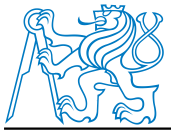


České vysoké učení technické v Praze
Stavební fakulta

Studentská vědecká odborná činnost
Akademický rok 2015/2016

TWO-PHASE OPTIMIZATION OF TILE-BASED TRUSS GROUND STRUCTURE

Jméno a příjmení, ročník a obor: Bc. Marek Tyburec, 5. ročník, C
Konzultant: Doc. Ing. Jan Zeman, Ph.D.
Katedra: mechaniky (132)

Anotace

Tvarová optimalizace příhradových konstrukcí je jednou z nejrozvinutějších disciplín optimalizace konstrukcí. Běžným cílem projektanta je návrh příhradových konstrukcí, které jsou optimální vzhledem ke specifikovaným požadavkům investora. Tato práce zadanou konstrukci rozděluje do omezené sady Wangových dlaždic a zavádí dvouúrovňovou topologickou optimalizaci, ve které je zároveň optimalizován skladebný plán dláždění a průřezová plocha jednotlivých prutů. Výsledkem je konstrukce, která má minimální hmotnost a je prefabrikovatelná.

Anotation

Topology optimization of trusses is one of the most developed field of the structural optimization. The usual aim of a designer is to design trusses that are optimal in some objective(s) required by the investor. This work divides the initial structure into a limited set of Wang tiles and performs two-phase topology optimization of the tile placement and of the cross-sectional areas of individual bars, respectively, securing thus the possibility of prefabrication of the structure and concurrently providing the least-weight structure.

Keywords

Truss topology optimization, plastic formulation, simulated annealing, linear programming, Wang tiles

Contents

1 Introduction

Topology optimization of trusses is one of the most developed field of the structural optimization. The usual aim of a designer is to design trusses that are optimal in some objective(s) required by the investor. Common objectives can be, for example, the least weight or the maximum stiffness.

Typical design variables of the topology optimization are cross-sectional areas of individual bars. Therefore, mostly two approaches are being used: either continuous or discrete optimization.

Discrete topology optimization prescribes a list of possible cross-sectional areas available for each bar. The problem is thus usually solved by some sort of enumeration method (e.g. branch-and-bound algorithm), the computational demands are thus enormous and even 10-bars truss can be a challenge when the guaranteed global optima are to be sought.

The second approach, continuous topology optimization, can be, under suitable assumptions, solved very rapidly, using convex optimization methods. The drawback is that the cross-sectional areas of individual bars in general differ, such trusses could thus be hardly prefabricated which makes the use case limited.

The aim of this paper is to combine the benefits of both the approaches into one, hence making the optimization feasible in a limited time and concurrently allow to prescribe a limited set of continuously varied cross-sectional areas using groups. To ensure the true optimality of the final truss the truss is divided into a limited set of Wang tiles and a two-phase optimization approach is presented. The developed theory is implemented using MATLAB software.

This paper is structured as follows: Firstly, truss topology optimization (the lower-level of the two-phase optimization) will be described and basic properties of the optimal design will be stated. In the subsequent chapter we will introduce an idea of tiling of the initial structure and add groups into the original optimization problem formulation. Also, three different types of initial tiles will be presented. Then, tile placement optimization (the upper level of the two-phase optimization) will be adopted and solved using the method of simulated annealing. The last chapter presents selected illustrative examples.

2 Truss topology optimization

Finding optimal structures has always been a challenging task in the interest of many researchers. The so called structural optimization problems usually try to achieve one of the following objectives: the least-weight structure, the stiffest structure or the structure as insensitive to instability and buckling as possible [Peter W. Christensen, 2009].

Structural optimization can be generally divided into three classes. In the *sizing optimization* the cross-sectional areas of individual members are optimized. The second class, *shape optimization*, considers fixed nodal connectivity and the location of nodes, structural shape, is optimized. The last class, *topology optimization*, the subject of this chapter, is similar to the sizing optimization, it does, moreover, permits the individual members reach a zero cross-sectional area and vanish.

Both the topology and shape optimization allow the designer to find optimal geometry, thus can be also called *geometry optimization*. Simultaneous optimization of shape and topology is called *layout optimization*.

2.1 Michell's optimality criteria

We will now focus on topology optimization of truss structures. The fundamental properties of the optimal grid-like structures in a structural domain D were given by [Michell, 1904], who derived optimality for the least volume trusses subjected to stress constraints and a single loading condition. The cross-sectional areas of individual bars are allowed vary continuously.

Let $\sigma_{\mathrm{T},i}$ and $\sigma_{\mathrm{C},i}$ denote the bounds on the permissible stresses in tension and compression of the ith member, respectively. The longitudinal stress of the ith member σ_i is then constrained by

$$-\sigma_{\mathrm{C},i} \leq \sigma_i \leq \sigma_{\mathrm{T},i}. \tag{2.1}$$

Theorem 1. *The least volume truss subjected to stress constraints and a single load condition is fully-stressed.*

Proof. Let us consider there exist a member j of the truss strictly fulfilling the constraint

$$-\sigma_{\mathrm{C},j} < \sigma_j < \sigma_{\mathrm{T},j}. \tag{2.2}$$

The axial force of the jth member can then be, considering linear elasticity and homogeneous material, computed as

$$s_j = \int_A \sigma_j \mathrm{dA} = \int_A E_j \varepsilon_j \mathrm{dA} = E_j \varepsilon_j \int_A \mathrm{dA} = E_j \varepsilon_j a_j = \sigma_j a_j, \tag{2.3}$$

where E_j denotes the Young's modulus of the jth bar, ε_j stands for the strain of the jth member and a_j is the cross-sectional area of the jth member, respectively. Consequently, the cross-sectional area of the jth member can be stated as

$$a_j = \frac{s_j}{\sigma_j}. \tag{2.4}$$

Based on the limits of the permissible longitudinal stresses it is easy to specify a lower bound on the cross-sectional area of the jth member:

$$a_{\min,j} = \begin{cases} \dfrac{-s_j}{\sigma_{\mathrm{C},j}} & \text{for } s_j \leq 0 \\ \dfrac{s_j}{\sigma_{\mathrm{T},j}} & \text{for } s_j \geq 0 \end{cases}. \tag{2.5}$$

In the case of compressional axial force and after the application of the Equation (2.2) we obtain the inequality

$$\frac{s_j}{\sigma_j} > \frac{-s_j}{\sigma_{\mathrm{C},j}} \Leftrightarrow a_j > a_{\min,j}. \tag{2.6}$$

The same procedure can be similarly applied for the case of tensional axial force

$$\frac{s_j}{\sigma_j} > \frac{s_j}{\sigma_{\mathrm{T},j}} \Leftrightarrow a_j > a_{\min,j}. \tag{2.7}$$

Because the length of the jth bar has not changed, the not-fully-stressed truss simply can not be the least weight, as the cross-sectional area of the jth bar can be further reduced to the corresponding $a_{\min,j}$ value, making thus the truss lighter.

□

Let us denote the column vectors of compressive and tensile axial forces of all bars by $\mathbf{s}_\mathrm{C}$ and $\mathbf{s}_\mathrm{T}$, respectively. The column vectors are prescribed to contain only non-negative numbers, hence the axial forces of all members $\mathbf{s}$ can be expressed as

$$\mathbf{s} = -\mathbf{s}_\mathrm{C} + \mathbf{s}_\mathrm{T}. \tag{2.8}$$

It should be noted that if the members of the truss are manufactured from the same material the least weight optimization is equivalent to the least volume optimization. Based on Equation (2.5) we can thus write the volume of the optimal fully-stressed truss as

$$V = \mathbf{l}^\mathrm{T}(\mathbf{s}_\mathrm{C} \oslash \boldsymbol{\sigma}_\mathrm{C} + \mathbf{s}_\mathrm{T} \oslash \boldsymbol{\sigma}_\mathrm{T}), \tag{2.9}$$

where the symbol $\oslash$ represents element-wise division, $\mathbf{l}$ is a column vector of lengths of all members and column vectors σ_C and σ_T denote the bounds on the maximal allowed stresses in compression and tension of all members, respectively.

The optimal least weight truss also needs to satisfy the equilibrium conditions against the specified static load and there must exist deformations compatible with the strains [Ohsaki, 2011]. Also, the so called *Michell truss* consists of bars arranged in the directions of the principal strains. Because the trajectory of the principal strains is generally not straight (for example see Figure 2.1) the total count of bars can be infinitesimal.

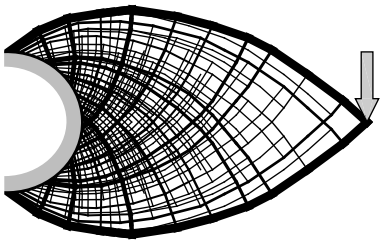


Figure 2.1: Example of Michell truss.

2.2 Ground structure approach

To overcome the problem of infinitesimal count of bars [Dorn et al., 1964] proposed an approximate solution using the so called ground structure approach. The basic idea of the ground structure approach is a discretization of the structural domain D into a finite set of fixed nodes and a set of potential connections between them [Bendsoe and Sigmund, 2004]. Throughout the optimization the continuous cross-sectional bar areas are considered as continuous design variables and they are allowed to take zero values, thus enabling the change of topology by vanishing members.

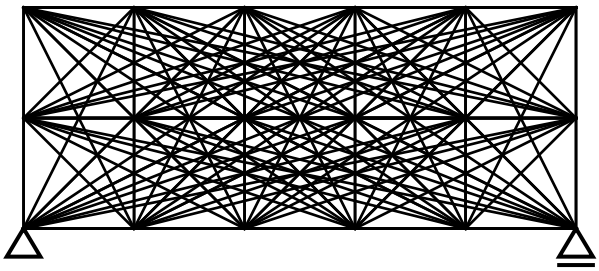


Figure 2.2: Example of highly connected ground structure.

2.2.1 Plastic formulation

In the case when a ground structure is subjected to a single load case, only elastic and linear conditions are considered and the stress bounds are prescribed, the optimization problem can be formulated as a linear program

$$\min V = \min \mathbf{l}^{\mathrm{T}}(\mathbf{s}_{\mathrm{C}} \oslash \boldsymbol{\sigma}_{\mathrm{C}} + \mathbf{s}_{\mathrm{T}} \oslash \boldsymbol{\sigma}_{\mathrm{T}}) \tag{2.10a}$$

$$\text{s.t.: } \left(-\mathbf{A}, \mathbf{A}\right) \begin{Bmatrix} \mathbf{s}_{\mathrm{C}} \\ \mathbf{s}_{\mathrm{T}} \end{Bmatrix} = \mathbf{f} \tag{2.10b}$$

$$\mathbf{s}_{\mathrm{C}}, \mathbf{s}_{\mathrm{T}} \geq 0, \tag{2.10c}$$

where $\mathbf{f}$ denotes a column vector of nodal forces and $\mathbf{A}$ denotes nodal equilibrium matrix. For each ith member only one of the $s_{\mathrm{C},i}$ or $s_{\mathrm{T},i}$ can be equal to a nonzero value, as there can not exist a situation in which a bar simultaneously carries both the tensional and compressional force.

The objective of the plastic formulation is a minimization of the structural weight (see Equation (2.10a) and Equation (2.9)) such that the resulting optimal structure is capable of carrying the prescribed nodal forces $\mathbf{f}$ through bars into fixed supports. The just described condition is actually expressed by the static Equation (2.10b) stating the equilibrium between the nodal forces $\mathbf{f}$ and the axial forces s.

All Equations (2.10) are linearly dependent on the design variables s, hence providing a linear formulation, which can be efficiently solved to global optimality by several available solvers. The basic properties of the optimal trusses are established by the previously described Michell's optimality criteria. Firstly, any optimal truss has to be fully stressed, which we have already proven in the Theorem 1.

The structure needs to be able to carry the nodal forces into supports, therefore the Equation (2.10b) has to be solvable. Let us now look closer on the nodal equilibrium matrix $\mathbf{A}$. The rows of the matrix represent degrees of freedom of the ground structure and the columns correspond to all permissible connections between them. The Equation (2.10) does have a solution if one of the following condition is satisfied:

- The final matrix $\mathbf{A}$ is regular, i.e. the rank of the matrix $\mathbf{A}$ is equal to the number of columns of the matrix $\mathbf{A}$. This condition can be equivalently rewritten to the formulation: Matrix $\mathbf{A}$ is regular if the count of degrees of freedom of the optimal structure is equal to the number bars with non-zero cross-sectional area. Consequently, the solution of the equation $\mathbf{A}\mathrm{s} = \mathbf{f}$ can be uniquely determined and the final structure is statically determinate.

- The final matrix $\mathbf{A}$ is singular, so the rank of matrix $\mathbf{A}$ is lower than the number of columns of the same matrix; or similarly, the final truss is statically indeterminate and the equation $\mathbf{A}\mathrm{s} = \mathbf{f}$ has an infinite count of solutions.

Based on the above theory it can be assumed that the least volume (weight) truss will be statically determinate, as the structure contains lesser bars. We have not proven it, but this statement is generally true for a single load case.

Another important property of the plastic formulation is that the result of the least weight optimization is in the case of a single load case a dual problem of the minimum compliance formulation [Bendsoe and Sigmund, 2004], hence both the formulations are equivalent.

It should be noted, however, that the problem formulation in Equation (2.10) does not include any compatibility or stress-strain relation. For a single load case this condition is not needed, when the optimal solution is statically determinate and thus the stress-strain relation is automatically satisfied. Another reason is that such formulation is not a linear program and thus not so easily solvable to global optimality. In the case of multiple loading conditions and/or prescribed groups of cross-sectional areas the solution is not, in general, statically determinate. The optimal plastic design is then not equivalent to the optimal elastic design, but it can be used as a lower bound of the elastic formulation.

2.2.2 Post-processing

The resulting least-weight truss usually consists of several artifacts. These can be removed by further post-processing, such that reduced-optimal-structure (ROS) is obtained [Zegard and Paulino, 2014, Dorn et al., 1964]:

- Nodes in which only two collinear bars are present are removed and the bars merged to a single long one.
- Nodes in which no bar is present are removed.
- Bars associated with the basic variables of the linear program should have a minimal cross-sectional area $a_{\min} > 0$ to account for imperfections.

3 Tiling of the initial truss

The result of topology optimization presented in the previous chapter is, generally, a globally optimal least weight and minimal compliance truss. It is common that, in case of no symmetry, each bar of the optimal truss has a unique cross-sectional area. Also, the number of bars of the optimal truss is not known in advance. The optimal structure thus can be inappropriate for practical usage – the truss consisting of so many non-repetitive elements that it would be just too time-consuming and uneconomical to build, see for example Figure 2.1.

To overcome the problem of a huge amount of different cross-sectional areas it is possible to divide the initial ground structure into a tile-based grid (see Figure 3.1) composed of a limited set of tiles, so that each tile can be placed multiple times or be unused, making thus the prefabrication of such structures possible.

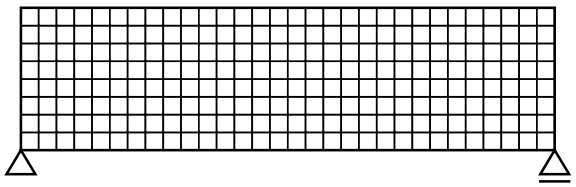


Figure 3.1: Tiled 2-dimensional beam.

Tiles to be placed need to be compatible in a meaning that connected bars and nodes have to be able to carry prescribed static loading into supports. Such structure is then called *statically admissible*.

To achieve statical admissibility one can either use an appropriate predefined tile set containing a number of bars not participating on the transfer of static loads, or perform topology optimization using groups that define bars with equal cross-sectional area. This paper deals with the latter approach.

3.1 Grouping of cross-sectional areas

Let us denote the total count of bars of the ground structure by n_{B} and a group vector assigning group number for each individual bar by $\mathbf{g}$, the length of the group vector is thus n_{B}. Bars which are assigned the same group number share the equal cross-sectional area. Therefore, the total count of cross-sectional areas n_{G} is the number of unique elements in $\mathbf{g}$.

The binary group matrix $\mathbf{G}$ of size $n_{\mathrm{G}} \times n_{\mathrm{B}}$ is then defined as

$$\mathbf{G}(i,j) = \begin{cases} 0 & \text{for } j \neq \mathbf{g}(i) \\ 1 & \text{for } j = \mathbf{g}(i) \end{cases}, \tag{3.1}$$

where $i = \{1, 2, \ldots, n_{\mathrm{B}}\}$. The initial optimization problem (see Equation (2.10)) can thus be rewritten to:

$$\min V = \min \mathbf{l}^{\mathrm{T}}\mathbf{a} \tag{3.2a}$$

$$\text{s.t.: } \left(-\mathbf{A}, \mathbf{A}\right) \begin{Bmatrix} \mathbf{s}_{\mathrm{C}} \\ \mathbf{s}_{\mathrm{T}} \end{Bmatrix} = \mathbf{f} \tag{3.2b}$$

$$\mathbf{G}\mathbf{a} \geq \mathbf{s}_{\mathrm{C}} \oslash \boldsymbol{\sigma}_{\mathrm{C}} + \mathbf{s}_{\mathrm{T}} \oslash \boldsymbol{\sigma}_{\mathrm{T}} \tag{3.2c}$$

$$\mathbf{a}, \mathbf{s}_{\mathrm{C}}, \mathbf{s}_{\mathrm{T}} \geq 0. \tag{3.2d}$$

Note that because of the prescribed groups of cross-sectional areas the resulting optimal structure is not in general statically determinate, hence the optimal plastic design is not generally equivalent to the optimal elastic design and there may not exist compatible displacements fulfilling the prescribed stress limits. This is caused by the fact that the normal stresses can be computed not only based on the equilibrium (static) equation

$$\boldsymbol{\sigma} = \mathbf{s} \oslash \mathbf{a} \tag{3.3}$$

that is used in the plastic formulation, but also from merged physical and geometric equations

$$\boldsymbol{\sigma} = \mathbf{E} \odot \boldsymbol{\varepsilon}_{\mathrm{x}}^{\mathrm{l}} = \mathbf{E} \odot \frac{\mathrm{d}\mathbf{r}_{\mathrm{x}}^{\mathrm{l}}}{\mathrm{d}x^{\mathrm{l}}} = \mathbf{E} \odot \Delta\mathbf{l} \oslash \mathbf{l}, \tag{3.4}$$

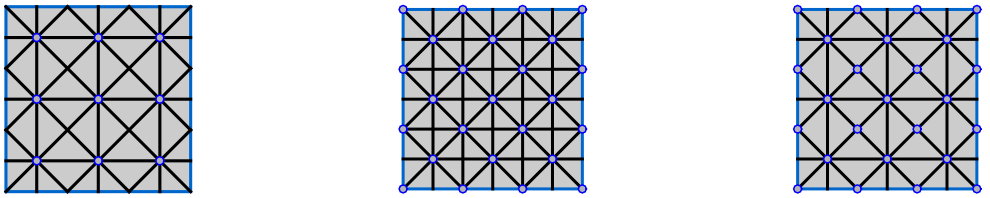
where $\varepsilon_{\mathrm{x}}^{\mathrm{l}}$ represents local normal strain and $\mathbf{r}_{\mathrm{x}}^{\mathrm{l}}$ displacements along the local axis x^{l}, respectively. The symbol $\odot$ represents Hadamard product, i.e. element-wise multiplication.

The resulting optimal structure is commonly statically indeterminate, the count of bars with non-zero cross-sectional area is therefore greater than the rank of the final nodal equilibrium matrix $\mathbf{A}$, which is thus singular; and there does not exist unique solution to the Equation (3.2b). Consequently, the stress vector computed by the Equation (3.3) is also not unique. In reality, the unique internal forces $\mathbf{s}$ and stresses $\boldsymbol{\sigma}$ can be computed based on the stiffnesses of individual members from the Equation (3.4), which is, however, not considered by the plastic formulation, making the optimization problem not so tightly constrained. Subsequently, the result of the plastic formulation creates a lower bound on the optimal elastic solution.

3.2 Tile design

In the previous sections we have introduced an idea to divide an initial structure into tiles (recall Figure 3.1) and it has been theoretically explained how to assign an equal cross-

sectional area to multiple bars. These two assumptions will be used in this section to define three truss tiles used throughout this paper to assemble ground structures.



(a) Tile of type A. (b) Tile of type B. (c) Tile of type C.

Figure 3.2: Initial definition of three truss tiles. The blue circles represent nodes of the initial ground structure and the black lines represent bars.

The most important aspects of the initial tile design seem to be, until the time of writing of this paper, the connectivity of the tiles and the influence of the tile design on the degree of statical indeterminacy of the resulting optimal structure.

Following the former aspect the design of the tile of type A (see Figure 3.2a) has been introduced. The tile consists of 9 uniformly spaced nodes. Considering the height and the width of the tile equal to 1 and coordinates of the centroid of the tile as $(0, 0)$ the nodes are positioned in any combination of the elements $\{-\frac{1}{3}, 0, \frac{1}{3}\}$. The bars are generated in a way that the maximal allowed length of a bar is equal to $\frac{1}{9}$.

A tile of type B (see Figure 3.2b) is very similar to the tile of type A. It contains additional nodes in the place of visual intersections of bars from the former tile type, hence some prescribed nodes are located on the edges and in the corners. The length limit of a bar is again equal to $\frac{1}{9}$. The tile of type B served actually as an intermediate implementation between the types A and C.

The last tile type C, shown in Figure 3.2c, combines the design of the former tiles. While the location of individual nodes is the same as in the tile of type B, the placement of bars, although in most cases split, is based on the type A. The benefit of the tiles of type C is that the resulting optimal structure has lower degree of statical indeterminacy and because the tiles do not contain any only visual intersections they are easily producible.

3.2.1 Completely random tile placement

It has been already written that the connectivity of the tiles is needed to be ensured to produce a statically admissible ground structure. An initial approach can then be proposed:

- Creation of the initial tile set containing n_{W} tiles of the same tile type.
- Random placement of tiles from the tile set into domain of a structure D, such that no blank space is left. Assembly of ground structure.
- Division of bars from which the ground structure consists of into groups based on assumptions: if any of the tiles from the tile set is placed multiple times, all such tiles share the same group vector g. The individual bars exceeding the tile dimensions (also called overlapping bars; see Figure 3.2a) need to share the same cross-sectional area in the both tiles where they appear.

As there exist an optimal solution of the Equation (3.2) it has been proven that this procedure produces statically admissible tiled ground structure. The benefit of this approach

is the relative ease of implementation. On the contrary, the simultaneous existence of overlapping bars and completely random tile placement generally lead to results where the most of the final truss weight is placed in the overlapping bars. Consequently, Wang tiles have been implemented to constrain the placement of tiles.

3.2.2 Wang tiles

The idea of domino-like Wang tiles has been firstly proposed in [Wang, 1961] by Hao Wang, hence the name Wang tiles. Wang tiles are a set of squares, each thus containing four edges. Each edge is assigned an integer value that can be visualized as a specific color. To create a valid tiling the edges shared between tiles need to have the same color. This work uses only binary values: 0 is rendered as white and 1 as blue, see Figure 3.3 as a reference example.

$$\mathbf{C} = \begin{pmatrix} & 0 & & 0 & & 0 & & 1 & & 0 & \\ 0 & & 1 & & 1 & & 1 & & 0 & & 1 \\ & 0 & & 1 & & 0 & & 0 & & 0 & \\ 0 & & 1 & & 0 & & 1 & & 1 & & 0 \\ & 1 & & 0 & & 1 & & 0 & & 0 & \end{pmatrix} \iff$$

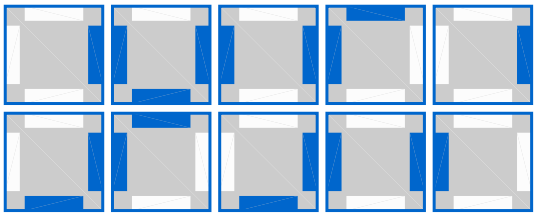


Figure 3.3: Example Wang tiling with the connectivity information stored in edges.

Wang tiles are usually applied in computer graphics [Cohen et al., 2003] because of their ability to tile a plane with aperiodic pattern even with a small tile set, which significantly accelerates rendering of large aperiodic textures [Berger, 1966]. This attribute applies only to special tile sets which are, however, not used in this paper. The usage of Wang tiles is advantageous as it offers us the possibility to define several types of edges and corners.

Similarly to the tiles proposed in [Wang, 1961], where the connectivity information of individual tiles is stored in their edges, the connectivity can be provided by the corners [Lagae and Dutré, 2006], introducing corner tiles. The corner tiles share all the attributes of the classical Wang tiles such as the size of full tile set, but their connectivity is more easily stored, as the connectivity matrix $\mathbf{C}$ is rectangular, for comparison see Figure 3.3 and Figure 3.4.

$$\mathbf{C} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \iff$$

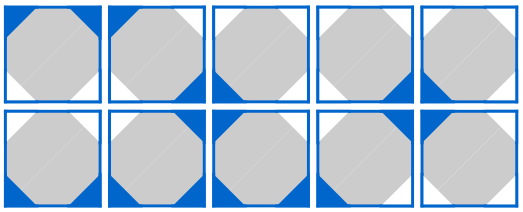


Figure 3.4: Example Wang tiling with the connectivity information stored in corners.

The full tile set of the classical and also corner Wang tiles consists of $4^2 = 16$ tiles. Tile set of corner tiles adopted in this paper is shown in Figure 3.5.

3.2.3 Assigning groups to the bars in tiles

In this section the corner tile set is considered to be composed solely by the tiles of type A (see Figure 3.2a). The same procedure can also be applied for the tiles of type B and C, though. The tiles of type A have been chosen due to the fact they contain all the groups described in this section.

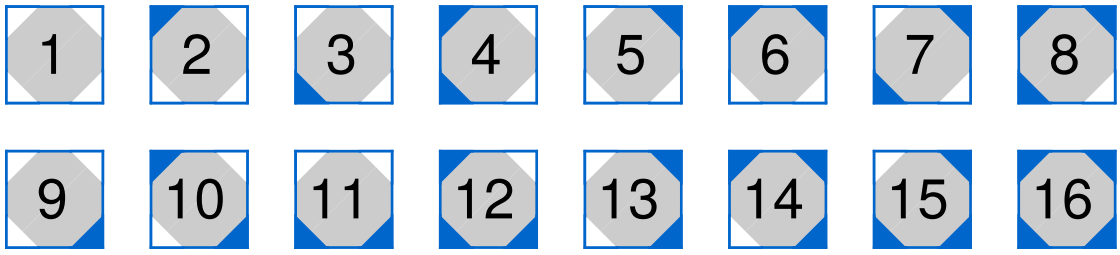


Figure 3.5: Full tile set of corner Wang tiles.

Corner groups

Following the Figure 3.5 we have defined two types of corners of the tiles. The first type is rendered in a white color and the corresponding integer value is 0, the second type is labeled by 1 and drawn in a blue color. Let us now consider there exist bars coming exactly through corners of the tiles, as shown in Figure 3.6, and there does not exist any node placed exactly in the corner. The bars fulfilling this definition will be referred to in the following text as corner-associated bars.

Figure 3.6: Corner groups of the tile set containing tiles of type A.

Based on the definition of Wang tiles and the requirement of fully-connected ground structure it is clear that the corner-associated bars are bound to the specific corner type. The count of corner groups is thus equal to the sum of unique corner-associated bars coming through each corner. The tile set of type A thus includes 4 groups, as clearly shown in Figure 3.6.

Edge groups

Because there exist two types of corners represented by 0 and 1, respectively, it is possible to create 4 types of vertical and 4 types of horizontal edges, as a 2-combination of 0 and 1 values. All the possible combinations are shown in the Figure 3.7.

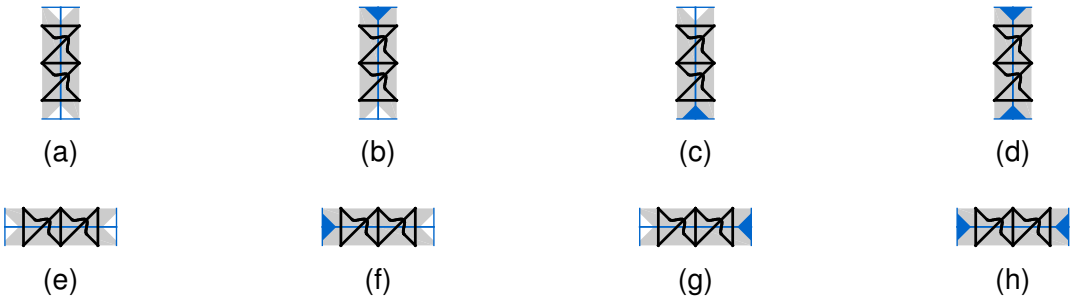


Figure 3.7: Edge groups of the tile set containing tiles of type A.

Similarly to the definition of corner-associated bars the edge-associated bars are bars coming exactly through one edge (i.e. they can not come through corner) and concurrently any of nodes associated with the bars are located exactly on the edge of a tile.

Consequently, the count of the edge groups is equal to the sum of the unique edge-associated bars coming through each type of edge, hence in the considered example the count of edge groups is equal to $7 \cdot 8 = 56$, see Figure 3.7.

Tile groups

We have already defined corner-associated and edge-associated bars. The remaining bars, from which the tile consists of, will be named as tile-associated bars, because they are bound only to the specific tile.

The full tile set includes 16 tiles, see Figure 3.5, hence the count of tile groups is equal to sum of the unique tile-associated bars in each tile. Because the tile of type A consists of 20 tile-associated bars the count of tile groups in the considered example equals to $16 \cdot 20 = 320$, see Figure 3.8 for reference.

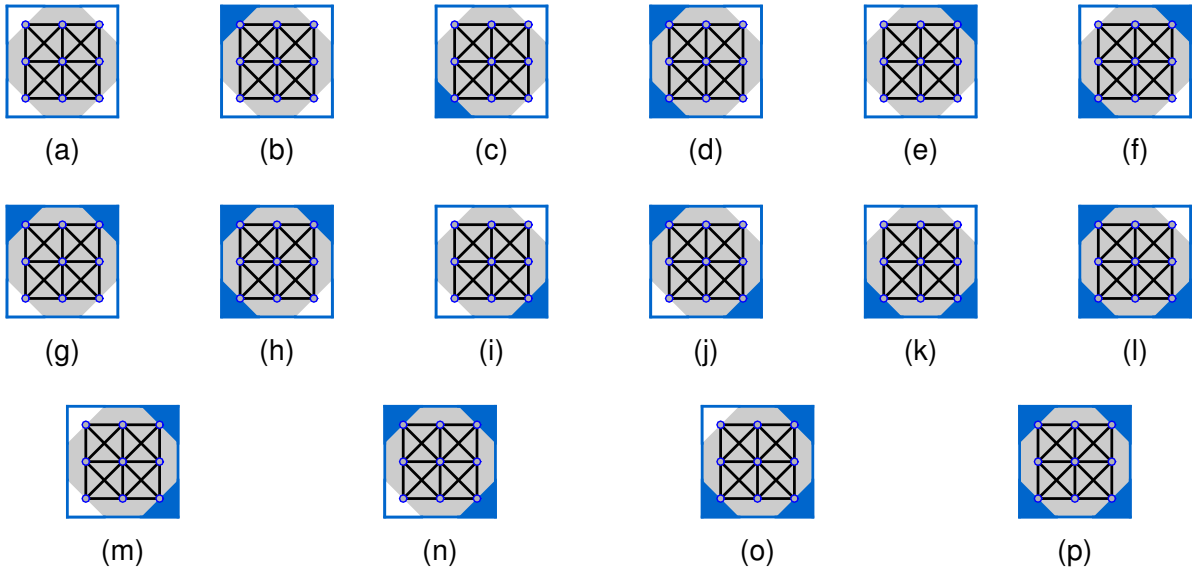


Figure 3.8: Tile groups of the tile set containing tiles of type A.

The total count of groups for the treated tile set thus equals to $4 + 56 + 320 = 380$.

4 Two-phase optimization

The optimization of a tiled truss can be divided into two levels: the lower level, where global topology optimization is launched on a given ground structure, see Equation (3.2); and the upper, where the tile placement is optimized. The tile placement optimization, the main subject of this chapter, is really important, as the difference of volume of the optimal truss between the best (usually not known in advance) and the worst tiling is significant (for examples see Chapter 5).

4.1 Simulated annealing

Optimization tasks are in general divided into two categories [Cook, 1971] – problems that are solvable deterministically in a polynomial time (class P problems); the rest of problems, which are not deterministically solvable in polynomial time, are of class NP. An example

of the class P problem is linear programming. NP class problems may be represented for example by commonly referenced traveling salesman problem (TSP).

In reality both the above mentioned categories of optimization are common and both are needed to be solved efficiently. In this chapter the non-convex NP-hard tiling problem is considered. One can choose two ways to obtain a solution: either deterministic optimization algorithms with a price of huge computational demands and times or fast heuristic algorithms with a price of approximate or local solution. One of the most commonly used local-search heuristic algorithms is simulated annealing.

4.1.1 Analogy with annealing of solids

Simulated annealing has been firstly introduced in [Kirkpatrick, 1984, Černý, 1985] as an analogy to the physical process of annealing of solids, hence the name simulated annealing.

Annealing is a thermal process that is used to bring a solid into a low-energy state, while improving properties of the solid. The first step is to increase the temperature of the solid such that it melts. The particles, from which the solid consists of, are then arranged randomly. Afterwards, the solid is very slowly cooled, permitting thus the particles to arrange into a minimal-energy grid.

According to this analogy it is possible to state the equivalence between the terminology commonly used in optimization methods and the terminology used in simulated annealing [Salamon et al., 2002]. The state of the solid is described by $\boldsymbol{\omega}$, which is analogous to a feasible solution $\mathbf{x}$ in optimization methods. The aim of the method is to reach a solution with the lowest energy $E = \texttt{energy}(\boldsymbol{\omega})$, similarly to the objective function $f(\mathbf{x})$. The optimal solution is then called a ground state $\boldsymbol{\omega}^*$, which is equivalent to $\mathbf{x}^*$.

4.1.2 Basic simulated annealing algorithm

The inputs to the simulated annealing algorithm are the initial feasible state $\boldsymbol{\omega}_0$, initial temperature T_0, number of temperatures n_T and number of steps n_S. These parameters are set for the particular optimization problem.

Firstly, the algorithm computes the energy E of the initial state $\boldsymbol{\omega}_0$ and enters the iteration cycle. The iteration cycle mimics slow cooling of the system with n_T successive temperatures. The system is then brought for each temperature T into equilibrium by using n_S iterations.

Every iteration a new neighbor state $\boldsymbol{\omega}_\mathrm{T}$ and its energy E_T are computed. Neighbor state in this context means that both the states $\boldsymbol{\omega}$ and $\boldsymbol{\omega}_\mathrm{T}$ have "something" in common, hence the definition of the $\texttt{neighbor}(\boldsymbol{\omega})$ function is mostly dependent on the optimization problem. The definition of neighbor function used in this work is mentioned in the next section.

Because the aim of simulated annealing is to minimize the energy, it is checked whether the neighbor-state energy E_T is lower than the energy E. If the answer is positive the state will be accepted, otherwise, the solution might be accepted within some probability, which is dependent on the temperature T. Initially, when the temperature is high, almost all the states with higher energies are accepted. As the temperature decreases the probability to accept states with higher energies also decreases. Therefore, the algorithm, unlike gradient methods, avoids being trapped in a local optimum even for non-convex problems and with correct settings converges to the global optima.

After all the n_S iterations have been performed the temperature is decreased accordingly to the cooling schedule predefined by function $\texttt{cooling}(T)$. The algorithm terminates when the temperature T has been $(n_\mathrm{T} - 1)$ times cooled. Based on the algorithm settings

and the number of iterations either local or global optimum is obtained. The Algorithm 1 shows the pseudo-code of simulated annealing.

Algorithm 1 Simulated annealing

```
function SIMULATEDANNEALING(ω_0, T_0, n_T, n_S)
    ω ← ω_0                                          ▷ Get initial state
    T ← T_0                                          ▷ Get initial temperature
    E ← energy(ω)                                    ▷ Compute initial energy
    for i ← 1, n_T do                                ▷ For the count of decreasing temperatures
        for j ← 1, n_S do                            ▷ For the steps while the same temperature
            ω_T ← neighbor(ω)                        ▷ Compute trial neighbor state
            E_T ← energy(ω)                          ▷ Compute energy of the trial neighbor state
            ΔE ← E_T − E                             ▷ Compute energy improvement
            if ΔE ≤ 0 then                           ▷ If the system energy is lower
                ω ← ω_T                              ▷ The state is always accepted
                E ← E_T                              ▷ And energy is updated
            else                                     ▷ Otherwise
                if rand(1) < exp(−ΔE/T) then         ▷ Accepted based on probability
                    ω ← ω_T
                    E ← E_T
                end if
            end if
        end for
        T ← cooling(T)                               ▷ Cool the temperature
    end for
    return ω
end function
```

4.1.3 Settings used for the two-phase optimization

Tiling of a structure composed by a full set of Wang corner tiles is described by a binary connectivity matrix C, see Figure 3.4 for an example. The connectivity matrix thus describes the state of simulated annealing algorithm. The initial connectivity matrix characterizing the ground state is created as a random binary matrix of specified dimensions.

Based on the connectivity matrix the ground structure is assembled and topology optimization is launched using the formulation of the problem defined by Equation (3.2). The topology optimization then yields optimal cross-sectional areas and the value of the objective function – optimal truss volume. The individual optimal cross-sectional areas are not anyhow used by the simulated annealing algorithm, the value of the objective function does, however, represent the current energy of the system. The function $\texttt{energy}(\boldsymbol{\omega})$ is thus equivalent to a solution of the topology optimization with a given ground structure uniquely defined by the connectivity matrix.

Neighbor state selection

Any state of the system can be described definitely by the connectivity matrix C. The neighbor state is achieved in a way that a random element ind of the connectivity matrix is changed to the opposite binary value:

$$\mathbf{C}_{ind} = 1 - \mathbf{C}_{ind} \tag{4.1}$$

See Algorithm 2 for the specific $\texttt{neighbor}(\boldsymbol{\omega})$ function pseudo-code.

Algorithm 2 Neighbor state generation in simulated annealing

function NEIGHBOR($\boldsymbol{\omega}$)
  $\boldsymbol{\omega}_{\mathrm{T}} \leftarrow \boldsymbol{\omega}$
  $\mathbf{ind} \leftarrow \texttt{random\_element\_indices}(\boldsymbol{\omega})$
  $\boldsymbol{\omega}_{\mathrm{T}}(\mathbf{ind}) \leftarrow 1 - \boldsymbol{\omega}_{\mathrm{T}}(\mathbf{ind})$
  return $\boldsymbol{\omega}_{\mathrm{T}}$
end function

Control parameters

The problem of the correct settings of control parameters have been widely addressed for example in [Salamon et al., 2002, Burke and Kendall, 2013, Kirkpatrick, 1984]. The control parameters are however very tightly connected to the specific optimization problem, it is thus not possible to propose a generally valid parameters.

Therefore, the settings of control parameters used in this paper have been developed heuristically based on the previous optimization runs. The parameter defining the number of temperatures n_{T} is computed as

$$n_{\mathrm{T}} = \texttt{round}(\texttt{size}(\mathbf{C}, 1) \times \texttt{size}(\mathbf{C}, 2) \times 1.5), \tag{4.2}$$

where the $\texttt{size}(\mathbf{C}, j)$ function returns the size of jth dimension of the $\mathbf{C}$ matrix and the $\texttt{round}$ function rounds the input to the nearest integer. Parameter prescribing the number of steps n_{S} is computed as

$$n_{\mathrm{S}} = \texttt{round}(\texttt{size}(\mathbf{C}, 1) \times \texttt{size}(\mathbf{C}, 2)/3). \tag{4.3}$$

Cooling schedule and initial temperature

The last and most important parameter is the initial temperature T_0, which can be computed by from

$$T_0 = \frac{E_{\mathrm{worst}} - E_{\mathrm{ideal}}}{50}, \tag{4.4}$$

where E_{worst} represents an energy of the worst-case tiling – i.e. when all the elements of connectivity matrix $\mathbf{C}$ are equal to 0 and thus the structure is tiled from only one tile type. The ideal energy E_{ideal} is obtained by optimizing the ground structure without prescribed groups. The temperature is in every iteration multiplied by 0.95 and the system is thus cooled. Pseudo-code of a cooling function is shown in Algorithm 3.

Algorithm 3 Cooling schedule of simulated annealing

function cooling(T)
  $T \leftarrow 0.95T$
  return T
end function

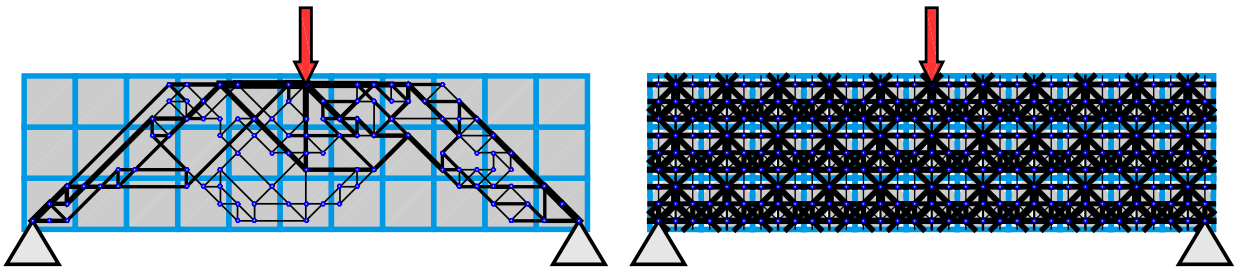
5 Reference examples

This chapter presents three initial examples of the two-phase optimization. All the previously defined tile types have been evaluated on a selected structure, beam tiled in a 11×3 grid. Each tile type is represented by an ideal solution, which has been obtained by launching the topology optimization without specified groups of cross-sectional areas, see Equation (2.10); then by the two optimal results of the two-phase optimization and the worst-case tiling.

The first optimal solution has been obtained by applying a condition that the connectivity matrix $\mathbf{C}$ has to be symmetric, i.e. symmetric tiling of the structure is enforced. The second optimal solution is an ordinary optimal solution obtained by the previously described two-phase optimization approach. The optimization result with enforced symmetry is hoped to be of better quality.

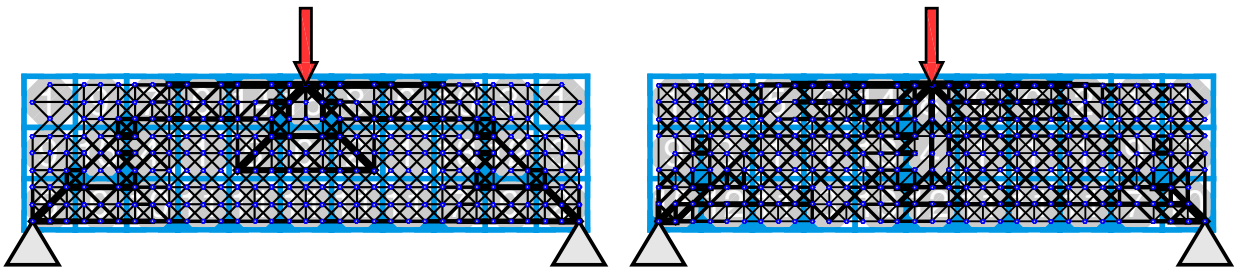
It should be noted that all of the optimal topologies, except for the worst-case examples, have been post-processed.

5.1 Beam tiled with 11x3 tiles of type A



(a) (b)

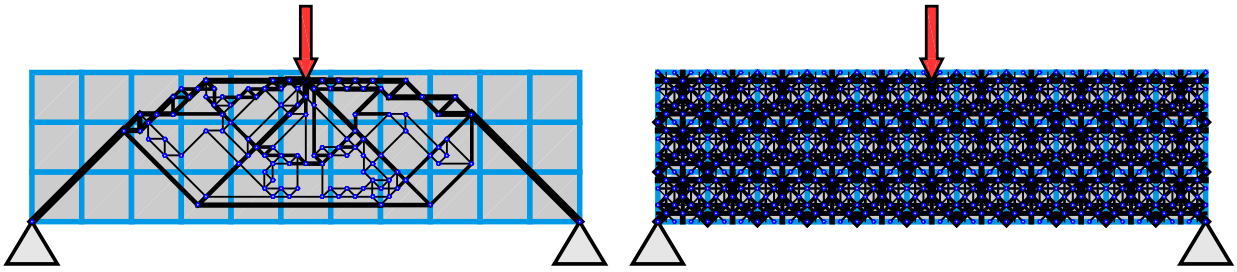
Figure 5.1: The figure (a) shows an ideal non-grouped beam tiled with 11×3 tiles of type A. The ideal volume is $V_{\text{ideal}} = 0.79433$. The latter figure (b) shows the worst case solution of the same beam. The worst volume is $V_{\text{worst}} = 3.0837$.



(a) (b)

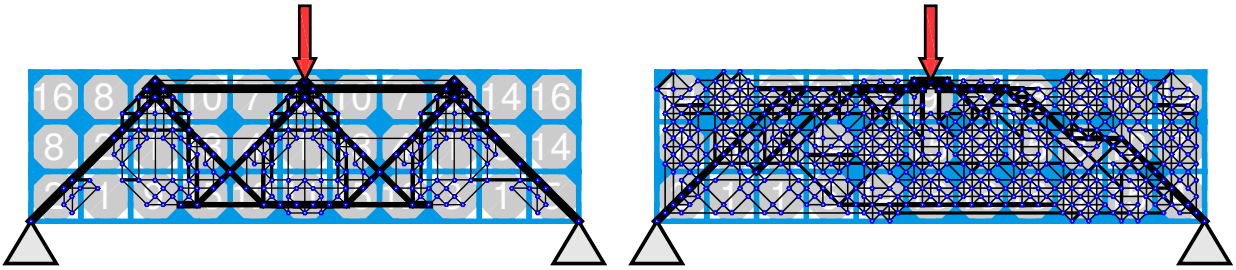
Figure 5.2: Beam tiled with 11×3 tiles of type A after two-phase optimization. The figure (a) is a result of optimization with prescribed symmetry with final volume $V = 1.2481$. The figure (b) is a result of optimization without prescribed symmetry. The final volume is $V = 1.1863$.

5.2 Beam tiled with 11x3 tiles of type B



(a) (b)

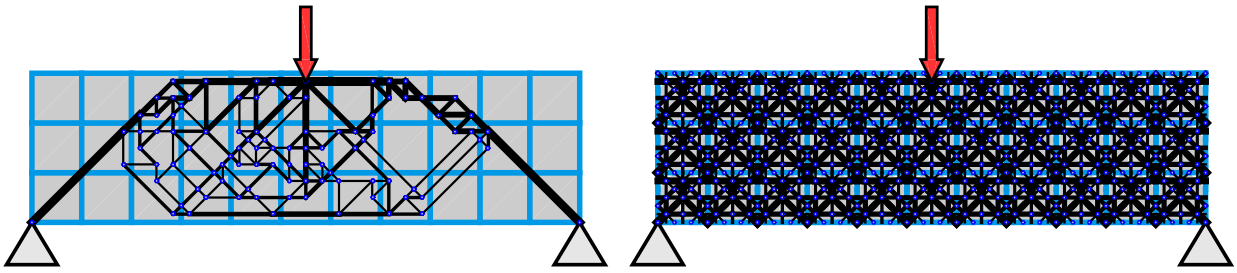
Figure 5.3: The figure (a) shows an ideal non-grouped beam tiled with 11×3 tiles of type B. The ideal volume is $V_{\text{ideal}} = 0.80851$. The latter figure (b) shows the worst case solution of the same beam. The worst volume is $V_{\text{worst}} = 3.4809$.



(a) (b)

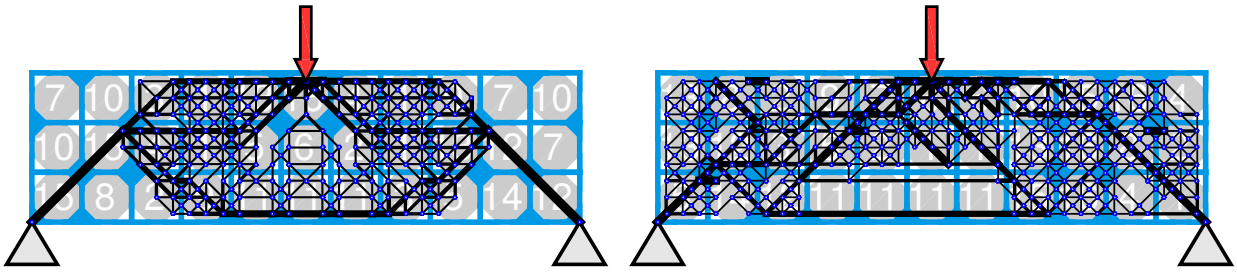
Figure 5.4: Beam tiled with 11×3 tiles of type B after two-phase optimization. The figure (a) is a result of optimization with prescribed symmetry with final volume $V = 0.96671$. The figure (b) is a result of optimization without prescribed symmetry. The final volume is $V = 1.0826$.

5.3 Beam tiled with 11x3 tiles of type C



(a) (b)

Figure 5.5: The figure (a) shows an ideal non-grouped beam tiled with 11×3 tiles of type C. The ideal volume is $V_{\text{ideal}} = 0.80851$. The latter figure (b) shows the worst case solution of the same beam. The worst volume is $V_{\text{worst}} = 3.5419$.



(a) (b)

Figure 5.6: Beam tiled with 11×3 tiles of type C after two-phase optimization. The figure (a) is a result of optimization with prescribed symmetry with final volume $V = 1.0299$. The figure (b) is a result of optimization without prescribed symmetry. The final volume is $V = 1.1687$.

6 Conclusion

Within this work a two-phase optimization of Wang-tile-based trusses has been successfully implemented in the software MATLAB. The results, as shown in the Chapter 5, proved it is possible to lower the weight of a tiled structure, such that it does not significantly vary from the ideal result of optimization of the non-tiled ground structure. The optimal tiled truss can then be more easily prefabricated.

There, however, still remain a lot of work to be done and many issues to be solved. Firstly, although the plastic formulation used as the lower level of the two-phase optimization produces globally-optimal solutions, it is not appropriate for this type of problem, because the resulting topologies are not generally statically determinate and thus not fulfilling the compatibility (stress-strain) relation. The possible future path might be semidefinite programming, which will simultaneously solve the problem of the final truss stability. Also, the objective function will be probably predefined to the minimization of compliance[1].

The upper level of optimization does work as initially observed. However, it can be assumed that the settings of simulated annealing can be further tweaked to speed up the optimization and possibly obtain preferable results. To further improve the performance the possibility of implementation of another heuristic algorithm, such as tabu search or genetic algorithm, will be considered.

Acknowledgement

This work was supported by the Czech Science Foundation, through project No. 14-00420S.

The author acknowledges fruitful discussion and helpful feedback from the supervisor Doc. Ing. Jan Zeman, Ph.D.; and from others who contributed to the initial idea, namely Doc. Ing Matěj Lepš, Ph.D. and Ing. Martin Doškář.

The example Michell truss in Figure 2.1 has been generated using the GRAND package [Zegard and Paulino, 2014].

[1] Recall again that if the optimal truss is statically determinate, the minimum weight optimization using the plastic formulation is a dual of minimum compliance formulation. The optimums are thus equivalent.

Bibliography

M. Bendsoe and O. Sigmund. *Topology Optimization: Theory, Methods, and Applications*. Springer Berlin Heidelberg, 2004. ISBN 9783662050866.

R. Berger. *The undecidability of the domino problem*, volume 66. American Mathematical Soc., 1966.

E. Burke and G. Kendall. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. SpringerLink: Bücher. Springer US, 2013. ISBN 9781461469407.

V. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of optimization theory and applications*, 45(1):41–51, 1985.

M. F. Cohen, J. Shade, S. Hiller, and O. Deussen. *Wang tiles for image and texture generation*, volume 22. ACM, 2003.

S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.

K. Culmann. *Die graphische statik*, volume 2. Meyer & Zeller (A. Reimann), 1875.

W. S. Dorn, R. E. Gomory, and H. J. Greenberg. Automatic design of optimal structures. *Journal de Mecanique*, 3:25–52, 1964.

S. Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of statistical physics*, 34(5-6):975–986, 1984.

A. Lagae and P. Dutré. An alternative for wang tiles: colored edges versus colored corners. *ACM Transactions on Graphics (TOG)*, 25(4):1442–1459, 2006.

A. Michell. LVIII. the limits of economy of material in frame-structures. *Philosophical Magazine Series 6*, 8(47):589–597, 1904. doi: 10.1080/14786440409463229.

M. Ohsaki. *Optimization of finite dimensional structures*. CRC Press/Taylor & Francis, Boca Raton, 2011. ISBN 9781439820049.

A. K. Peter W. Christensen, Anders Klarbring. Peter W. Christensen. *An introduction to structural optimization*. Springer, Dordrecht, online-ausg. edition, 2009. ISBN 9781402086663.

P. Salamon, P. Sibani, and R. Frost. *Facts, Conjectures, and Improvements for Simulated Annealing*. Monographs on Mathematical Modeling and Computation. Society for Industrial and Applied Mathematics, 2002. ISBN 9780898715088.

H. Wang. Proving theorems by pattern recognition – II. *Bell System Technical Journal*, 40, 1961. doi: 10.1002/j.1538-7305.1961.tb03975.x.

T. Zegard and G. H. Paulino. Grand—ground structure based topology optimization for arbitrary 2d domains using matlab. *Structural and Multidisciplinary Optimization*, 50(5): 861–882, 2014.