Solving topology optimization with {0,1} design variables and mathematical programming: the TOBS method

R. Picelli
University of São Paulo

Raghavendra Sivapuram
University of California, San Diego

As recent as 2020 solving TO problems to have 0/1 solutions is still challenging.

Solving topology optimization with {0,1} design variables and mathematical programming: the TOBS method

Renato Picelli[1]*, **Raghavendra Sivapuram**[2]

[1]*Department of Mining and Petroleum Engineering, University of São Paulo, 11013-560, Santos, Brazil*
* *Corresponding author:* rpicelli@usp.br

[2]*Structural Engineering Department, University of California, San Diego, CA 92093, San Diego, USA*

Abstract

Gradient-based topology optimization problems using binary {0,1} design variables have been considered extremely challenging or even impossible to be solved, leading to the creation of relaxed problems such as the ones used in the popular density-based methods. This work investigates a combination of numerical ingredients that allows the original {0,1} problem to be solved sequentially and with formal mathematical programming. This method was named Topology Optimization of Binary Structures (TOBS) and it was developed in Sivapuram and Picelli [1] and Sivapuram et al. [2]. In the TOBS method, the objective and constraint functions are sequentially linearized using Taylor's first order approximation. Sensitivity numerical filtering is applied to avoid checkerboard and mesh-dependency issues. Integer Linear Programming (ILP) is used to compute the optimized solutions for the linearized problems at each iteration, allowing the method to accommodate the set of constraints explicitly. The constraint targets are relaxed to allow only small changes in topology during an update and to ensure the existence of feasible solutions for the ILP. This work discusses the possible advantages and disadvantages of the TOBS method, as well as its details and comparisons with other discrete methods.

Keywords: *Structural Optimization, Topology Optimization, Discrete Methods, Binary Variables, Integer Programming*

1. Introduction

Topology optimization for elastic structures is essentially a material distribution problem with binary {0,1} design variables, where 0 means the absence of material (void) and 1 represents the location of solid material. The binary problem has been considered for decades very hard or even impossible to be solved using formal mathematical programming. Hence, the idea of relaxing the binary constraint by allowing intermediate densities became popular with the Solid Isotropic Material with Penalization (SIMP) formulation [3]. The SIMP method usually starts with intermediate (gray scale) densities and during optimization it steers the solution to a nearly solid/void design. The advent of projection schemes helped the SIMP solution to converge to a solid/void design [4].

The first attempt to carry out topology optimization of continuum structures using binary variables was made by Xie and Steven [5]. The idea is that inefficient material can be removed from the structure using sensitivity information. This method was called Evolutionary Structural Optimization (ESO). The method became popular after convergent and mesh independent solutions were presented by Huang and Xie [6], who proposed a Bi-directional ESO (BESO). The BESO method is based on a heuristic design update scheme that allows material to be also added to the structure. In the BESO algorithm, the structure is usually (but not necessarily) first considered as a full solid design and a target volume is used to quantify the amount of removed/add material until convergence, only allowing {0,1} variables. One advantage of using binary variables is in the solution of problems where the structure interacts with other physics, e.g., fluid-structure interaction design, where the switch between solids and fluids/voids is straightforward and the equilibrium conditions at the interfaces are explicit [7]. On the other hand, the main disadvantage of the BESO method is that its update scheme is based on a target volume, therefore always requiring the presence of a volume constraint in the optimization formulation. This precludes problems such as mass minimization or multiple constraints to be solved in a schematic way.

Beckers [8] introduced formal mathematical programming to solve a structural topology optimization problem with binary {0,1} variables. Beckers' method was based on sequential approximate programming and a Lagrangian dual method optimizer. In this way, the solutions obtained were convergent and mesh independent but limited to very few constraints. Svanberg and Werme [9] found that if only one element is binarily changed (from material to void or from void to material), the new global stiffness matrix is just a low-rank modification of the old one, therefore validating discrete sensitivities. Relying on that, Svanberg and Werme [10] introduced integer linear programming to solve a sequentially approximate problem with binary variables. As a drawback, their choice of hierarchically refining the finite element mesh can lead to mesh dependent solutions. To the best of our knowledge, we presented the first convergent and mesh independent solutions for topology optimization with binary design variables and integer programming in Sivapuram and Picelli [1]. The method so called Topology Optimization of Binary Structures (TOBS) combines four numerical ingredients: sequential problem linearization, constraints' relaxation (move limits), sensitivity filtering and an integer programming solver. This represents a more general scheme for topology optimization with {0,1} variables than [8] and [10], as also showed by Sivapuram et al. [2] for microstructural optimization problems with multiple nonvolume constraints. Munk [11] used the same four numerical ingredients as the TOBS method in [1] and developed some heuristics for the move limits. However, in our understanding, the author in [11] is misleading when calling his method BESO because the intrinsic heuristic BESO update scheme is not present anymore. In fact, the TOBS method applies well known numerical ingredients previously used by other methods in order to find an algorithm that leads to convergent binary solutions in a general topology optimization framework. Very recently, Liang and Cheng [12] proposed a similar method for binary topology optimization including a few other techniques. Liang and Cheng applied the Canonical Dual Theory [13] to solve the integer problem while the TOBS method uses a branch and bound solver [14] which worked well for a wide range of problems investigated.

This paper aims to provide details of the TOBS method as well as brief comparisons with the BESO method [8] and the methods by Svanberg and Werme [10] and Liang and Cheng [12]. The remainder of the paper is as follows. Section 2 describes the TOBS method and Sec. 3 summarizes it in steps. Section 4 presents numerical results and Sec. 5 concludes the paper.

2. The TOBS method

2.1 Problem linearization

A generic topology optimization problem with objective function $f(\boldsymbol{x})$, constraints $\boldsymbol{g_i}(\boldsymbol{x}) \leq \bar{\boldsymbol{g}}_{\boldsymbol{i}}$, where $\boldsymbol{x}$ are the design variables, is herein linearized via Taylor's first order approximation. The approximate problem is given by:

$$\begin{aligned} &\text{minimize} && \frac{\partial f}{\partial x}\Delta\boldsymbol{x}^k \\ &\text{subject to} && \frac{\partial g_i}{\partial x}\Delta\boldsymbol{x}^k \leq \Delta g_i^k && i \in [1, N_g] \\ & && \Delta x_j^k \in \{-x_j^k, 1 - x_j^k\} && j \in [1, n_d] \end{aligned} \tag{1}$$

where N_g is the number of constraints and N_d is the number of design variables, Δg_i^k is the change in the ith constraint function at the kth iteration and $\Delta\boldsymbol{x}^k$ represents the change in design variables. At the end of each iteration, the design variables are updated as $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \Delta\boldsymbol{x}^k$.

In order to maintain the validity of the linear approximations and the truncation error low, an extra constraint is added:

$$\|\Delta\boldsymbol{x}^k\|_1 \leq \beta N_d \tag{2}$$

This equation implies that the number of flips between solid to void and vice versa is constrained to be a fraction β of the total number of design variables. By limiting the change of the structural topology at every iteration, the number of solid material layers to be removed is also kept low.

2.2 Move limits

Gradient-based optimization requires an initial solution to start with, which might lie in the infeasible space of the optimization problem. To guarantee the linearized constraints are satisfied at every iteration, the constraints are relaxed, i.e., the values Δg_i^k are chosen such that the linearized problem has a feasible solution. The relaxation for a constraint g at the iteration k is given by:

$$\Delta g^k = \begin{cases} -\varepsilon_1 g^k & : \overline{g} < (1-\varepsilon_1)g^k \\ \overline{g} - g^k & : \overline{g} \in [(1-\varepsilon_1)g^k, (1+\varepsilon_2)g^k] \\ \varepsilon_2 g^k & : \overline{g} > (1+\varepsilon_2)g^k \end{cases} \tag{3}$$

where ε_1 and ε_2 are small numbers chosen to keep Δg^k small enough so that the subproblem in Eq. (1) has a feasible solution. Throughout this work, $\varepsilon_1 = \varepsilon_2 = \varepsilon$ is used.

2.3 Sensitivity analysis and filtering

The TOBS is a gradient-based topology optimization method. The first gradients (sensitivities) must be computed in order to evaluate the linearized functions from Eq. (1) and solve the subproblem at iteration *k*. The sensitivities can be evaluated using any sensitivity analysis method [15], e.g., analytical, semi-analytical or finite differences if only the corresponding binary values are used. In this work, we have used the semi-analytical method to evaluate the sensitivities of the mean compliance *C* of an elastic structure, which final function can be written as:

$$\frac{\partial C}{\partial x_j} \approx (\Delta C)_j = -\frac{1}{2}\boldsymbol{u}_j^T \boldsymbol{K}_j^e \boldsymbol{u}_j \tag{4}$$

where $\mathbf{u}j$ and $\mathbf{K}j$ are, respectively, the vector of displacements and the stiffness matrix of the j^{th} element. Using the same method, the sensitivity of the volume with respect to each design variable is the volume of the element.

Numerical filtering on the sensitivity field has been massively used in structural topology optimization to avoid numerical issues, e.g., mesh dependency and checkerboard problems, besides guaranteeing a sort of length scale. Herein, we filter sensitivities by first creating a nodal sensitivity field. In a regular grid, the nodal sensitivity (with respect to a virtual nodal design variable y_n) is the average of the sensitivities of each element the node is connected to. The filtered sensitivity field with respect to the design variables is reconstructed from the nodal sensitivity field by using weighted-distance averaging in a neighborhood of each finite element. This filtered sensitivity is given by:

$$\frac{\partial f(x)}{\partial x_j} = \frac{\sum_{i \in N} w_{ji} \partial C / \partial y_n}{\sum_{i \in N} w_{ji}} \tag{6}$$

where *N* is the set of nodes in a neighborhood of the j^{th} element. We use a circular neighborhood of radius *r* with the center being the centroid of the j^{th} finite element.

We have observed small oscillations in the structural members due to the high number of local minima for some thermomechanical problems. Therefore, we use the stabilization scheme proposed by Huang and Xie [6], which is the averaging of the sensitivity field with its previous iteration. It can be pointed out that this stabilization scheme is not essentially required but its effects improve convergence considerably, as also showed in [6]. We have also observed that normalizing sensitivities helps the optimizer to find an integer solution faster since all sensitivity numbers will have a similar range, which also avoids ill-conditioning. This only implies that the subproblem in the k^{th} iteration is scaled.

2.4 Integer programming

The linear subproblems are solved using Integer Linear Programming (ILP). An ILP problem is the same as Linear Programming (LP) problem with additional constraints that the feasible space of design variables contains only integers. ILP problems can be solved using branch-and-bound methods by initially solving the LP problem, i.e., without the integer constraints on design variables, and the obtained solution is used to create branches of LP problems with additional inequality constraints to achieve integer solutions [14]. In this work, the package CPLEX is used for solving the ILP problems. The **cplexmilp** function of the CPLEX MATLAB package uses the branch-and-bound method for solving mixed integer linear problems.

3. Code and algorithm

A demonstration code with a possible implementation of the TOBS method is available upon request by e-mail at rpicelli@usp.br. This code is based on the following steps:

1. Discretize the design domain, apply loads and boundary conditions.
2. Choose the TOBS parameters, ε to relax each constraint function, β to limit the total amount of solid-to- void and void-to-solid flips, filter radius *r* and convergence criteria.
3. Perform Finite Element Analysis (FEA) of the structure.
4. Compute sensitivities and the objective and constraint functions' values.
5. Filter the sensitivities of each of the function.
6. Average the sensitivity fields with the corresponding fields from previous iteration.
7. Normalize the sensitivities.

8. Build the optimization problem (Eq. (1)) with relaxed constraints (Eq. (3)) and add the extra flip limit constraint (Eq. (2)).
9. Solve the problem with ILP.
10. Update design variables $\boldsymbol{x}^{k+1} = \boldsymbol{x}^{k} + \Delta\boldsymbol{x}^{k}$.
11. If converged, stop. Otherwise, return to step 3.

4. Discussions and numerical results

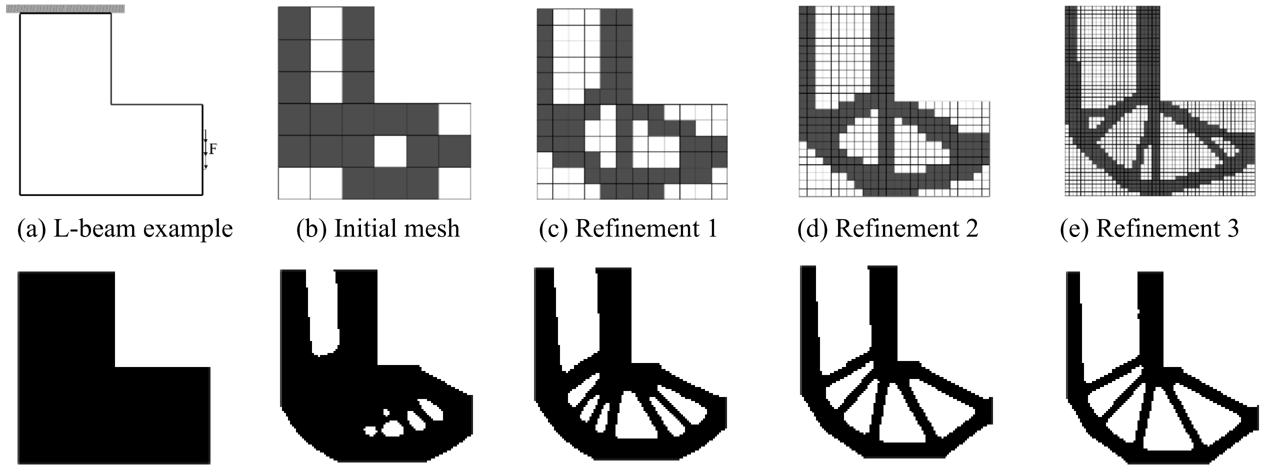
Although Beckers [8] have brilliantly obtained mesh independent and convergent solutions in 1999, the role of using binary design variables in topology optimization of continuum structures have been mostly carried out by the BESO method. In its early years, BESO has been criticized by its lack of formal mathematical programming. Nevertheless, it became a popular algorithm due to its efficiency and the ability to produce binary solutions that are comparable with other methods. However, the problems that BESO can solve are within the range that include only volumetric constraints because of BESO's target volume update. Therefore, the TOBS method can replace the BESO method for more complex problems out of the BESO range, like mass minimization and multi-constrained problems. The convergence of the nonvolume constraints is better in TOBS because they are dealt explicitly, i.e., without using any Lagrange multipliers like BESO. The advantages of using binary variables include mostly problems where material interpolation (like used in density-based methods) is challenging or leads to numerical issues. Example of these problems are thermoelastic design, fluid-structure interaction, acoustic-structure interaction, i.e., design-dependent physics. For single physics problems, density-based methods outperform discrete methods because they present a higher design space and have more powerful optimizers available. Furthermore, density-based methods have proven to solve a wide range of challenging problems. Therefore, future recommendations for discrete methods include new applications and the development of new optimizers that are faster and can solve large scale problems.

We advocate that the four numerical ingredients from Sec. 2 are the recipe to carry out topology optimization with binary variables. Naturally, there are options between the available tools for each of the ingredients, e.g., different integer programming solvers like **cplexmilp** from CPLEX library or **intlinprog** from MATLAB. In order to illustrate this recipe, we start comparing the TOBS with the method by Beckers [8]. For the benchmark MBB beam example with a regular unitary 150x50 mesh, Young's modulus 1.0, Poisson's ratio 0.3 and unitary applied force, Fig. 1 shows the solutions for minimum structural compliance with a volume fraction of 50%. Herein, $\varepsilon = 0.01$ and $\beta = 0.05$. Beckers' method applied all the four numerical ingredients and it was the first to obtain mesh independent solutions. The main difference between the TOBS and Beckers' [8] method lies in the integer program solver, in which Beckers applied a Lagrangian dual solver, limiting the application to a few constraints. The TOBS method uses integer linear programming, able to accommodate a higher number of constraints but slower in obtaining the solution. Another difference is the introduction of a perimeter constraint in [8], not present in the TOBS solution from Fig. 1(b), which possibly explains the different number of holes in the final topology. The final compliance was 189.72 for the TOBS method and 193.75 for Beckers' optimized solution.

(a) (b) (c)

Figure 1: Solutions obtained for (a) MBB beam with the (b) TOBS method and (c) Beckers' [8] method.

The work by Svanberg and Werme [10] seems to indicate the authors did not use sensitivity filtering. Instead, they proposed a mesh refinement scheme in which the optimization problem is solved for each mesh level. Although they also used constraint limits (relaxation), mesh refinement guaranteed the validity of the discrete sensitivities, as very few element layers are removed every time. The TOBS method guarantees these move limits with constraint relaxation (parameter ε) and the extra constraint on the binary flips (parameter β). Furthermore, sensitivity filtering plays the crucial role of extrapolating the sensitivity field so as void elements nearby high stressed regions can return to solid. Figure 2 shows the solutions for a L-beam structure for minimum compliance and volume constraints with the method by Svanberg and Werme [10] and the TOBS, using the same initial parameters as the previous example.

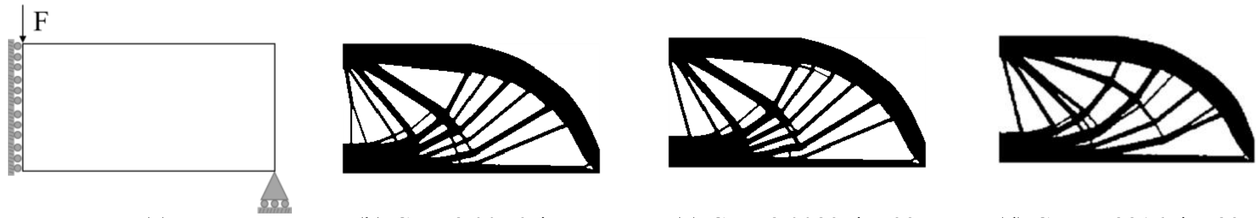


(a) L-beam example (b) Initial mesh (c) Refinement 1 (d) Refinement 2 (e) Refinement 3

(f) it.: 0 (g) it.: 25 (h) it.: 50 (i) it.: 75 (j) it.: 98

Figure 2: Solutions obtained for (a) L-beam example with the (b-e) method by Svanberg and Werme [10] with mesh refinement and (f-j) the TOBS method.

Liang and Cheng [12] recently proposed the use of Canonical Dual Theory (CDT) with the argument that the canonical relaxation solver can accommodate more design variables than integer programming. Nevertheless, it seems that the number of constraints solved by CDT is limited, similarly to Beckers [8]. Figure 3 presents the solutions of a regular 300x150 mesh structural design for minimum compliance with 50% of constrained volume fraction using the TOBS method (with same parameters of the previous examples), our own implementation of the BESO method and the solution presented by Liang and Cheng [12]. The specific particularities of these methods include that Liang and Cheng [12] use move limits based on the volume or densities while the TOBS method relaxes (limits) the constraint changes at every iteration. The BESO method does not use either sequential approximate problems or formal mathematical programming and its move limit is actually based on a heuristic update scheme. Anyhow, the three solutions showed to be similar local minima. Liang and Cheng [13] also presented a similar solution but using quadratic programming, which decreased the number of iterations by half.



(a) (b) C = 78.0076; it: 77 (c) C = 78.0089; it.: 93 (d) C = 77.8816; it.: 83

Figure 3: Solutions obtained for (a) MBB beam with (b) the TOBS method, (c) our own implementation of the BESO method [6] and (d) sequential linear integer programming via canonical relaxation by Liang and Cheng [12]. Total of 45000 design variables.

With the previous comparisons, one can observe that Beckers [13], Liang and Cheng [17], BESO [8] and TOBS methods present similar solutions when solving a minimum compliance with volume constraints problem. Beckers [13] and Liang and Cheng [17] did not show volume minimization examples, although their methods are theoretically able to solve them. We would like to point out that one good use of the TOBS method is for solving minimum volume problems with multiple constraints as showed by Sivapuram et al. [2] and by Munk [16].

5. Conclusions

This paper formalizes the four numerical ingredients of doing topology optimization with binary {0,1} design variables, specifically in the context of the proposed TOBS method. Numerical results show that the method presents comparable solutions with literature (except for Svanberg and Werme [10]) when compared to a simple compliance minimization with volume constraint problem.

Acknowledgments

Renato Picelli would like to thank the support of São Paulo Research Foundation (FAPESP), grants 2018/05797-8 and 2019/01685-3.

References

1. Sivapuram R and Picelli R, Topology optimization of binary structures using integer linear programming. Finite Elemen. Anal. Des., 2018, 139:49–61
2. Sivapuram R, Picelli R and Xie YM, Topology optimization of binary microstructures involving various non-volume constraints. Comput. Mater. Sci., 2018, 154:405–425
3. Rozvany GIN, Zhou M and Birker T, Generalized shape optimization without homogenization. Struct. Optim., 1992, 4(3):250–252
4. Guest JK, Prévost JH and Belytschko T, Achieving minimum length scale in topology optimization using nodal design variables and projection functions. Int. J. Numer. Methods. Eng., 2004, 61(2):238–254
5. Xie YM and Steven GP, A simple evolutionary procedure for structural optimization. Comput. Struct., 1993, 49(5):885–896
6. Huang X and Xie YM, Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method. Finite. Elem. Anal. Des, 2007, 43(14):1039–1049
7. Vicente WM, Picelli R, Pavanello R and Xie YM, Topology optimization of frequency responses of fluid–structure interaction systems, Finite Elem. Anal. Des., 2015, 98:1–13
8. Beckers M, Topology optimization using a dual method with discrete variables, Struct. Optim., 1999, 17:14–24
9. Svanberg K and Werme M, A hierarchical neighbourhood search method for topology optimization, Struct. Multidiscip. Optim., 2005, 29:325–340
10. Svanberg K and Werme M, Topology Optimization by Sequential Integer Linear Programming In: Bendsøe MP, Olhoff N, Sigmund O (eds) IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials. Solid Mechanics and Its Applications, vol 137, 2007, Springer, Dordrecht
11. Munk DJ, A bidirectional evolutionary structural optimization algorithm for mass minimization with multiple structural constraints. Int. J. Numer. Methods Eng., 2019, 118:93–120
12. Liang Y and Cheng G, Topology optimization via sequential integer programming and Canonical relaxation algorithm, Comp. Meth. Appl. Mech. Eng., 2019, 348:64–96
13. Gao DY, Ruan N, Solutions to quadratic minimization problems with box and integer constraints, J. Global Optim., 2010, 47:463–484
14. Williams HP. Logic and integer programming. Boston: Springer, 2009
15. Haftka RT and Gürdal Z. Elements of structural optimization. 3rd ed. Dordrecht: Springer, 1992