22nd International Conference

ENGINEERING MECHANICS 2016

Svratka, Czech Republic, 9 – 12 May 2016

GLOBAL TOPOLOGY WEIGHT OPTIMIZATION OF 52-BAR BENCHMARK TRUSS WITH DISCRETE CROSS-SECTIONS

M. Tyburec*, M. Lepš**

Abstract: *Truss topology weight optimization problem with discrete cross-sections can be formulated as a mixed-integer linear program (MILP), which is solvable to global optimality. It is however very difficult to obtain proven globally optimal solution, as there usually exist a very large number of possible combinations. This contribution implements several types of additional cuts and solves the problem using a commercial branch-and-bound software Gurobi, hence making it possible to obtain a guaranteed globally optimal solution. Such solution can then be used as a lower bound for sizing optimization.*

Keywords: branch and bound method, mixed-integer linear programming, topology optimization, 52-bar truss, benchmark.

1. Introduction

Truss topology optimization with discrete variables is a NP-hard problem (Yates et al., 1982). In the case of 52-bar truss (see Fig. 1), which is being used in this contribution, simple enumeration of all possible solutions yields 5,69×10^{21} combinations making it computationally non-manageable.

2. Problem formulation

According to (Rasmussen et al., 2008) the minimum-weight topology optimization problem is formulated as a mixed-integer linear program:

$$\min \rho \sum_{j=1}^{J} l_j \sum_{i=1}^{I} a_i x_{ij} \tag{1}$$

$$\text{s.t.}: Bs = f \tag{2}$$

$$x_{ij} a_i \sigma^{\min} \le s_{ij} \le x_{ij} a_i \sigma^{\max} \quad \forall(i,j) \tag{3}$$

$$(1-x_{ij}) c_{ij}^{\min} \le \frac{E_j a_i}{l_j} b_j^T u - s_{ij} \le (1-x_{ij}) c_{ij}^{\max} \quad \forall(i,j)\,. \tag{4}$$

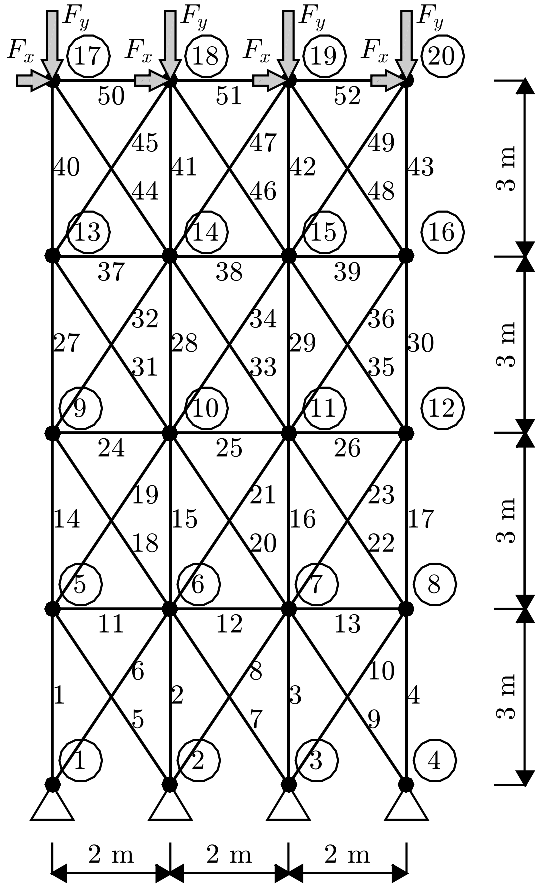
$$u_{\min} \le u \le u_{\max} \tag{5}$$

$$\begin{gathered} \sum_{i=1}^{I} x_{ij} \le 1 \quad \forall j \\ x_{ij} \in \{0,1\} \quad \forall(i,j) \end{gathered} \tag{6}$$

The truss consists of J bars. Each bar j of the truss may have a cross-sectional area selected from the predefined list $\{a_1, a_2, \dots a_I\}$, where I represents the number of available areas. To describe which area is used by the bar j, an additional binary variable x_{ij} is introduced: if the ith area of the bar j is present, then x_{ij} is equal to one; zero otherwise. Each bar has at most one area present (6). If the bar does not have any area, it is removed, thus enabling to influence topology. The bars are divided into several groups with the same area, for implementation details the reader is referred to (Pospíšilová, Lepš, 2013).

* Bc. Marek Tyburec: Faculty of Civil Engineering, CTU in Prague, Thákurova 7/2077; 166 29 Prague; CZ, marek.tyburec@gmail.com

** Doc. Ing. Matěj Lepš, Ph.D.: Faculty of Civil Engineering, CTU in Prague; Thákurova 7/2077; 166 29, Prague; CZ, matej.leps@fsv.cvut.cz



Group	Bars
A$_1$	1, 2, 3, 4
A$_2$	5, 6, 7, 8 ,9, 10
A$_3$	11, 12, 13
A$_4$	14, 15, 16, 17
A$_5$	18, 19, 20, 21, 22, 23
A$_6$	24, 25, 26
A$_7$	27, 28, 29, 30
A$_8$	31, 32, 33, 34, 35, 36
A$_9$	37, 38, 39
A$_{10}$	40, 41, 42, 43
A$_{11}$	44, 45, 46, 47, 48, 49
A$_{12}$	50, 51, 52

External forces

$F_x = 100$ kN; $F_y = 200$ kN

Possible bar cross-sections [mm^2]

71,613; 90,968; 126,451; 161,29; 198,064; 252,258; 285,161; 363,225; 388,386; 494,193; 506,451; 641,289; 645,16; 792,256; 816,773; 940; 1008,385; 1045,159; 1161,288; 1283,868; 1374,191; 1535,481; 1690,319; 1696,771; 1858,061; 1890,319; 1993,544; 2019,351; 2180,641; 2238,705; 2290,318; 2341,191; 2477,414; 2496,769; 2503,221; 2696.769; 2722,575; 2896,768; 2961,284; 3096,768; 3206,445; 3303,219; 3703,218; 4658,055; 5141,925; 5503,215; 5999,998; 6999,986; 7419,34; 8709,66; 8967,724; 9161,272; 9999,98; 10322,56; 10903,204; 12129,008; 12838,684; 14193,52; 14774,164; 15806,42; 17096,74; 18064,48; 19354,8; 21612,86

Material properties

$E = 207$ GPa

$\rho = 7860$ kg/m^3

$\sigma_{\max} = -\sigma_{\min} = 180$ MPa

Fig. 1: Initial 52-bar truss.

The truss is also characterized by prescribed material properties – material density ρ, Young's modulus E and by minimal and maximal allowed stresses $\sigma^{\min}$ and $\sigma^{\max}$. Based on linear elasticity it is possible to limit internal normal forces of individual bars s_{ij}, see Equation (3).

It is usually needed to limit displacements u of the truss by a minimal $u^{\min}$ and a maximal displacements $u^{\max}$ (5), respectively. To preserve linearity of the formulation, the internal forces s are bounded by $c^{\min}$ and $c^{\max}$ (Rasmussen et al., 2008):

$$\begin{aligned} c_{ij}^{\min} &= \frac{E_j a_i}{l_j} \min_{u^{\min} \le u \le u^{\max}} \left\{ b_j^T u \right\} \quad \forall (i,j) \\ c_{ij}^{\max} &= \frac{E_j a_i}{l_j} \max_{u^{\min} \le u \le u^{\max}} \left\{ b_j^T u \right\} \quad \forall (i,j) \end{aligned} , \qquad (7)$$

where b_j represents the jth row of the static matrix B. These bounds should be valid only when the area i is present (4). To preserve compatibility for trusses without specified displacement limits, the values of $c^{\min}$ and $c^{\max}$ are replaced by extreme normal forces from Equation (3). The force equilibrium in nodes is described by the static matrix B and Equation (2), where f represents the external nodal forces vector.

2.2 Branch-and-bound algorithm

Because the formulation of the problem (1)-(6) is linearly dependent on design variables x, s and u, respectively, the problem may be solved by branch-and-bound algorithm. The algorithm starts with a relaxed problem formulation, without considering binary (integer) variables, thus making it possible to solve an ordinary linear program (LP). If all the binary variables obtained by the solution of the relaxed problem have only binary values, global optimum is obtained. In the opposite case the initial relaxed problem is branched into multiple sub-problems (leave nodes) fixing specific binary variable on values 0 and 1. All the relaxed sub-problems are ordered by their objective value and solved from the smallest one, creating a lower bound of the problem.

Branch-and-bound algorithm uses heuristics for finding a feasible integer solution fulfilling all the prescribed constraints, hence solving the full problem (1)-(6) and creating the upper bound. Therefore, if any relaxed sub-problem takes higher objective value than the upper bound, it is discarded. The algorithm terminates only if the upper bound is equal to the lower bound and no spare leave nodes are left.

The benefits of branch-and-bound algorithm are obvious: limiting the number of combinations needed to solve the problem to global optimality and thus significantly speeding the computations up.

3. Tightening of the problem

To tighten the general formulation of the problem (1)-(6) even further, additional cuts are introduced. Their purpose is to limit the design space of the problem while preserving all feasible binary solutions and consequently keeping the same global optimum. The cuts have significant effect on both the number of branched nodes and the speed of the branch-and-bound algorithm itself.

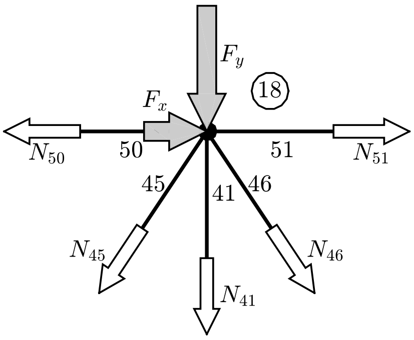


Fig. 2: Loaded node 18.

3.1 Loaded-nodes cuts

Each externally loaded node has to fulfill the force equilibrium (2). The vertical force F_y located at the node k has to be carried only by non-horizontal bars connected to the node k. It is evident that to carry the force F_y in Figure 2 at least one of the bars 41, 45 or 46 has to be present (Rasmussen et al., 2008) and the total area of these bars orthogonally displayed to vertical axis has to be greater than the minimal area needed to carry the external vertical force. The same procedure is also applied to the horizontal direction.

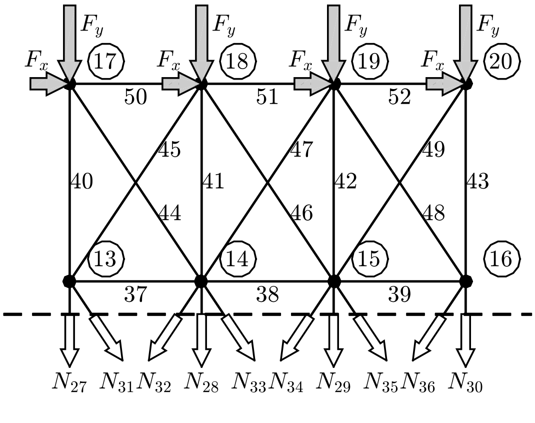


Fig. 3: Method of section – cut through bars 27-36.

3.2 Method-of-sections cuts

Similar idea is used for the method-of-sections cuts (Lepš et al., 2014). All the cuts are generated as follows: Firstly, the truss is divided into two sections. It is then checked if there exist a situation that all the supports in one direction (e.g. x) are on one section and there exist at least one external force, in the same direction as the supports, located on the other section. Consequently, there have to exist at least one bar in the direction of the external force connecting the two sections and the total area of the orthogonally displayed bars have to be greater than the minimum area needed to carry the external force. Considering Figure 3 and the fact that the area of bars 31-36 is equal (see Fig. 1), all the bars 31-36 have to be present and their minimal area needs to carry external forces F_x to the supports located on the other section.

3.3 Inner-nodes cuts

Inner-nodes cuts apply for all nodes without external forces or supports. They arise from an idea there cannot exist single horizontal (or vertical) bar in a node. If there would not be any horizontal force carried through the node then no bar was needed, otherwise there have to be at least two bars to carry the normal force (Rasmussen et al., 2008). However, inner-nodes cuts are useful only for topology optimization.

4. Results

Topology optimization of the 52-bar truss benchmark was launched on a computer with 16 cores, a limit of 128 GB RAM and GUROBI MILP solver (Gurobi Optimization, Inc., 2015). The globally optimal truss structure with a weight of 1902,606 kg has been obtained. The optimization progress is shown in the Fig. 4.

Note that the described implementation is not easily applicable for general use; still, it did significantly speed up the optimization. The global optimum of the benchmark truss is useful as a comparison for faster but not global heuristic algorithms or as a lower bound for sizing optimization.

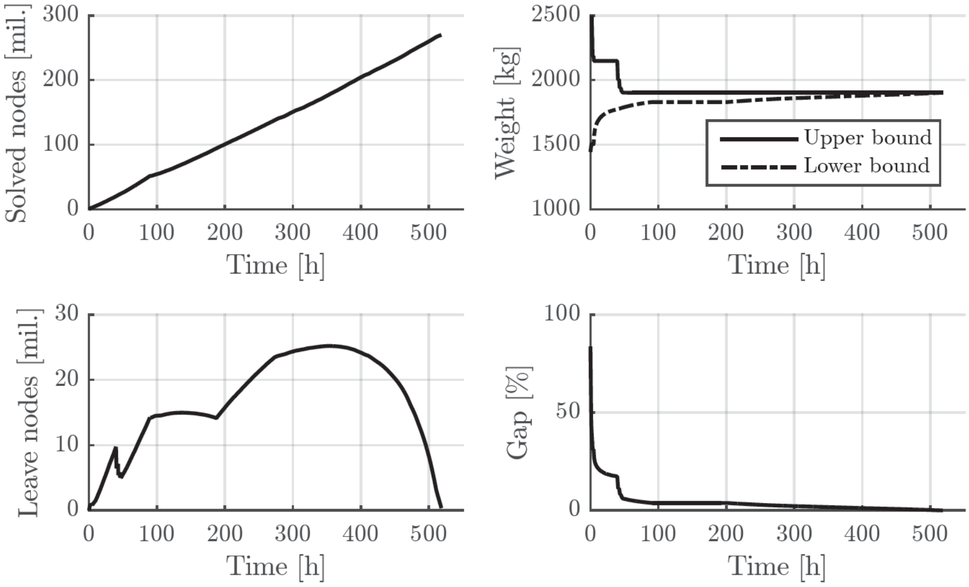


Fig. 4: 52-bar benchmark truss topology optimization.

Acknowledgement

The implementation has been built upon the MATLAB codes provided by Ing. Adéla Pospíšilová. The authors also acknowledge the financial support from the Grant Agency of the CTU in Prague, the grant SGS16/037/OHK1/1T/11.

References

Gurobi Optimization, Inc. (2015) Gurobi Optimizer Reference Manual, http://www.gurobi.com.

Lepš, M., Nosek, J. & Pohlídalová, E. (2014) 52-bar truss optimization benchmark: computational enhancements, in: Proceedings on the 5th Conference Nano & Macro mechanics, Czech Technical University, pp. 111-114.

Lepš, M. & Pospíšilová, A. (2013) Parallel Branch and Bound Method for Size Optimization Benchmarks. In: Proceedings of the Third International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering, Civil-Comp Press, Stirlingshire, UK, Paper 20.

Rasmussen, M. H. & Stolpe, M. (2008). Global optimization of discrete truss topology design problems using a parallel cut-and-branch method. Computers & Structures, 86(13), pp. 1527-1538.

Boffey, T. B., Templeman, A. B. & Yates, D. F. (1982). The complexity of procedures for determining minimum weight trusses with discrete member sizes. International Journal of Solids and Structures, 18(6), pp. 487-495.