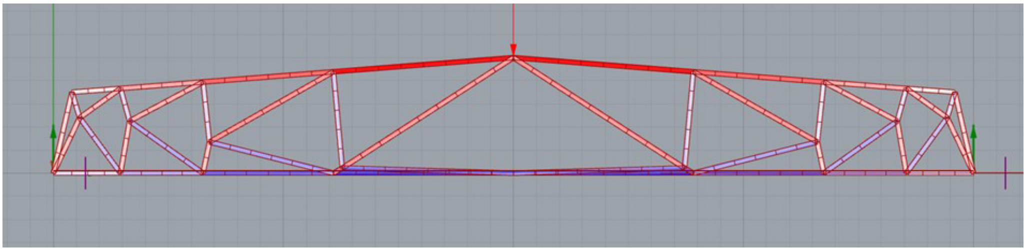


Figure 4. Statically determinate IG solution with Galapagos

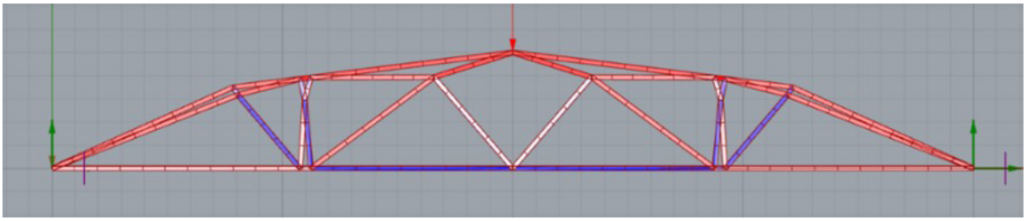


Figure 5. Statically indeterminate IG solution with Galapagos

One can easily see how in this case the formulation of the genes (parameters) can influence the running time of the process and the results. Given the capabilities of the software and the skills of the creator, there can be combinations of genes that are clearly not nearly optimal but because of the nature of evolutionary solvers, they cannot be ruled out. This again pushes the computing time further.

Figure 6. Karamba and Galapagos illustrative video: https://youtu.be/ccU-f6D-FdM

To show the flexibility of these methods another example in less detail is presented where instead of Karamba we have used another external software to solve the truss. With the help of Pangolin, a free Plug-in developed by Consteel for Grasshopper.

It should not be overlooked how much time is needed for one iteration in each case. While Karamba can be faster with orders of magnitude, commercial software can be more advanced in the case of real-world design tasks, so it is a matter of a decision which one to use.

To demonstrate this, we would like to show one more video without deeper details *(Fig. 7)*.

Figure 7. Consteel and Galapagos illustrative video: https://youtu.be/8d0THsY1wEg

This was a small project to test out the possibility of connecting Galapagos and Consteel. What is interesting is the processing time of the two algorithms. While the algorithm where Karamba was used ran for about 6 hours and processed ~36,800 iterations, the one with an external software Consteel only ran for about an hour and it only achieved 280 iterations. This means that the first script needed ~0.6 seconds/iteration while the second script spent between 12–13 seconds with one iteration. The two structures are not exactly the same, but they do not have this much of a difference regarding their complexity. This agrees with the difference in computation speed described by Karamba.

MECHANICAL CLASSIFICATION OF THE ADAPTED SOFTWARE

Based on mechanical aspects there are different fields of topology optimization, such as minimum volume design, full stress design, limit design, shakedown design, etc. Topology optimization for skeletal structures is called layout optimization. Here the used software is suitable for layout design due to its relative simplicity. Here we will introduce an example by the help of Peregrine.

In Peregrine, we need to define an initial layout (user layout) in the later discretized design domain, with which optimized results will be compared *(Fig. 8a–g, Table 1)*. After that, we get a raw optimized solution on which we are able to perform various algorithms to make the solution as feasible as possible. These are as follows:

1. User Layout: it optimizes only the cross sections of the initial layout.
2. Geometry Optimization.[23]
3. Merge: merges joints in a set radius.
4. Simplify: "adds" material to the optimized solution making it simpler.

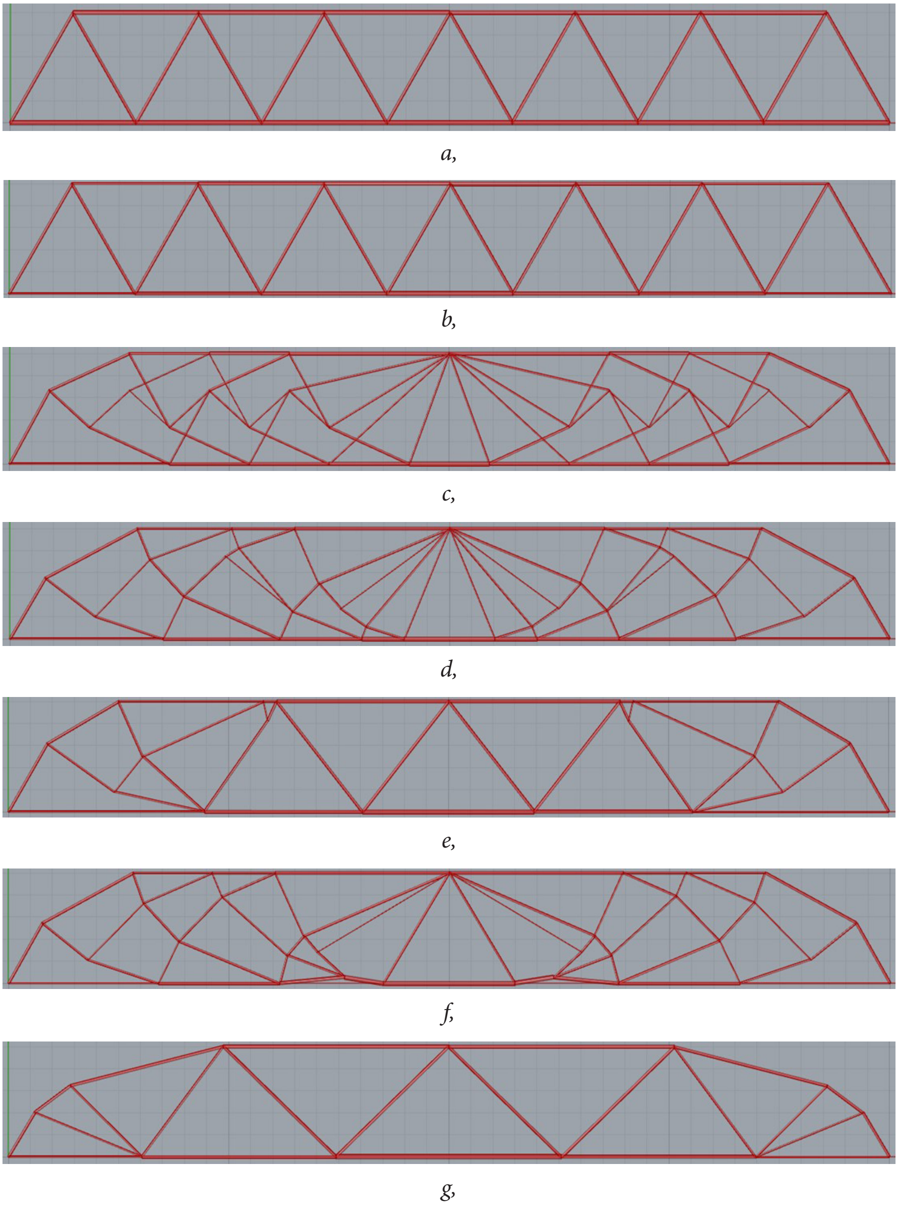


Figure 8. Peregrine initial geometry and its optimized solutions

[23] He–Gilbert 2015. 677–694.

Table 1. Layout optimization results

Sign in *Figure 8*	Applied algorithm(s) (0)	The volume of the structure	Percentage difference in volume
a,	–	0.635 m^3	0 %
b,	1	0.433 m^3	–31.7 %
c,	Raw solution	0.399 m^3	–37.2 %
d,	2	0.394 m^3	–37.9 %
e,	2, 4	0.408 m^3	–35.7 %
f,	2, 3	0.399 m^3	–37.1 %
g,	1, 2	0.408 m^3	–35.8 %

As it is indicated in the introduction, this paper is seeking ways to make these programs more user-friendly and easier to use. Considering this an algorithm was developed both for Karamba and Consteel that extracts required cross-section areas and compares it with the respective software's library and chooses there one that is closest to it. This is obviously not the best solution, the goal of this was to test if such an algorithm could work, but if we formulate better criteria to match sections to the general data, we will be able to fully automatize this kind of workflow. See *Fig. 9* (here different colors indicate different cross sections) and *Fig. 10*.

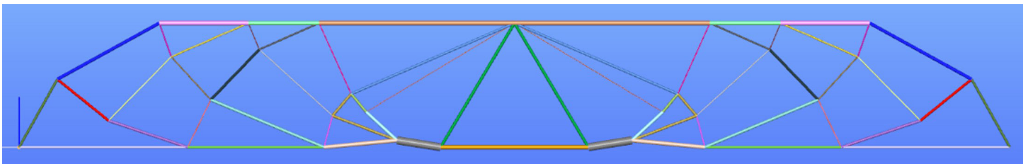


Figure 9. Cross sections in Consteel based on required cross-sectional areas according to Peregrine

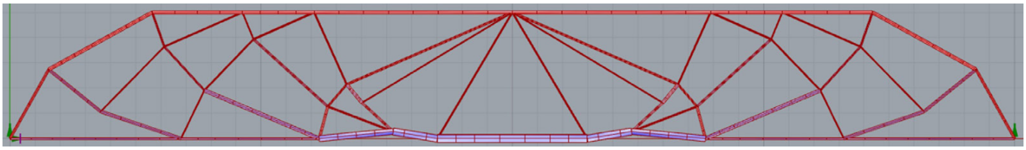


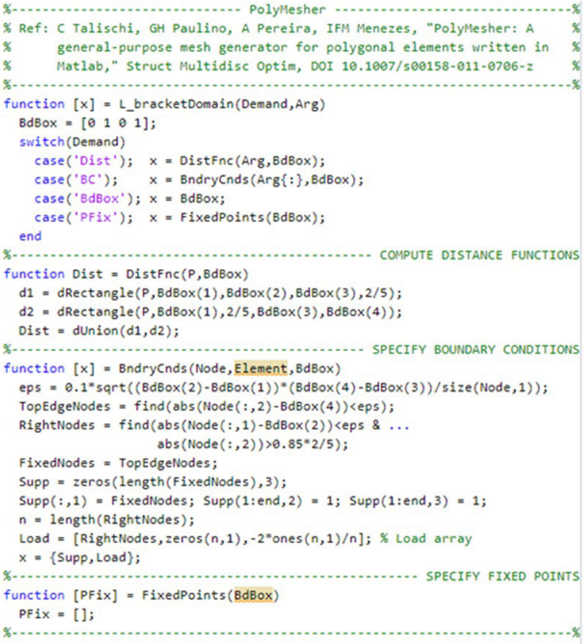
Figure 10. Cross sections in Karamba based on required cross-sectional areas according to Peregrine

MESH GENERATION

The finite element method is the most frequently applied numerical method to solve the optimization problem. The key question is how the discretization is performed. This is the reason why this subsection is started with the mesh generation because the optimal solution very often depends on the level of discretization (it is known as a problem).

Both polyTop and PolyStress need a finite element mesh to run the analysis and the optimization routines. Given the great flexibility these programs offer, one has more possibilities to create the mentioned mesh.

One such intended way is to use PolyMesher to generate a polygonal mesh. It works well but is not necessarily user-friendly if one is not that familiar with Matlab or programming in general. It takes around 30 lines of Matlab code *(Fig. 11)* for a simple definition of an "L-bracket" domain that is needed to generate the polygonal mesh with PolyMesher. The result is in *Figure 12*.



```
%---------------------------- PolyMesher -----------------------------%
% Ref: C Talischi, GH Paulino, A Pereira, IFM Menezes, "PolyMesher: A    %
%      general-purpose mesh generator for polygonal elements written in  %
%      Matlab," Struct Multidisc Optim, DOI 10.1007/s00158-011-0706-z    %
%-------------------------------------------------------------------------%
function [x] = L_bracketDomain(Demand,Arg)
  BdBox = [0 1 0 1];
  switch(Demand)
    case('Dist');  x = DistFnc(Arg,BdBox);
    case('BC');    x = BndryCnds(Arg{:},BdBox);
    case('BdBox'); x = BdBox;
    case('PFix');  x = FixedPoints(BdBox);
  end
%------------------------------------------------- COMPUTE DISTANCE FUNCTIONS
function Dist = DistFnc(P,BdBox)
  d1 = dRectangle(P,BdBox(1),BdBox(2),BdBox(3),2/5);
  d2 = dRectangle(P,BdBox(1),2/5,BdBox(3),BdBox(4));
  Dist = dUnion(d1,d2);
%------------------------------------------------ SPECIFY BOUNDARY CONDITIONS
function [x] = BndryCnds(Node,Element,BdBox)
  eps = 0.1*sqrt((BdBox(2)-BdBox(1))*(BdBox(4)-BdBox(3))/size(Node,1));
  TopEdgeNodes = find(abs(Node(:,2)-BdBox(4))<eps);
  RightNodes = find(abs(Node(:,1)-BdBox(2))<eps & ...
                  abs(Node(:,2))>0.85*2/5);
  FixedNodes = TopEdgeNodes;
  Supp = zeros(length(FixedNodes),3);
  Supp(:,1) = FixedNodes; Supp(1:end,2) = 1; Supp(1:end,3) = 1;
  n = length(RightNodes);
  Load = [RightNodes,zeros(n,1),-2*ones(n,1)/n]; % Load array
  x = {Supp,Load};
%------------------------------------------------------ SPECIFY FIXED POINTS
function [PFix] = FixedPoints(BdBox)
  PFix = [];
%-------------------------------------------------------------------------%
```

Figure 11. Script for the "L-bracket" domain

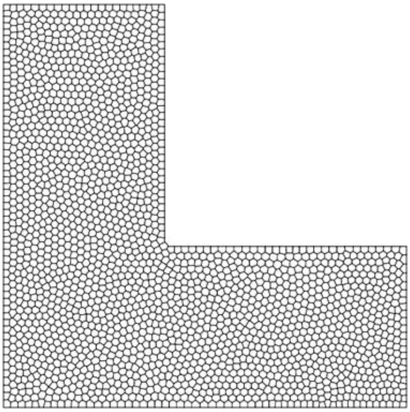


Figure 12. "L-bracket" with polygonal mesh

The goal of this paper is also to give ideas to make these programs more user-friendly and try out methods that can help to create simple workflows suitable for everyday use. Exploiting the earlier mentioned modularity and accessibility of these software we explored the data structure in which PolyStress and polyTop reads the necessary information of the mesh to run the FE analysis and modified their code to be able to read in this necessary information from external text files. The aim of this was to develop a more intuitive graphical user interface based domain creation with the help of our script.

To generate our own data that we wanted to provide to the optimizers we created a routine in Grasshopper that automatically reads in the geometries drawn in Rhino and identifies their layer. For this we refer to an illustrative video that can be seen with the help of *Fig. 13*. With the use of built-in methods, a quad mesh and a triangle mesh can be generated in the drawn domain. The workflow is as follows:

- Domain is drawn in Rhino on Layer 0.
- Geometry is read automatically by Grasshopper.
- With built-in methods quad or triangular mesh can be generated inside Grasshopper.
- After that we need to draw a frame around nodes where we want to have our freedom constraints and loads. X and Y support, and loads have their own layer on which the frame must be drawn. This way Grasshopper can identify where it should put the given information.
- Next step is to name the files and give the export location inside Grasshopper and click the export button on the Python component developed by the authors.
- This will generate 4 text files in one folder which are named according to the label given in the previous section.
- This folder must be copied to the folder of PolyStress and polyTop.
- In the modified scripts, we only need to modify the “filename” variable to the required folders name and run the code.

Figure 13. Rhino-PolyStress illustrative video: https://youtu.be/ABozSKRZzl4

Due to the length limitations a link is provided for exemplary files to try it out *(Fig. 14).* These include a Rhino, the Grasshopper file that creates the mesh and exports the necessary files and the modified PolyStress and polyTop scripts which can read given text files based on the described process.

Figure 14. Example files for the script: https://github.com/KennyKa/CompStudyfroOptTopologies.git

TOPOLOGY OPTIMIZATION WITH POLYTOP

Here the results produced by polyTop for the IG are presented. In these pictures, we can see filtered elements of the mesh. About the filtering function, one can read extensively in the original paper.[24]

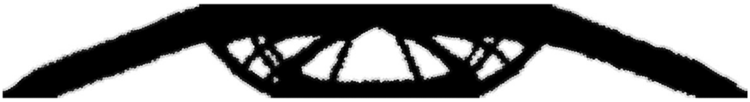


Figure 15. PolyTop Polygonal mesh

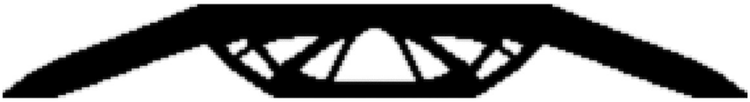


Figure 16. PolyTop Quad mesh

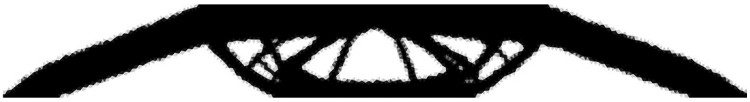


Figure 17. PolyTop Triangle mesh

In *Figs 15–17*, the only difference is in the finite element mesh. Changing it does influence the solutions significantly, and it also takes more time to process a Voronoi mesh. The "cleanest" optimum we get here is when a quad mesh is used *(Table 2).*

Table 2. PolyTop process times

Mesh type	Processing time
Polygonal mesh	156 s
Quad mesh	30 s
Triangle mesh	37 s

TOPOLOGY OPTIMIZATION WITH POLYSTRESS

In a stress-constrained topology optimization problem, we aim to find the lightest structure which can withstand the loads applied to the structure without material failure anywhere in the design domain. In PolyStress certain stress constraints are adopted to achieve this. These are evaluated with von Mises stresses which are calculated based on the Cauchy stress tensor that is

[24] Talischi et al. 2012b. 329–357.

computed from the strain energy density of the solid material. Von Mises stress is a value used to determine if a given material will yield or fracture. It is mostly used for ductile materials, such as metals. PolyStress will plot von Mises stresses, evaluated in the centroid of the discrete elements in the design domain. This can be seen in *Figs 18–20*.

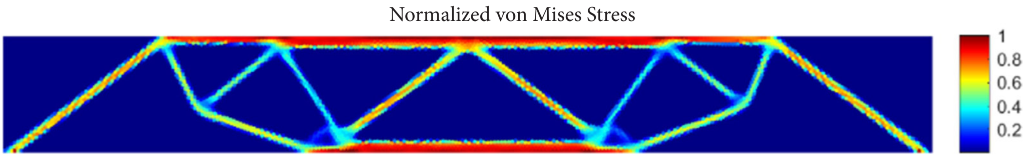


Figure 18. PolyStress Polygonal mesh

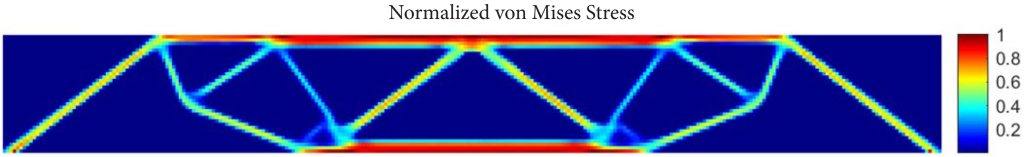


Figure 19. PolyStress Quad mesh

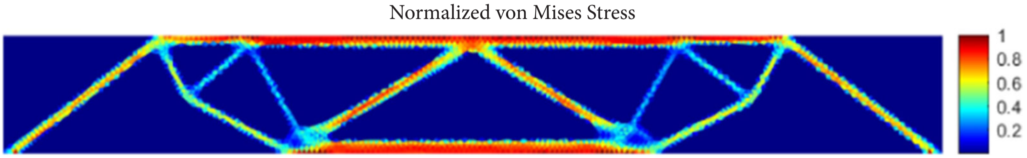


Figure 20. PolyStress Triangular mesh

In terms of the optimized geometry, the difference in the FE meshes does not have a significant effect *(Table 3).*

Table 3. PolyStress process times

Mesh type	Processing time
Polygonal mesh	66 s
Quad mesh	20 s
Triangle mesh	15 s

In terms of process time, PolyStress (at least in these examples) tends to get a solution faster than polyTop.

The results heavily resemble a Warren-type structure *(Fig. 21)* if we modify the height of IG from 5 to 4 meters *(Fig. 22)* or increase the filter radius of the algorithm *(Fig. 23)*, and with a decreased filter radius the solution shows similarities with the layout optimization's results *(Fig. 8d, Fig. 24).*

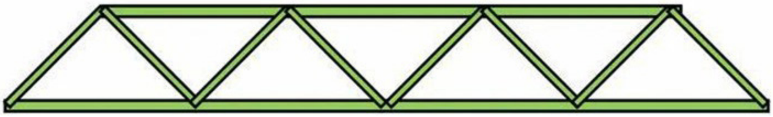


Figure 21. Warren truss

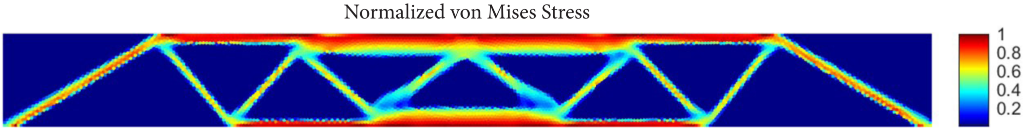


Figure 22. IG with modified height in PolyStress

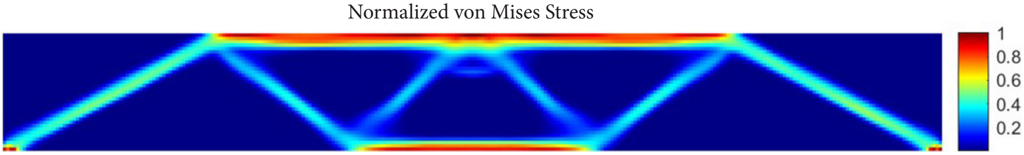


Figure 23. IG with increased filter radius

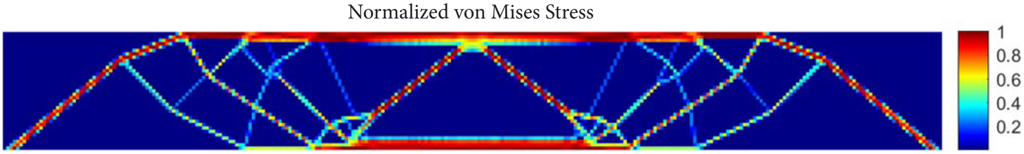


Figure 24. IG with decreased filter radius

CONCLUSIONS

Based on the data gathered for this paper the conclusions are as follows:

The runtime and solution of evolutionary algorithms, because of their dependence on the tuning parameters, have an uncertain running time. The method applied here is highly dependent on how we construct the parametric structure, what are the genes that can be changed, and it is not an easy task to construct a model flexible enough for this kind of optimization and it will never give a mathematically proven global optimum for any structure, but it will always suggest a better solution than the initial solution.

The layout optimization shown with Peregrine seems to give the most easily usable data to work with in exact problems, but these are still just near-optimal structures. The method mentioned in mechanical classification section to find ways to transfer optimized data to industrial design software automatically can be a good direction and it is worth the effort and thought.

Topology optimization in a "continuous" design domain seems to be the most sophisticated method, but in both presented cases it will give out results that are not trivial to implement in real-world problems given today's tools in the Civil Engineering industry. These methods can be extended also in 3D domains and with this kind of extension actual 3D models can be created which with the development of 3D printing (metal and composite printing) could be used almost directly from such programs with little effort.

REFERENCES

Bendsøe, M.P. – Kikuchi, N.: Generating Optimal Topologies in Structural Design Using a Homogenization Method. *Computer Methods in Applied Mechanics and Engineering* 71 (1988) 2. 197–224.

Bollinger, K. – Hofman, A. – Preisinger, C.: *Algorithmic Generation of Complex Space Frames.* [s. l.]. Institute for Architecture, Structural Design, University of Applied Arts, Vienna 2010. Research project funded by the Austrian Research Fund FWF under the translational research programme, grant L538., 2010.

Bruggi, M.: On the Solution of the Checkerboard Problem in Mixed-FEM Topology Optimization. *Computers & Structures* 86 (2008) 19–20. 1819–1829.

Bruggi, M. – Duysinx, P.: Topology Optimization for Minimum Weight with Compliance and Simplified Nominal Stress Constraints for Fatigue Resistance. *Structural and Multidisciplinary Optimization* 46 (2012) 3. 369–384.

Bruggi, M. – Laghi, V. – Trombetti T.: Stress-Based Form-Finding of Gridshells for Wire-and-Arc Additive Manufacturing Considering Overhang Constraints. *Engineering Structures* (2023) 279. 1–15.

Chan, A. S. L.: *The Design of Michell Optimum Structures.* Aeronautical Research Council Reports and Memoranda No. 3303. 1960. 1–40.

Cox, H. L.: *Structures of Minimum Weight: The Basic Theory of Design Applied to the Beam Under Pure Bending*. Aeronautical Research Council Report No. 19791. 1958.

Giraldo, O. – Paulino, G. H.: PolyStress: A Matlab Implementation for Local Stress-Constrained Topology Optimization Using the Augmented Lagrangian Method. *Structural and Multidisciplinary Optimization* 63 (2021) 4. 2065–2097.

He, L. – Gilbert, M.: Rationalization of Trusses Generated via Layout Optimization. *Structural and Multidisciplinary Optimization* 52 (2015) 4. 677–694.

Kazinczy, G.: Kísérletek befalazott tartókkal [Experiments with Clamped End Beams]. *Betonszemle* 2 (1914) 4–5–6. 68–71, 83–87, 101–104.

Kazinczy, G.: Az anyagok képlékenységének jelentősége a tartószerkezetek teherbírása szempontjából [Plastic Behaviour of the Materials Regarding the Load Carrying Capacity of Structures]. In *A Mérnöki Továbbképző Intézet kiadványai* III. kötet, Budapest 1942.

Kozlowski, W. – Mróz, Z.: Optimal Design of Disks Subject to Geometric Constraints. *International Journal of Mechanical Sciences* 12 (1970) 12. 1007–1021.

Lógó, J.: *Szerkezetoptimálás determinisztikus és sztochasztikus esetekben [Structure Optimization in Deterministic and Stochastic Cases].* HAS Doctoral Dissertation. HAS, Budapest 2014.

Lógó, J. – Ismail, H.: Milestones in the 150-Year History of Topology Optimization: A Review. *CAMES* 27 (2020) 2–3. 97–132.

Maxwell, J. C.: On Reciprocal Figures, Frames, and Diagrams of Forces. *Transaction of the Royal Society of Edinburgh* 26 (1870) 1. 1–40.

Michell, A. G. M.: The Limits of Economy of Material in Frames Structures. *Philosophical Magazine* 8 (1904) 6. 589–597.

Preisinger, C.: Linking Structure and Parametric Geometry. *Architectural Design* 83 (2013) 2. 110–113.

Rutten, D.: Evolutionary Principles Applied to Problem Solving. Available at: https://ieatbugsforbreakfast.wordpress.com/2011/03/04/epatps01/ (Accessed 14 October 2022).

Shield, R. T.: Optimum Design of Structures Through Variational Principles. In *Proceedings of Optimization and Stability Problems in Continuum Mechanics*. Springer-Verlag, Berlin, Heidelberg, New York 1973.

Talischi, C. – Paulino, G. – Pereira, A. – Menezes, I.: PolyMesher: A General-Purpose Mesh Generator for Polygonal Elements Written in MATLAB. *Structural and Multidisciplinary Optimization* 45 (2012) 3. 309–328. (2012a)

Talischi, C. – Paulino, G. – Pereira, A. – Menezes, I.: PolyTop: A Matlab Implementation of a General Topology Optimization Framework Using Unstructured Polygonal Finite Element Meshes. *Structural and Multidisciplinary Optimization* 45 (2012) 3. 329–357. (2012b)

Zhou, M. – Rozvany, G. I. N.: The COC Algorithm. Part II. Topological, Geometrical and Generalized Shape Optimization. *Computer Methods in Applied Mechanics and Engineering* 89 (1991) 1–3. 309–336.

Zhou, M. – Rozvany, G. I. N.: DCOC: An Optimality Criterion Method for Large Systems. Part I. Theory; Part II. Algorithm. *Structural Optimization* (1992–1993) 5, 6. 12–25, 250–262.

Összehasonlító tanulmány optimális topológiákra

ÖSSZEFOGLALÓ

A topológia optimálás vezető eszköz a szerkezeti tervezésben. Az iparág rohamosan növekedő igénye miatt kereskedelmi szoftverek is elérhetőek a piacon. Általában ezek a szoftverek egy részfeladat (előfeldolgozás, utófeldolgozás, feszültségszámítás stb.) megoldására alkalmasak, de némi felhasználói manipulációt igényelnek, hogy más részprobléma megoldására alkalmasabb programokkal összekapcsolhatóak legyenek. Ez az oka annak, hogy a rendelkezésre álló szoftverekről tanulmányt készítettünk, és javaslatokat tettünk a használhatóságukra. A kutatás célja, hogy röviden bemutassa a topológia optimálás elméletének felhasználásával a kiválasztott szoftvereket, mint például a Rhino 3D, Grasshopper, Peregrine, Karamba, Galapagos, polyTop és PolyStress programokat. Az ipari alkalmazásuk iránti igény miatt további cél az, hogy ezeknek a szoftvereknek a felhasználóbarátabbá tételére, valamint olyan algoritmusok létrehozására tegyen javaslatot, amelyek az iparban használt szoftverekhez, például a Consteelhez kapcsolódnak. Ez a munka a kapcsolódó algoritmusokat és optimálási módszereket is tárgyalja, mint a Peregrine által végzett hálózatoptimálás, valamint a polyTop és a PolyStress topológiaoptimálása. Számos szemléltető videót mellékelünk. A dolgozat szövege mellett az alkalmazások bemutatói is láthatók a mellékelt YOUTUBE hivatkozások használatával.

KULCSSZAVAK

topológia optimálás, parametrikus, összehasonlító vizsgálat, PolyStress, polyTop, alkalmazás