Proceedings of the ASME 2018 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2018
August 26-29, 2018, Quebec City, Quebec, Canada

DETC2018-86116

FINDING BETTER LOCAL OPTIMA IN TOPOLOGY OPTIMIZATION VIA TUNNELING

Shanglong Zhang
Structural Optimization Laboratory
Department of Mechanical Engineering
University of Connecticut
Storrs, Connecticut 06269
shanglong.zhang@uconn.edu

Julián A. Norato*
Structural Optimization Laboratory
Department of Mechanical Engineering
University of Connecticut
Storrs, Connecticut, 06269
norato@engr.uconn.edu

ABSTRACT

Topology optimization formulates the material distribution problem as a non-convex optimization problem. Therefore, there are many local minima exist in topology optimization problem and gradient-based optimizers are employed to find one of them. However, the optimization is prone to converging to a suboptimal design when structural responses are very non-linear or multiple design constraints are presented. This issue is more severe for the topology optimization of geometric primitives. In this paper, we investigate the use of tunneling in topology optimization to alleviate this issue. The tunneling method used in this work is a gradient-based deterministic method finds a better minimum than the previous one in a sequential manner. The couple of tunneling method and topology optimization enables the discovery of better designs. Numerical examples are provided to demonstrate the effectiveness of proposed method.

INTRODUCTION

Topology optimization is an efficient tool to find the material distribution within a design space for the best structural performance. Since the pioneering development in [1], numerous methods have been proposed to solve topology optimization problems. We refer readers to [2, 3] for detailed reviews.

The density-based topology optimization method proposed in [4, 5] is widespread in both industrial and academic applications. In this approach, the design space is discretized in a voxel-like manner using the elements of a finite element mesh. A continuous density variable is allocated to each element, and the mesh remains fixed throughout the optimization. The elemental densities are the design variables in the optimization, and they are used to weight the material stiffness within each element via

$$\mathbf{C}(\rho_i) = \rho_i^p \mathbf{C}_o \tag{1}$$

where ρ_i is the density for element i and p is a penalization power. This power law approach is the so-called solid isotropic material with penalization (SIMP), and its goal is to penalize intermediate values of the elemental densities so as to obtain a mostly 0-1 (solid-void) design. This is necessary because, in most cases, it is not possible to interpret and physically realize intermediate densities.

If $p = 1$, the problem of minimum compliance subject to a volume constraint is convex and has a unique solution [6]. However, extensive regions with intermediate densities will appear in the final design and, as aforementioned, the resulting design will not be in general manufacturable. If, on the other hand, $p > 1$, intermediate density values will render lower stiffness per amount of material and therefore be structurally inefficient, and thus the optimizer will favor a 0-1 design. However, the optimization problem becomes non-convex. In the absence of some control mechanism, it also lacks a global minimum, since the optimal design would have an infinite number of holes. In the following discussion, however, we assume that a length control mechanism

*Corresponding author.

such as, for example, filtering [7, 8], slope control [9] or perimeter control [10] is employed and thus a solution to the optimization problem exists and is mesh-independent.

Gradient-based optimizers are employed to find a local optimum for this non-convex problem. Different optimizers may render different local optima as shown in [11, 12]. Moreover, a small change in the initial parameters of the problem may lead to a different optimum (cf. [13]). In [14], a continuation method is proposed to improve the convergence of the optimizer towards the global optimum for compliance problems. The optimization starts from an unpenalized ($p = 1$) convex problem, then increases p by a small amount at every optimization iteration until reaching the desired SIMP penalization value (e.g., $p = 3$). Other continuation methods can be found in [13]. Although continuation methods may also work for non-compliance objective problems, the effectiveness cannot be guaranteed [2]. Therefore, a robust algorithm to locate the global optimum of topology optimization problems is desired.

Another important family of topology optimization techniques, which we do not study in this work, is that of level set methods (cf., [15]). In these techniques, the boundary of the design is represented by a level set of a function. The analysis can be done by mapping this level set to a density as in density-based methods, or by immersed-boundary techniques that are able to represent a sharp boundary. Level set techniques provide a sharper definition of the boundaries than density-based methods throughout the entire optimization. The way in which the design is updated is by advancing the level set of the initial design, i.e., by modifying the boundaries of the initial design [15]. In some techniques, holes are introduced at certain locations as dictated by an indicator function, such as the topological derivative. Although we do not cover these methods in this work, the fact that they produce design changes by modifying the boundaries of the initial design presumably also makes them prone to converging to suboptimal local minima. As such, we believe the techniques we advance in this paper may also be extended to this family of methods.

Convergence to undesirable or suboptimal local minima has also been observed in geometry projection methods for topology optimization with discrete geometric primitives [16–20]. In these methods, structural shapes with a a high-level geometric description such as bars and plates are smoothly mapped onto a continuous density field using the geometry projection technique formulated in [21]. These techniques render designs that are distinctly made of the prescribed geometric primitives, and they have been successfully applied to compliance-based [16–18, 20] and stress-based topology optimization problems [19] with various design constraints. However, as reported in these works, different initial designs (made of arbitrary layouts of primitives) or different optimization parameters may lead to different local optima and possibly a highly suboptimal design. This behavior is more severe in stress-based optimization, as demonstrated in [19]. Moreover, the optimization can get more easily stuck in a poor local minimum in the presence of multiple design constraints. A related family of techniques belong to the moving morphable components (MMC) method [22–25], whereby primitives such as bars and plates are represented via implicit functions (which the authors call the topological description functions), and then mapped onto the analysis via a smoothed Heaviside approximation.

In this work, we investigate the use of a gradient-based global optimization technique, the tunneling method, with topology optimization problems in order to obtain better local minima. Tunneling is a deterministic method to locate the global optimum of an optimization problem [26–29]. This method was first demonstrated for unconstrained optimization [26] and later extended to constrained optimization problems [27]. The tunneling process finds a better local minimum than the previous one in a sequential manner. Because an expensive finite element analysis is required for every function evaluation, it is infeasible to apply non-gradient methods based on random processes for topology optimization [30]. This includes random tunneling, which follows a similar idea to the deterministic tunneling method, but can only be applied to problems with inexpensive function evaluations. Random tunneling has been used for topology optimization of trusses with a few members [31]. On the contrary, the deterministic tunneling method utilizes gradient information at every stage, and thus requires less function evaluations. This property makes it a good candidate for finding better local minima using the aforementioned topology optimization techniques. The tunneling method will be briefly introduced and summarized in the next section.

THE TUNNELING METHOD

The tunneling method is a deterministic technique to find the global optimum (if it exists) by cycles. In each cycle, there are two phases. The first phase is an optimization phase to find a local optimum. The second phase is a tunneling phase to find a new starting point for the next optimization phase. Fig. 1 illustrates these cycles for an unconstrained optimization problem. In this example, the optimization phase renders the first local minimum $\mathbf{x}_1^\star$ from the starting point $\mathbf{x}_1^o$. The subsequent tunneling phase finds a starting point $\mathbf{x}_2^o \neq \mathbf{x}_1^\star$ for the next cycle with a function value $f(\mathbf{x}_2^o) = f(\mathbf{x}_1^\star)$ by solving the tunneling function

$$T(\mathbf{x}, f(\mathbf{x}_1^\star)) = f(\mathbf{x}) - f(\mathbf{x}_1^\star) = 0 \tag{2}$$

This phase of the cycle 'tunnels' under irrelevant minima until it finds the starting point $\mathbf{x}_2^o$ for the next optimization phase, which leads to a new local minimum $\mathbf{x}_2^\star$ with a function value

$$f(\mathbf{x}_2^\star) \leq f(\mathbf{x}_2^o) \tag{3}$$

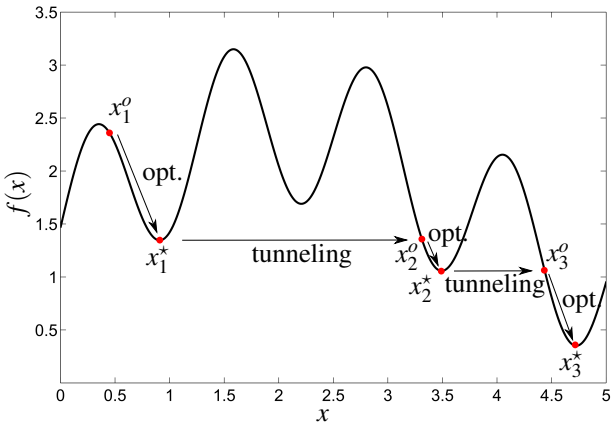

FIGURE 1: Schematic plot of the tunneling method applied to an unconstrained optimization problem.

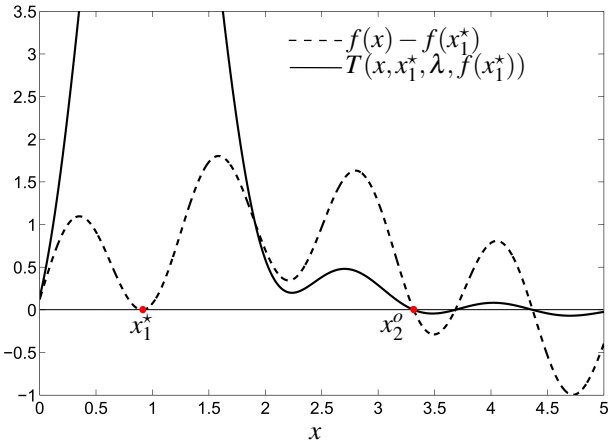

FIGURE 2: The effect of adding a pole at $x_1^\star$ to the tunneling function.

and consequently

$$f(\mathbf{x}_2^\star) \leq f(\mathbf{x}_1^\star) \tag{4}$$

This property can be used to demonstrate the global descent property of the tunneling method [26, 27]:

$$f(\mathbf{x}_i^\star) \leq f(\mathbf{x}_{i-1}^\star),\ i = 1, 2, \ldots, k \tag{5}$$

If the tunneling phase fails to find a solution of Eqn. (2) for $\mathbf{x}_i^\star$ after a certain number of attempts, one can assume that $\mathbf{x}_i^\star$ is the global optimum. An advantage of the tunneling method is that during the tunneling phase, it always tries to find a local optimum with a lower function value than the current one, and it ignores all irrelevant local minima that have higher function values regardless of how many there are. Thus, without knowing every local minima, the global optimum can be approached by finding local minima in a orderly fashion.

When solving the tunneling function (2) for $\mathbf{x}^o \neq \mathbf{x}^\star$, it is convenient to introduce a pole with strength λ to deflate the zero at $\mathbf{x}^\star$ as this facilitates moving away from the current minimum. The tunneling function is then redefined as:

$$T(\mathbf{x}, \mathbf{x}^\star, \lambda, f(\mathbf{x}^\star)) := \frac{f(\mathbf{x}) - f(\mathbf{x}^\star)}{\| \mathbf{x} - \mathbf{x}^\star \|^{2\lambda}} \tag{6}$$

Both tunneling functions in Eqns. (2) and (6) are plotted in Fig. 2 for comparison.

An adequate pole strength λ in Eqn. (6) can be found by starting with $\lambda = 1$ and from the offset point $\mathbf{x} = \mathbf{x}^\star + \varepsilon\mathbf{e}$, where ε is a small number and $\mathbf{e}$ is an uniformly distributed random unitary vector; the offset avoids division by zero in Eqn. (6). If $T(\mathbf{x}, \mathbf{x}^\star, \lambda, f(\mathbf{x}^\star)) > 0$, λ has an adequate value; otherwise if $T(\mathbf{x}, \mathbf{x}^\star, \lambda, f(\mathbf{x}^\star)) = 0$, λ is increased by a small increment.

Tunneling Method for Constrained Optimization Problems

When applying the tunneling method to constrained optimization problems, cf. Eqn. (7), the constraints also need to be taken into account in the tunneling phase.

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}) \leq 0 \\ & \mathbf{h}(\mathbf{x}) = 0 \end{aligned} \tag{7}$$

In [27], the active constraints are added to the tunneling function and Newton's method is used to find a root that is feasible. In this work, we use the same non-linear programming optimizer utilized in the optimization phase to solve the tunneling problem in Eqn. (6). Therefore, the constraints are taken care of by the optimizer. Moreover, to effectively get a starting point $\mathbf{x}_{i+1}^o$ for the next optimization phase that is away from the current local minimum $\mathbf{x}_i^\star$, we have to consider three different possibilities:

1. $\mathbf{x}_i^\star$ is located inside the feasible region, and it is a solution to the unconstrained optimization problem

$$\min_x f(x) \tag{8}$$

2. $\mathbf{x}_i^\star$ is located on the boundary the feasible region, and it is a solution of Eqn. (8).

3. $\mathbf{x}_i^\star$ is located on the boundary the feasible region, but it is not a solution of Eqn. (8), i.e., $\nabla f(\mathbf{x}_i^\star) \neq \mathbf{0}$.

For cases 1 and 2, there is no $\mathbf{x}$ in a neighborhood of $\mathbf{x}_i^\star$ that satisfies $f(\mathbf{x}) = f(\mathbf{x}_i^\star)$, therefore we can construct a pole on $\mathbf{x}_i^\star$. For case 3, where $\mathbf{x}_i^\star$ is a constrained minimum,there exist designs $\mathbf{x}$ in a neighborhood of $\mathbf{x}_i^\star$ that satisfy $f(\mathbf{x}) \leq f(\mathbf{x}_i^\star)$; however they are located in the infeasible region. In this case, attempts to make $T \leq 0$ while reducing the infeasibility will drive the iterates back to the constrained minimum $\mathbf{x}_i^\star$. In order to avoid such behavior, a pole $\mathbf{x}_c$ is added between $\mathbf{x}$ and $\mathbf{x}_i^\star$, and the pole strength is computed accordingly for each $\mathbf{x}$ in every iteration. Therefore, the tunneling function for case 3 is defined as

$$T(\mathbf{x}, \mathbf{x}^\star, \lambda, f(\mathbf{x}^\star)) := \frac{f(\mathbf{x}) - f(\mathbf{x}^\star)}{\| \mathbf{x} - \mathbf{x}_c \|^{2\lambda}} \tag{9}$$

In the above equation,

$$\mathbf{x}_c := \beta \mathbf{x}_i^\star + (1-\beta)\mathbf{x} \tag{10}$$

where $\beta \in [0,1]$. The the pole strength λ is computed explicitly as in [27] to avoid going back to $\mathbf{x}_i^\star$:

$$\lambda = \frac{(\mathbf{x} - \mathbf{x}_c)}{2 \mid f(\mathbf{x}) - f(\mathbf{x}_i^\star) \mid} \tag{11}$$

Movable Pole

In the tunneling phase, it is possible that iterates $\mathbf{x}$ get stuck in a local optimum of the tunneling function where $T > 0$. This behavior can be detected if only a small change of T occurs between successive iterates during the tunneling phase. When this happens, a movable pole can be added to the tunneling function [26, 27]. The tunneling function is now modified as

$$T(\mathbf{x}, \mathbf{x}^\star, \lambda, f(\mathbf{x}^\star)) := \frac{f(\mathbf{x}) - f(\mathbf{x}^\star)}{\| \mathbf{x} - \mathbf{x}^\star \|^{2\lambda} \| \mathbf{x} - \mathbf{x}_m \|^{2\eta}} \tag{12}$$

where $\mathbf{x}_m$ is the movable pole and $\mathbf{x}_m$ is set to the last iterate $\mathbf{x}_k$ when slow convergence was detected. Whenever iterates get stuck in another local minimum of the tunneling function, $\mathbf{x}_m$ and η are updated, hence the name movable. We refer readers to [26] for the calculation of η.

For the tunneling function of case 3 in Eqn. (9), $\mathbf{x}_m$ will be also used to enforce the poles at $\mathbf{x}_c$. In the first iteration of tunneling, $\mathbf{x}_m = \mathbf{x}_i^\star$. In the subsequent iterations, we compute the projection of ∇f onto the line $\mathbf{x} - \mathbf{x}_c$ as follow:

$$\| f_{\mathbf{x}proj} \| = \| f_{\mathbf{x}}(\mathbf{x}) - \frac{\mid f_{\mathbf{x}}^T(\mathbf{x})(\mathbf{x} - \mathbf{x}_c) \mid}{\| \mathbf{x} - \mathbf{x}_c \|^2} \cdot (\mathbf{x} - \mathbf{x}_c) \| \tag{13}$$

If $\| f_{\mathbf{x}proj} \| \geq \varepsilon_c$, then $\mathbf{x}_m = \mathbf{x} + \tau_c \frac{f_{\mathbf{x}proj}}{\| f_{\mathbf{x}proj} \|}$. ε_c is a small number and $\tau_c \in [0,1]$. If, on the other hand, $\| f_{\mathbf{x}proj} \| < \varepsilon_c$, we use bisection starting from $\beta_c = 1$ to find $\mathbf{x}_m = \beta_c \mathbf{x}_c + (1-\beta_c)\mathbf{x}$ until $\| \mathbf{x} - \mathbf{x}_m \| \leq \tau_c$.

Multiple Local Minima

When finding a new starting point $\mathbf{x}_{i+1}^o$ during the tunneling phase, it is possible that $\mathbf{x}_{i+1}^o$ is a local optimum at the same level of $\mathbf{x}_i^\star$ (i.e. $f(\mathbf{x}_{i+1}^o) = f(\mathbf{x}_i^\star)$). To prevent $\mathbf{x}$ from going back to previously found minima, the tunneling function is modified as

$$T(\mathbf{x}) := \frac{f(\mathbf{x}) - f(\mathbf{x}^\star)}{\| \mathbf{x} - \mathbf{x}_m \|^{2\eta} \prod_{i=1}^{\ell} \| \mathbf{x} - \mathbf{x}_i^\star \|^{2\lambda_i}} \tag{14}$$

where there are ℓ local minima found at the same level. The arguments of tunneling function T are omitted for clarity.

Stopping Criteria

If we cannot find a feasible $\mathbf{x}$ that satisfies $T(\mathbf{x}) \leq 0$, this implies the current local minimum $\mathbf{x}_i^\star$ is the global optimum. However, there is no rigorous test for the existence of a solution to the tunneling function. In [27], the following stopping criteria are suggested: 1) failure to find a solution after a specified maximum number of initial points is reached, where the initial point can be computed as $\mathbf{x} = \mathbf{x}_i^\star + \varepsilon \mathbf{e}$, with $\mathbf{e}$ a random unitary vector; and 2) failure to find a solution of the tunneling function after a specified maximum number of function evaluations is reached.

The tunneling method is summarized by the flow chart in Fig. 3.

PROBLEM SETUP

The general topology optimization problem is formulated as

$$\begin{aligned} &\min_{\mathbf{x}} f(\mathbf{u}(\mathbf{x}), \mathbf{x}) \\ &\text{subject to} \\ &g_i(\mathbf{u}(\mathbf{x}), \mathbf{x}) \leq 0,\ i = 1, \ldots, m \\ &\underline{\mathbf{x}} \leq \mathbf{x} \leq \bar{\mathbf{x}} \end{aligned} \tag{15}$$

where f is the objective function and g_i is the ith constraint, $i = 1, \ldots, m$.$\mathbf{u}$ is the displacement and $\mathbf{x}$ is the vector of design variables. $\underline{\mathbf{x}}$and $\bar{\mathbf{x}}$ are the lower and upper bounds of the design variables respectively. For density-based optimization with SIMP method, the design variables $\mathbf{x}$ are the elemental densities. For topology optimization with discrete geometric primitives, the design variables $\mathbf{x}$ are the geometric parameters of the primitives plus one size variable for each primitive. This size variable is penalized in the same spirit of SIMP, and it is an indicator of

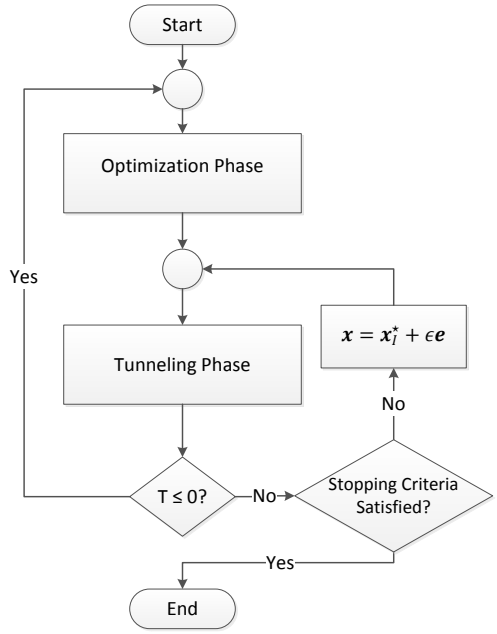

FIGURE 3: Summary of the tunneling method.

whether a primitive is void or solid. This penalized size variable greatly facilitates removing primitives from the design, and it is a hallmark of the geometry projection method.

Unlike the work in [27], here we convert the root-finding problem corresponding to the solution of the nonlinear tunneling function with the following minimization problem:

$$\begin{aligned} &\arg\min_{\mathbf{x}} T(\mathbf{u}(\mathbf{x}),\mathbf{x},\mathbf{x}^\star) \\ &\text{subject to} \\ &g_i(\mathbf{u}(\mathbf{x}),\mathbf{x}) \le 0,\ i=1,\ldots,m \\ &\quad \underline{\mathbf{x}} \le \mathbf{x} \le \bar{\mathbf{x}} \end{aligned} \tag{16}$$

where T is the tunneling function and $\mathbf{x}^\star$ is the design obtained from the previous optimization phase. By doing this, we can apply the same optimizer for both the optimization and tunneling phases. When the the minimum found by Eqn. (16) is larger than zero, the tunneling fails and we restart the tunneling from a new initial point.

In the numerical examples provided in the next section, we limit the tunneling method to only perform two tunneling steps due to the high computational expense to perform a complete search for the global optimum. The optimization phase in Eqn. (15) terminates when: 1) the absolute change in the objective function value between iterations $k+1$ and k satisfies $\mid f_{k+1} - f_k \mid \le \varepsilon_f(1+\mid f_{k+1} \mid)$, where ε_f is a specified tolerance; or 2) a specified maximum number of iterations $I^O_{\max}$ is reached. The termination criteria for the tunneling phase in Eqn. (16) iare: 1) the tunneling function $T \le 0$ and $\mid f_k - f(\mathbf{x}^\star) \mid > 0.01 * \mid f(\mathbf{x}^\star) \mid$ while all constraints are satisfied; or 2) a specified maximum number of iterations $I^T_{\max}$ is reached.

EXAMPLES

We present several numerical examples to demonstrate the proposed approach. For all examples, $\varepsilon_f = 0.00001$ and $I^O_{\max} = 200$. In Eqn. (10), β=0.2. ε_c and τ_c are set to 0.1. To perform the optimization in both phases, we employ the Method of Moving Asymptotes (MMA) in [32, 33]. However, other non-linear programming algorithms such as sequential quadratic programming could also be used. A move limit of 0.1 is enforced to each MMA design update. We note that a different selection of the foregoing parameters could have an effect on the efficiency of the tunneling approach, however this effect is not investigated in the current work and we defer such study to future work.

Michell Bridge with Density-Based Topology Optimization

The first example corresponds to compliance minimization subject to a volume constraint using density-based topology optimization. The design domain, boundary conditions, loading and initial density field are shown in Fig. 4. A point load $F = 1$ is applied to the midpoint of the bottom edge of the design domain. The design domain is meshed with 150×50 bilinear quadrilateral elements of uniform size $h = 1$. Instead of using an uniform density field as initial design, which will generally lead to a good optimum for the compliance problem, we employ a uniformly distributed random density field, generated with MATLAB's `rng` function and a seed of 1. This will allow us to better demonstrate effectiveness of the tunneling method. For this example, we em-

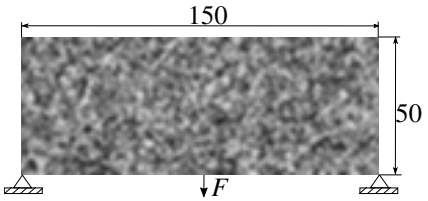

FIGURE 4: Configurations of Michell's bridge problem and the intial density field.

ploy the 88 lines Matlab code of [34] and replaced the optimizer with MMA. The penalization power p in Eqn. (1) is set to 3 and the volume fraction constraint limit is set to 0.3. We use a density filter with size set to $2h$.

We first let the optimization run without any stopping criteria or tunneling. The design and compliance C for iterations 200

(a) $C = 23.234$ (b) $C = 23.039$

FIGURE 5: Density-based designs for Michell bridge at iterations 200 and 2000.

and 2000 are shown in Fig. 5. Comparing the results in Fig. 5, we only see a 0.8% improvement in the compliance after an additional 1800 iterations, and no appreciable change in the topology of the design. We observe that the optimization slowly decreases the compliance by moving the boundary of the design once a solid-void design has been found. This behavior is also reported in [2,12]. The slow convergence indicates that the optimization is attracted to a local optimum, which lies in a neighborhood where the objective function becomes nearly flat.

Next, we apply tunneling to this problem and set the maximum number of iterations $I^T_{\max}$ to 200 to allow for a careful search during the tunneling phase. The iteration histories of the objective and constraint during the optimization phases are plotted in Fig. 6 along with the designs at the end of each optimization phase. The magnitude of the objective function values and the number of iterations for all phases are summarized in Tab. 1. By comparing the designs at the end of the first and third optimization phases, we observe that the tunneling technique allows the optimization to move away from one local optimum to another with smaller compliance. Moreover, the tunneling renders a topological change of the suboptimal design from the first optimization phase, thus circumventing the issue of slow moving boundaries. With a total of 479 iterations including the tunneling phase, we found a design with a compliance of $C = 22.938$ that is lower than the design in Fig. 5b with $C = 23.039$ obtained after 2000 iterations without tunneling.

Tunneling with Geometric Primitives

For the following examples, we apply the tunneling method to topology optimization with discrete geometric primitives. For simplicity, all examples emply the offset-surface 2-d bars described in [16]. The vector of design variables is $\mathbf{x} = \{\mathbf{z}_1^T, \mathbf{z}_2^T, \ldots, \mathbf{z}_{N_q}^T\}$, where $\mathbf{z}_i = [\mathbf{x}_{io}, \mathbf{x}_{if}, \alpha_i]$ is the vector of design variables for bar i, $\mathbf{x}_{io}$, $\mathbf{x}_{if}$ are the endpoints of its medial axis, and α_i is its size variable. The width of all bars is fixed. We refer the reader to [16, 19] for a detailed formulation of the geometry projection for these primitives and the corresponding sensitivity analysis. Although we only provide examples with 2-d bar problems, the tunneling method can be readily extended to the topology optimization of other primitives such as the 3-d flat and curved plates employed in [17, 18, 20].

For all of the following examples, the design domains are

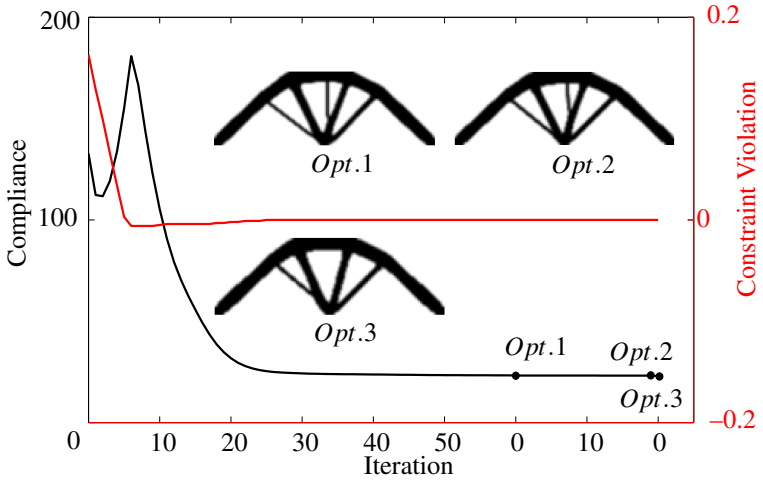

FIGURE 6: Convergence history for optimization phases. Opt. i correponds to the design at the end of ith optimization phase.

discretized with uniform bilinear quadrilateral elements. The 2-d bars we consider are made of a material that is homogeneous, isotropic and linear elastic with Young's modulus $E = 100000$ and Poisson's ratio $\nu = 0.3$. In the void region where bars are not present, we impose a lower bound $\rho_{\min} = 0.0001$ on the projected density to circumvent an ill-posed analysis. The maximum number of iterations $I^T_{\max}$ is set to 100. For all examples with geometric primitives, we implement our method in a C++ code that uses the deal.II library [35–37] to perform the finite element analysis.

Cantilever Beam. The first example consists of a cantilever beam with a downward tip load $F = 10$ applied on the bottom right corner of the design domain as shown in Fig. 7. Our past experience indicates that the optimization is prone to converging to a suboptimal design when starting with an initial design made of many bars. Therefore, in this example, the initial design is seeded with 100 bars of fixed width $w = 0.5$. We use a fixed 160×40 elements grid for the analysis. The objective function for this example is the structural compliance and we impose a volume fraction constraint of 0.3.

The iteration histories of the objective and constraint during the optimization phases are shown in Fig. 8. We observe that, unlike the design obtained at the end of the first optimization phase, the design obtained in the last optimization phase has diagonal members that fully connect the top and bottom portions of the beam, which is expected in optimal 2-d frames [38]. In terms of magnitude, the final design is 3.4% less compliant than the design obtained at the end of the first optimization phase, as shown in Tab. 1.

When the tunneling phase is performed, we always obtain a

new starting point whose compliance is the same or lower than that of the previous minimum. However, in Fig. 8 we note that when we start a new optimization phase after a tunneling phase, the compliance jumps to a large value before it starts decreasing again. We suspect this is due to the particular optimization algorithm we are employing (that is, the specific version of MMA we are currently using), however more extensive numerical experiments beyond the ones presented in this paper are required to ascertain, and if needed alleviate, the origin of these jumps. Nevertheless, we note that the optimization phase consistently converges to a better local minimum than the previous one in a few iterations.

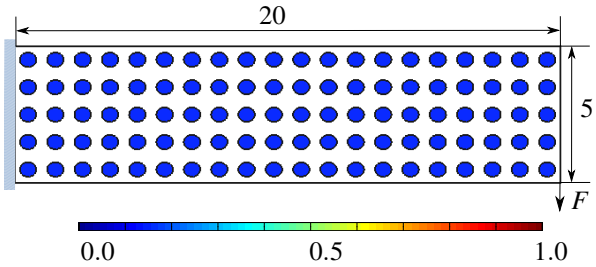

FIGURE 7: Initial design, geometry, loads and boundary conditions for the compliance-based topology optimization with discrete bars of a cantilever beam. Bar color denotes the size variable α.

2-d L-bracket. The second example consists of the stress minimization subject to a volume fraction constraint of a 2-d L-bracket. This is a widely used benchmark for stress-based topology optimization. The design envelope, loading, boundary conditions and initial design are shown in Fig. 9. The objective function for this example is the p-norm global stress measure given by

$$\sigma_{PN} := \left(\int_{\Omega} (\sigma_{VM}(\mathbf{x})^P \, dv \right)^{1/P} \tag{17}$$

where σ_{VM} is the von Mises stress, P is the p-norm power and Ω is the design domain. We refer the reader to [19] for details on the formulation and the design sensitivities. As discussed in [19], the stress-based topology optimization with discrete bars converges to a poor, suboptimal design if it starts from a disconnected initial design. The tunneling method proposed in this work can alleviate this behavior by helping the optimization move away from these poor minima. In this example, we use the same initial design made of disconnected bars used in [19] and shown in Fig. 9. All bars have a fixed width of 5. For this example, a volume fraction constraint with constraint value of 0.35 is imposed. To avoid an artificial stress concentration near the region where the load is applied, the force $F = 3$ is distributed over six nodes along the vertical edge. The design domain is meshed with uniform quadrilateral elements of size $h = 1$, which results in 6400 elements.

The result of the optimization is shown in Fig. 10. The first optimization phase terminates at an undesired design. The subsequent tunneling and optimization phases improve the design and we end up with a design that has almost half the p-norm stress value in the structure than the first design with the same amount of material.

However, it is worth noting that unlike the stress minimization problem, the tunneling technique will not be of help for the problem of minimizing volume subject to a stress constraint when starting from an initial design with disconnected bars. The later problem is too non-linear and far away from the feasible region due to the lack of an uninterrupted load path between the portion of the boundary where traction loads are applied and the portion where Dirichlet boundary conditions are imposed. Thus, it is very difficult for the optimizer to reach the feasible region in the first place so that it can find a local minimum. In other words, tunneling cannot do much if a feasible design cannot be found in the first optimization phase.

2-d L-bracket with multiple design constraints. The final example we present corresponds to minimization of stress subject to multiple geometric constraints. As discussed in [18], the optimization can get easily stuck in a poor local minimum in the presence of many geometric constraints. In this example, we introduce two geometric constraints in addition to the volume fraction constraint: 1) a placement constraint that ensures bars lie entirely within the design domain [20], which prevents cuts to the primitives that may be difficult to manufacture; 2) a length constraint that limits the maximum length of the bars and that can reflect, for example, commercial availability of stock material. The initial design, mesh, boundary condition and loading are the same as those of the previous example. The length constraint is set to 40 so that the maximum length any bar can have is 40. In a similar manner to the aggregation of stresses of Eqn. (17), we use an aggregate function to impose a maximum length on all bars via a single constraint in the optimization.

The iteration history of the optimization phases is plotted in Fig. 11. The magnitude of the global stress measure and the number of iterations for all of the optimization and tunneling phases are summarized in Tab. 1. Similar to the last example, a poor suboptimal is attained after the initial optimization phase. After the tunneling technique has taken effect, a much better design is obtained at the end of the third optimization phase. The optimization is able to move bars away from the reentrant corner

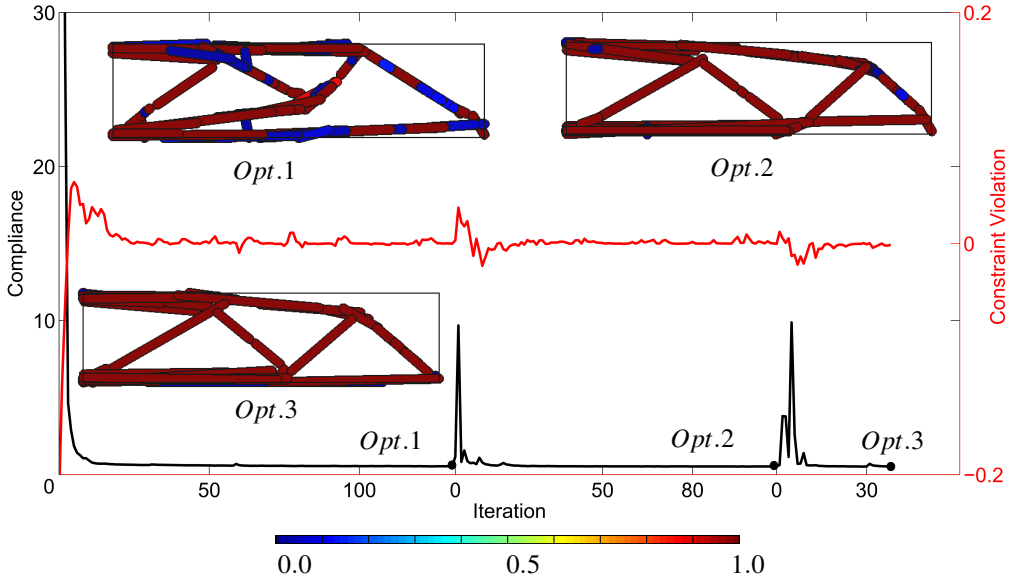

FIGURE 8: Iteration history during optimization phases of cantilever beam design. Opt. *i* corresponds to the design at the end of the *i*-th optimization phase. Bar color denotes the size variable.

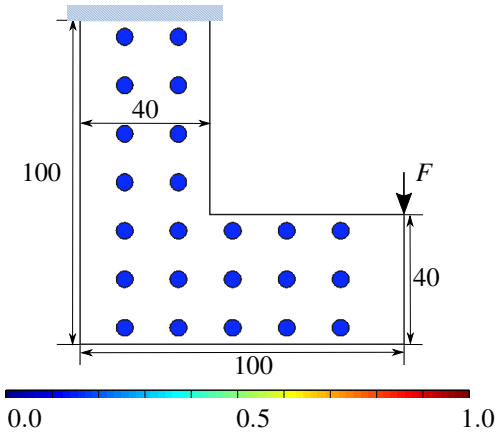

FIGURE 9: Initial design, geometry, loads and boundary conditions for the stress-based topology optimization with discrete bars of a 2-d L-shape bracket. Color denotes the size variable α.

to avoid stress concentration. In the presence of the multiple design constraints, it takes more iterations to find a lower level in the tunneling phases. Nonetheless, subsequent local minima are found that successively improve the design while satisfying all geometric constraints.

CONCLUSIONS

In this work, we propose a framework to couple tunneling to topology optimization to find better minima. Through numerical examples, we demonstrate that the proposed method allows the optimization to move from a suboptimal design to a better one in topology optimization problems, both with the density-based method and the geometry projection method. For problems that are highly non-convex, such as the stress minimization problem in geometry projection problems, the tunneling method can significantly improve the design. We believe the tunneling framework can be readily extended to other responses, design parameterizations (e.g., the level set method), and nonlinear optimization methods.

REFERENCES

[1] Bendsøe, M. P., and Kikuchi, N., 1988. "Generating optimal topologies in structural design using a homogenization method". *Computer methods in applied mechanics and engineering,* ***71***(2), pp. 197–224.

[2] Sigmund, O., and Maute, K., 2013. "Topology optimization approaches". *Structural and Multidisciplinary Optimization,* ***48***(6), pp. 1031–1055.

[3] Bendsøe, M. P., and Sigmund, O., 2013. *Topology optimization: theory, methods, and applications*. Springer Science & Business Media.

[4] Bendsøe, M. P., 1989. "Optimal shape design as a material distribution problem". *Structural optimization,* ***1***(4), pp. 193–202.

[5] Zhou, M., and Rozvany, G., 1991. "The COC algorithm, Part II: topological, geometrical and generalized shape optimization". *Computer Methods in Applied Mechanics and Engineering,* ***89***(1-3), pp. 309–336.

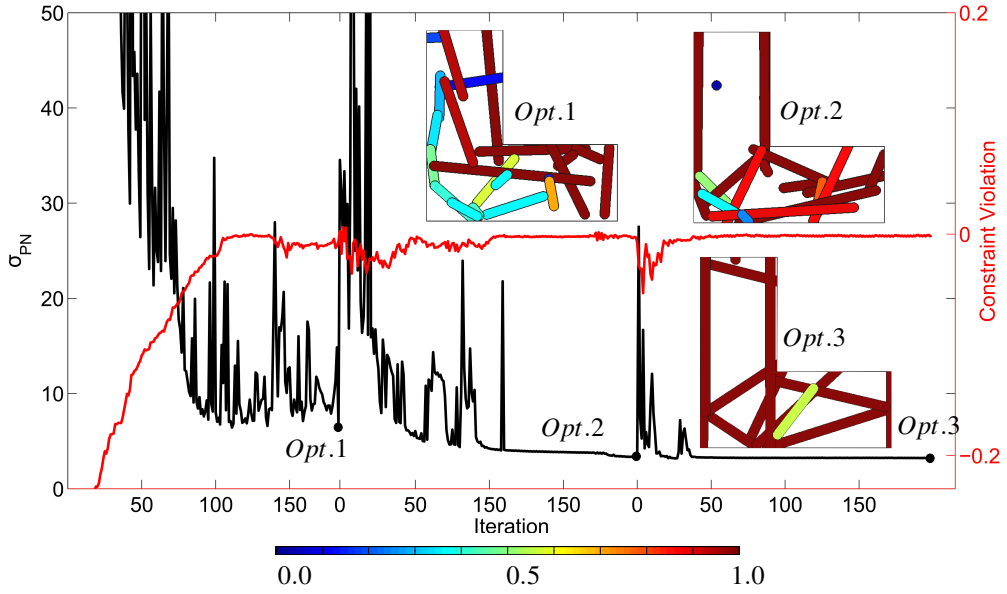

FIGURE 10: Iteration history during optimization phases of L-shape design. Opt. *i* corresponds to the design at the end of the *i*-th optimization phase. Bar color denotes the size variable.

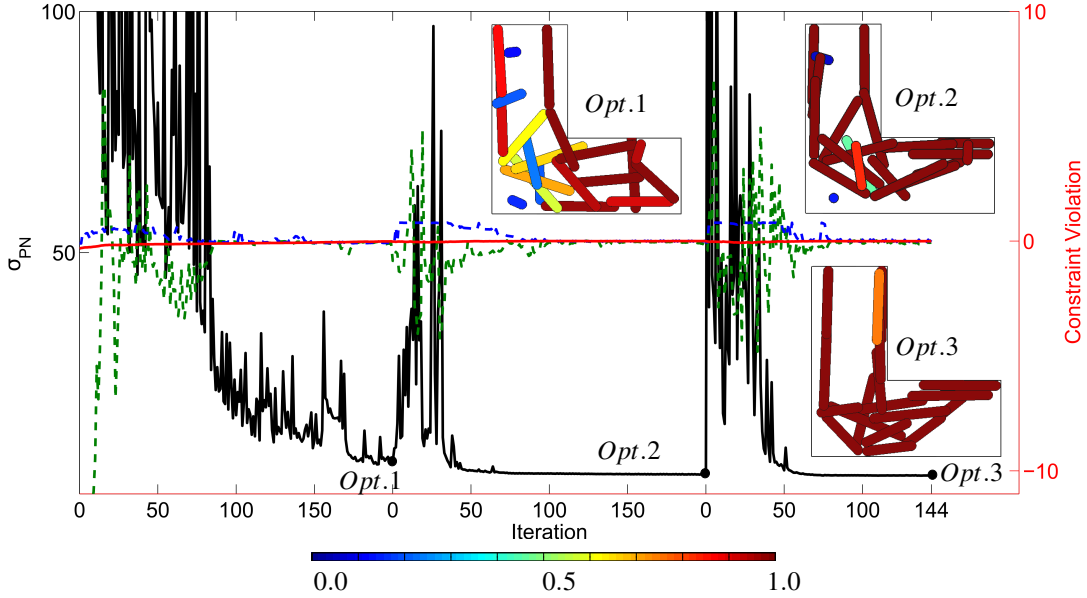

FIGURE 11: Iteration history during optimization phases of L-shape design with multiple geometric constraints. Opt. *i* corresponds to the design at the end of the *i*-th optimization phase. Bar color denotes the size variable. The red solid, blue dashed and green dashed lines correspond to the constraint violation for the volume fraction, placement and length constraints respectively.

[6] Petersson, J., 1999. "A finite element analysis of optimal variable thickness sheets". *SIAM journal on numerical analysis,* ***36***(6), pp. 1759–1778.

[7] Sigmund, O., 1994. "Design of material structures using topology optimization". PhD thesis, Technical University of Denmark Denmark.

[8] Sigmund, O., 1997. "On the design of compliant mechanisms using topology optimization". *Journal of Structural Mechanics,* ***25***(4), pp. 493–524.

[9] Petersson, J., and Sigmund, O., 1998. "Slope constrained topology optimization". *International Journal for Numerical Methods in Engineering,* ***41***(8), pp. 1417–1434.

Problem	Objective	Constraint	Opt. 1	Tunnel 1	Opt. 2	Tunnel 2	Opt. 3	Improve
Ex. 1	Compliance	Volume	23.303/59	23.188/200	23.169/19	22.939/200	22.938/1	1.6%
Ex. 2	Compliance	Volume	0.538/130	0.532/82	0.528/107	0.522/74	0.520/38	3.4%
Ex. 3	Stress (σ_{PN})	Volume	6.448/184	6.372/6	3.341/200	3.306/60	3.238/200	49.8%
Ex. 4	Stress (σ_{PN})	Volume, Placement, Length	7.141/200	6.605/100	4.039/200	3.983/147	3.802/144	46.8%

TABLE 1: Summary of results for all examples. The first value in a pair (/) is the magnitude of the objective function and the second is the number or iterations for each phase. The last column lists the percentage improvement between designs at the end of the last and first optimization phases.

[10] Haber, R. B., Jog, C. S., and Bendsøe, M. P., 1996. "A new approach to variable-topology shape design using a constraint on perimeter". *Structural optimization,* ***11***(1-2), pp. 1–12.

[11] Rojas-Labanda, S., and Stolpe, M., 2015. "Benchmarking optimization solvers for structural topology optimization". *Structural and Multidisciplinary Optimization,* ***52***(3), pp. 527–547.

[12] Rojas-Labanda, S., Sigmund, O., and Stolpe, M., 2017. "A short numerical study on the optimization methods influence on topology optimization". *Structural and Multidisciplinary Optimization,* ***56***(6), pp. 1603–1612.

[13] Sigmund, O., and Petersson, J., 1998. "Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima". *Structural optimization,* ***16***(1), pp. 68–75.

[14] Rozvany, G., Zhou, M., and Sigmund, O., 1994. "Optimization of topology".

[15] van Dijk, N. P., Maute, K., Langelaar, M., and Van Keulen, F., 2013. "Level-set methods for structural topology optimization: a review". *Structural and Multidisciplinary Optimization,* ***48***(3), pp. 437–472.

[16] Norato, J., Bell, B., and Tortorelli, D., 2015. "A geometry projection method for continuum-based topology optimization with discrete elements". *Computer Methods in Applied Mechanics and Engineering,* ***293***, pp. 306–327.

[17] Zhang, S., Norato, J. A., Gain, A. L., and Lyu, N., 2016. "A geometry projection method for the topology optimization of plate structures". *Structural and Multidisciplinary Optimization*, pp. 1–18.

[18] Zhang, S., and Norato, J. A., 2017. "Optimal design of panel reinforcements with ribs made of plates". *Journal of Mechanical Design,* ***139***(8), p. 081403.

[19] Zhang, S., Gain, A. L., and Norato, J. A., 2017. "Stress-based topology optimization with discrete geometric components". *Computer Methods in Applied Mechanics and Engineering,* ***325***, pp. 1–21.

[20] Zhang, S., Gain, A. L., and Norato, J. A. "A geometry projection method for the topology optimization of curved plate structures with placement bounds". *International Journal for Numerical Methods in Engineering*.

[21] Norato, J., Haber, R., Tortorelli, D., and Bendsøe, M. P., 2004. "A geometry projection method for shape optimization". *International Journal for Numerical Methods in Engineering,* ***60***(14), pp. 2289–2312.

[22] Guo, X., Zhang, W., and Zhong, W., 2014. "Doing topology optimization explicitly and geometrically—a new moving morphable components based framework". *Journal of Applied Mechanics,* ***81***(8), p. 081009.

[23] Zhang, W., Zhang, J., and Guo, X., 2016. "Lagrangian description based topology optimization—a revival of shape optimization". *Journal of Applied Mechanics,* ***83***(4), p. 041010.

[24] Zhang, W., Yuan, J., Zhang, J., and Guo, X., 2016. "A new topology optimization approach based on moving morphable components (mmc) and the ersatz material model". *Structural and Multidisciplinary Optimization,* ***53***(6), pp. 1243–1260.

[25] Zhang, W., Li, D., Yuan, J., Song, J., and Guo, X., 2017. "A new three-dimensional topology optimization method based on moving morphable components (mmcs)". *Computational Mechanics,* ***59***(4), pp. 647–665.

[26] Levy, A. V., and Gómez, S., 1985. "The tunneling method applied to global optimization". *Numerical optimization,* ***1981***, pp. 213–244.

[27] Gomez, S., and Levy, A. V., 1982. "The tunnelling method for solving the constrained global optimization problem with several non-connected feasible regions". In *Numerical analysis*. Springer, pp. 34–47.

[28] Barron, C., and Gomez, S., 1991. "The exponential tunneling method". pp. 1–23.

[29] Gómez, S., del Castillo, N., Castellanos, L., and Solano, J., 2003. "The parallel tunneling method". *Parallel Computing,* ***29***(4), pp. 523–533.

[30] Sigmund, O., 2011. "On the usefulness of non-gradient approaches in topology optimization". *Structural and Mul-*

tidisciplinary Optimization, ***43***(5), pp. 589–596.

[31] Kitayama, S., and Yamazaki, K., 2005. "Generalized random tunneling algorithm for continuous design variables". *Journal of Mechanical Design,* ***127***(3), pp. 408–414.

[32] Svanberg, K., 1987. "The method of moving asymptotes-a new method for structural optimization". *International journal for numerical methods in engineering,* ***24***(2), pp. 359–373.

[33] Svanberg, K., 2002. "A class of globally convergent optimization methods based on conservative convex separable approximations". *SIAM journal on optimization,* ***12***(2), pp. 555–573.

[34] Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S., and Sigmund, O., 2011. "Efficient topology optimization in matlab using 88 lines of code". *Structural and Multidisciplinary Optimization,* ***43***(1), pp. 1–16.

[35] Bangerth, W., Hartmann, R., and Kanschat, G., 2007. "deal.II – a general purpose object oriented finite element library". *ACM Trans. Math. Softw.,* ***33***(4), pp. 24/1–24/27.

[36] Bangerth, W., Hartmann, R., and Kanschat, G., 2007. "deal.II – a general purpose object oriented finite element library". *ACM Trans. Math. Softw.,* ***33***(4), pp. 24/1–24/27.

[37] Bangerth, W., Davydov, D., Heister, T., Heltai, L., Kanschat, G., Kronbichler, M., Maier, M., Turcksin, B., and Wells, D., 2016. "The `deal.II` library, version 8.4". *Journal of Numerical Mathematics,* ***24***.

[38] Rozvany, G., 1998. "Exact analytical solutions for some popular benchmark problems in topology optimization". *Structural optimization,* ***15***(1), pp. 42–48.