DTU Library

A short numerical study on the optimization methods influence on topology optimization

Rojas Labanda, Susana; Sigmund, Ole; Stolpe, Mathias

A short numerical study on the optimization methods influence on topology optimization

Susana Rojas-Labanda · Ole Sigmund · Mathias Stolpe

Abstract Structural topology optimization problems are commonly defined using continuous design variables combined with material interpolation schemes. One of the challenges for density based topology optimization observed in the review article Sigmund and Maute (2013) is the slow convergence that is often encountered in practice, when an almost solid-and-void design is found.

The purpose of this forum article is to present some preliminary observations on how designs evolves during the optimization process for different choices of optimization methods. Additionally, the authors want to open a discussion on how to properly define and identify the *boundary translation* that is often observed in practice. The authors hope that these preliminary observations can open for fruitful discussions and stimulate further investigations concerning slowly moving boundaries. Although the discussion is centered on density based methods it may be equally relevant to level-set and phase-field approaches.

S. Rojas-Labanda
Technical University of Denmark, Department of Wind Energy
Frederiksborgvej 399, 4000 Roskilde, Denmark
Tel.: (+45) 8191 0527
E-mail: srla@dtu.dk

O. Sigmund
Technical University of Denmark, Department of Mechanical Engineering
Nils Koppels Allé, Building 404, 2800 Kgs. Lyngby, Denmark
Tel.: (+45) 4525 4256
E-mail: sigmund@mek.dtu.dk

M. Stolpe
Technical University of Denmark, Department of Wind Energy
Frederiksborgvej 399, 4000 Roskilde, Denmark
Tel.: (+45) 2151 8240
E-mail: matst@dtu.dk

1 Introduction

The review article Sigmund and Maute (2013) studies, among many other topics, the way optimization algorithms generate the designs in structural topology optimization, i.e. the design iteration history. The design parametrization is based on density based topology optimization combined with the finite element method for structural analysis and the Solid Isotropic Material with Penalization (SIMP) interpolation scheme (see e.g. Zhou and Rozvany (1991) and Bendsøe and Sigmund (1999)). The computational behaviour reported in Sigmund and Maute (2013) is that the optimization algorithms suggest the final topology in relatively few iterations. A substantial number of iterations are then required to find the final geometry by slowly moving the boundaries.

Sigmund and Maute (2013) use a mechanism design example to show that "the optimization rapidly finds a fairly good design but requires a very large number of iterations for just slight improvements in objective function but rather large changes in geometry." This effect is also illustrated by using two different starting points for a minimum compliance problem resulting in a two-bar truss like structure. From this example, Sigmund and Maute (2013) conclude as follows: "This shows that the SIMP approach is very good in locating a good design for a uniform grey starting guess but that it behaves similar to a phase field method, i.e. it operates with a solid and void phase and changes shape by moving boundaries, when the design has reached or is initialized in a solid-void state."

The underlying reasons for the slowly moving boundaries could be as fundamental as the choice of design parametrization, which would be unfortunate since it is widespread

use and difficult to remedy. The observed behaviour could also be because of the choice of regularization. A sensitivity based filtering technique is used in Sigmund and Maute (2013). It is claimed that the behaviour also occurs for density based filtering and this is supported by the numerical experiments in this study. In fact, it could be a combination of several of these issues or something completely different.

Unfortunately, the numerical studies performed herein do not lead to any definitive conclusions regarding the above. Nevertheless, the study provides insight that may spur further studies in this important challenge. Also, the study gives valuable insight in the convergence behavior of different optimization methods.

2 Problem formulations

Two structural topology optimization problems are considered throughout this article, the classical minimum compliance problem (Bendsøe and Sigmund, 2003) and the compliant mechanism design problem, see e.g. Sigmund (1997) and Sigmund (2009). More details regarding topology optimization problems can be found in e.g. the text book Bendsøe and Sigmund (2003). The problems are formulated in their discretized version. Isotropic material and a regular mesh with a constant density throughout each element are considered. Additionally, only a single and design-independent load is assumed. The SIMP material interpolation scheme is used to produce almost solid-and-void designs. Thus, the stiffness matrix is defined as in Bendsøe (1989), Bendsøe and Sigmund (2003), and Andreassen et al (2011). Finally, a density filter is used in order to regularize the problem (Bourdin, 2001).

The discretized minimum compliance problem in the nested approach is formulated as

$$\begin{aligned} \underset{\mathbf{t}\in\mathbb{R}^n}{\text{minimize}} \quad & \mathbf{u}^T(\mathbf{t})\mathbf{K}(\mathbf{t})\mathbf{u}(\mathbf{t}) \\ \text{subject to} \quad & \mathbf{v}^T\mathbf{t} \leq V \\ & \mathbf{0} \leq \mathbf{t} \leq \mathbf{1}, \end{aligned} \quad (1)$$

with $\mathbf{u}(\mathbf{t}) = \mathbf{K}^{-1}(\mathbf{t})\mathbf{f}$. The design variables are defined by $\mathbf{t}$, the stiffness matrix is represented with $\mathbf{K}(\mathbf{t})$ and the load vector with $\mathbf{f}$. A vector with all entries equal to one is denoted by $\mathbf{1}$.

Finally, the volume constraint is defined as a linear inequality with $\mathbf{v} = (v_1,\dots,v_n)^T \in \mathbb{R}^n$ the relative volume of each element and $0 < V < 1$ the maximum volume fraction allowed.

For the mechanism design problem, a constraint enforcing negative output displacement (u_{out}) is included in order to avoid ending in a bad local minimum.

The mechanism design problem in the nested approach used in this article is formulated as

$$\begin{aligned} \underset{\mathbf{t}\in\mathbb{R}^n}{\text{maximize}} \quad & u_{out} = \mathbf{l}^T\mathbf{u}(\mathbf{t}) \\ \text{subject to} \quad & \mathbf{v}^T\mathbf{t} \leq V \\ & u_{out} \leq 0 \\ & \mathbf{0} \leq \mathbf{t} \leq \mathbf{1}, \end{aligned} \quad (2)$$

The output displacement is defined using a unit length vector $\mathbf{l}$ with zeros in all the degrees of freedom expect at the output. The domain contains an input and an output spring stiffness (denoted k_{in} and k_{out}, respectively). An example of a mechanism design problem is found in Figure 3.

The complete formulation of the problems can be found in, for instance, Rojas-Labanda and Stolpe (2015).

3 Optimization methods and implementation details

The aim of the article is to assess by example, whether the boundary translation might be affected by the optimization solver choice. For this purpose, different optimization algorithms are considered. The classical first-order structural optimization Method of Moving Asymptotes (MMA), (Svanberg, 1987) is compared to the interior point method implemented in IPOPT (Wächter and Biegler, 2006), and a sequential quadratic programming method (SQP) implemented for structural optimization problems see e.g. Boggs and Tolle (1995) and Rojas-Labanda and Stolpe (2016).

All the optimization methods attempt to solve[1] the nested formulation of the topology optimization problem (1) and (2).

Interior point methods, such as IPOPT, approximately solve a sequence of barrier problems

$$\begin{aligned} \underset{\mathbf{x},\mathbf{s}}{\text{minimize}} \quad & f(\mathbf{x}) - \mu_k \phi(\mathbf{x},\mathbf{s}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}) + \mathbf{s} = \mathbf{0}. \end{aligned}$$

Corresponding to the general problem minimize $f(\mathbf{x})$ subject to $g(\mathbf{x}) \leq \mathbf{0}$. Here $f(\mathbf{x})$ is the objective function, $\phi(\mathbf{x}) = \sum_{i=1}^n \log(u_i - x_i) + \sum_{i=1}^n \log(x_i - l_i) + \sum_{i=1}^m \log(s_i)$ is the barrier function, $\mu_k > 0$ is the barrier parameter at the k-th iteration, with $\mu_k \downarrow 0$, and $\mathbf{s}$ is the slack variable (Wächter and Biegler, 2006). The inequality constraints are defined by $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$. To simplify notation, the equality constraints have been removed from the problem formulation.

The *monotone* barrier parameter strategy solves a sequence of barrier sub-problems. Once the sub-problem is solved for a fixed μ_k (and a given tolerance), the barrier parameter is decreased until convergence. It ensures global convergence, but in general, it requires more iterations than

---

[1] In this manuscript the word *solve* (in the context of optimization problems) should be understood as finding a point numerically satisfying the KKT conditions within some prescribed tolerances.

an *adaptive* barrier parameter update strategy, see Nocedal et al (2009). In the latter, the barrier problem is not solved for a fixed value of the barrier parameter. Additionally, μ_k is both allowed to decrease and increase. For large values of the barrier parameter μ, the optimization algorithm not only reduces the objective function but also forces the variables to the interior of the feasible set (due to the barrier function). More details of interior point methods can be found in e.g. Forsgren and Gill (1998), Nocedal and Wright (1999), Byrd et al (1999), and Yamashita (1998). Since the Hessian of the Lagrangian of both problems (in the nested approach) is dense and very expensive, IPOPT uses an approximation of the Hessian based on limited memory BFGS approximations (Dennis and Moré (1977) and Nocedal (1980)). The IPOPT version used in the numerical experiments is 3.10.3 with the default linear solver MUMPS (Amestoy et al, 2000).

On the other hand, SQP methods solve sequences of quadratic sub-problems with linearized constraints. In particular, two sequential programming methods are implemented for structural optimization problems. TopSQP is developed for solving the minimum compliance problem (Rojas-Labanda and Stolpe, 2016), while a general SQP method is implemented for solving the mechanism design problem (Boggs and Tolle, 1995). TopSQP includes two phases, the classical QP subproblem (IQP) and an equality constrained QP problem (EQP) to promote fast convergence, based on Morales et al (2010). Additionally, TopSQP takes advantages of the structure of the problem in order to use partial information of the Hessian and avoid its direct storage and computation. Both IQP and EQP phases use a positive definite approximation of the Hessian (Rojas-Labanda and Stolpe, 2016). On the other hand, the general SQP method (Boggs and Tolle, 1995) solves only the classical QP subproblem (IQP) using a limited memory BFGS approximation. More information about these solvers for structural topology optimization problems can be found in Rojas-Labanda and Stolpe (2015) and Rojas-Labanda and Stolpe (2016).

The stopping criteria for all the methods are based on the 2-norm of the first-order optimality conditions, i.e. the KKT conditions (Nocedal and Wright, 1999). The optimality tolerance for MMA is set to 10^{-4} while for the rest of the methods the value is 10^{-6}. The feasibility tolerance for all solvers is set to 10^{-8}. Additionally, a maximum number of iterations[2] equal to $1,000$ is set. Most of the problem data, such as the magnitude of the load, the element areas, the Young's modulus contrast, the Young's moduli of elasticity values of both, void (E_v) and solid (E_1), the spring stiffness values (k_{in}, k_{out}) and the constraint scaling are identical for all instances and methods. In particular, the Young's modulus contrast is set to $E_1/E_v = 10^6$. However, the objective function is scaled in order to improve the performance of the solvers (see Table 1). MMA, TopSQP, and SQP are im-

---

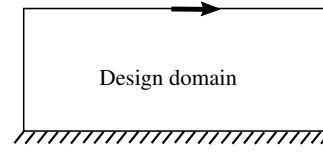2 Number of optimization sub-problems solved.

Fig. 1: Design domain representation for the minimum compliance example.
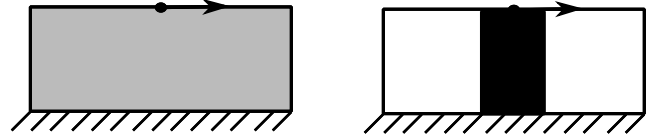

Fig. 2: Representation of two different starting points; a uniform design and a vertical solid bar.
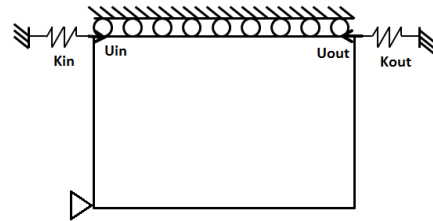

Fig. 3: Compliant mechanism inverter design domain, with its boundary conditions and external load definition (from Rojas-Labanda and Stolpe (2015)).

plemented in MATLAB and the analysis, density filter, and the sensitivity analysis are all based on the code presented in Andreassen et al (2011). Finally, Table 1 lists the values of the parameters used in the numerical experiments. The specific parameter settings for IPOPT can be found in Rojas-Labanda and Stolpe (2015).

4 Numerical experiments

First of all, the boundary translation effect is analysed for the minimum compliance problem (1). Sigmund and Maute (2013) suggest to use the benchmark example described in Figure 1. The minimum compliance problem is solved for two different starting points. The optimal design is a two-bar like structure.

Additionally, the inverter mechanism design problem (see Figure 3) is also taken into consideration.

Throughout the section the slowly moving boundaries are understood in the visual sense as in Sigmund and Maute (2013). The authors of this article consider that this effect

Table 1: Parameters used in the numerical experiments.

Parameter	Description	Value
E_1/E_v	Young's modulus contrast	10^6
E_1	Solid Young's modulus	1
E_v	Void Young's modulus	10^{-6}
scale obj	Objective function scale constant (minimum compliance problem, all solvers)	10^{-2}
scale obj	SQP objective function scale constant (mechanism design problem)	10^2
scale obj	IPOPT objective function scale constant (mechanism design problem)	10^4
scale disp const	$\mathbf{u}_{out}$ constraint scale constant (mechanism design problem, all solvers)	10^2
v	Poisson ratio	0.3
p	SIMP penalization parameter	3
$r_{\min}$	Radius density filter	$0.03 \times nelx$
v_i	Volume scale factor	$\frac{1}{\sqrt{n}}$
opt tol	Optimality tolerance	10^{-4} or 10^{-6}
feas tol	Feasibility tolerance	10^{-8}
max iter	Maximum number of iterations	1,000
μ_0	Initial barrier penalization value for IPOPT$_m$	10

occurs when a good topology is found relatively quickly, but numerous iterations are spent in moving the bars. To the best of our knowledge, no formal definition has been established.

4.1 Slowly moving boundaries using different starting points in the minimum compliance problem

The design domain considered for the compliance minimization problem is shown in Figure 1. The design domain is discretized as in Sigmund and Maute (2013) using 50 by 20 elements. The volume fraction V is set to 0.2. The radius filter is fixed to $1.5 = 0.03 \times 50$. Two different starting guesses are studied; a uniform density (of 0.2) and a straight, solid and vertical bar located in the middle of the design domain, and satisfying the volume constraints, see Figure 2.

This example is suggested in Sigmund and Maute (2013) as a good benchmark problem for the purpose of studying slowly moving boundaries. Table 2 contains the information of the final designs found by the different optimization methods using two different starting points. The table includes the objective function value (compliance) and the number of outer iterations (gradient evaluations) needed to solve the problem. For this numerical experiment, IPOPT is run using both, the monotone (IPOPT$_m$) and the adaptive (IPOPT$_a$) barrier parameter update strategy.

The difference between the number of iterations when different starting points are considered is remarkable. MMA and TopSQP divide the solid bar into two members and then slowly move the boundaries outwards. At every iteration, the design is (visually) almost solid-and-void. In contrast, throughout the first iterations of the interior point methods the design domain turns greyish, see Figures 4 and 5. In particular, the initial design of IPOPT$_m$ disappears and then the two bars are appearing in the correct position (see Figure 5).

Table 2: Objective function value and the number of iterations when the minimum compliance problem (see Figure 1) is solved with MMA, IPOPT, and TopSQP. The table shows the results using two different starting points illustrated in Figure 2. .

Solver	Starting point	Compliance	Iterations
MMA	Uniform	11.257256	24
MMA	Bar	11.536553	149
IPOPT$_a$	Uniform	12.034256	36
IPOPT$_a$	Bar	11.344896	290
IPOPT$_m$	Uniform	11.803645	58
IPOPT$_m$	Bar	11.682916	98
TopSQP	Uniform	11.247049	33
TopSQP	Bar	11.526452	144

The barrier parameter update strategy chosen (monotone vs. adaptive) considerably affects the amount of intermediate design variables (i.e. the amount of grey). The monotone approach turns the solid and void design in an almost completely grey design, while the adaptive strategy has more difficulties in changing the whole design to grey. The scaling of the problem can also considerably affect the behaviour of interior point methods.

In order to understand this effect it is important to keep in mind that interior point methods solve sequences of barrier sub-problems (see Section 3). The bound constraints of the original problem are included in the objective function of the barrier sub-problem, and thus, the bounds are penalized by the barrier parameter. For large values of μ, interior point methods try to reduce both the objective function and the barrier function. However, while μ goes to zero, the relevance of the barrier function decreases.
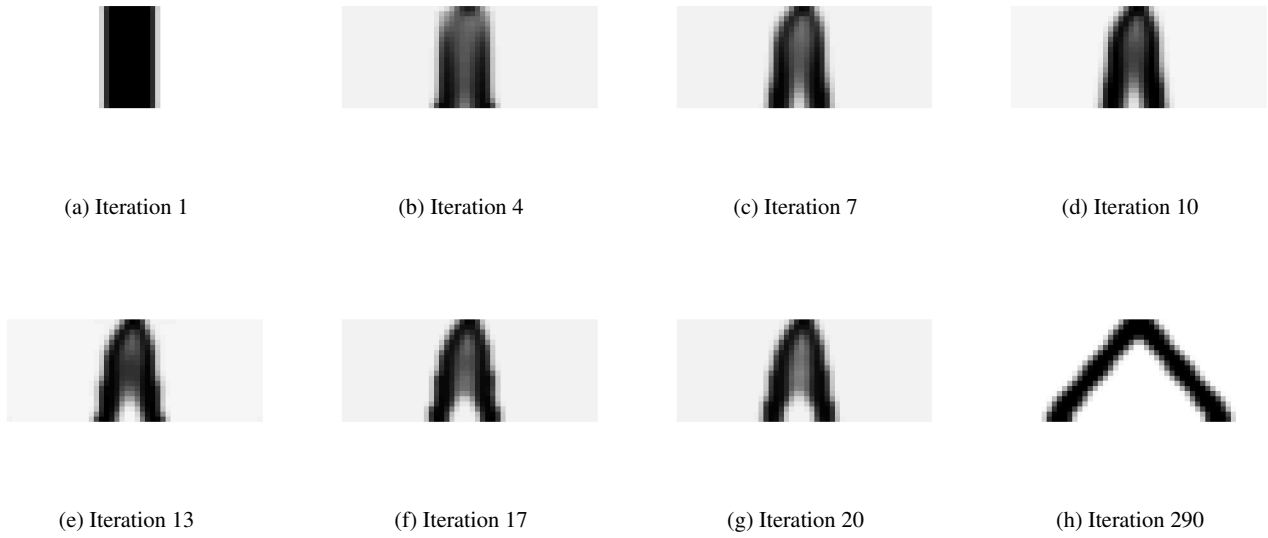
(a) Iteration 1   (b) Iteration 4   (c) Iteration 7   (d) Iteration 10

(e) Iteration 13   (f) Iteration 17   (g) Iteration 20   (h) Iteration 290

Fig. 4: Intermediate and final designs at different optimization iterations for IPOPT$_a$.

(a) Iteration 1   (b) Iteration 4   (c) Iteration 7   (d) Iteration 10

(e) Iteration 13   (f) Iteration 17   (g) Iteration 20   (h) Iteration 98
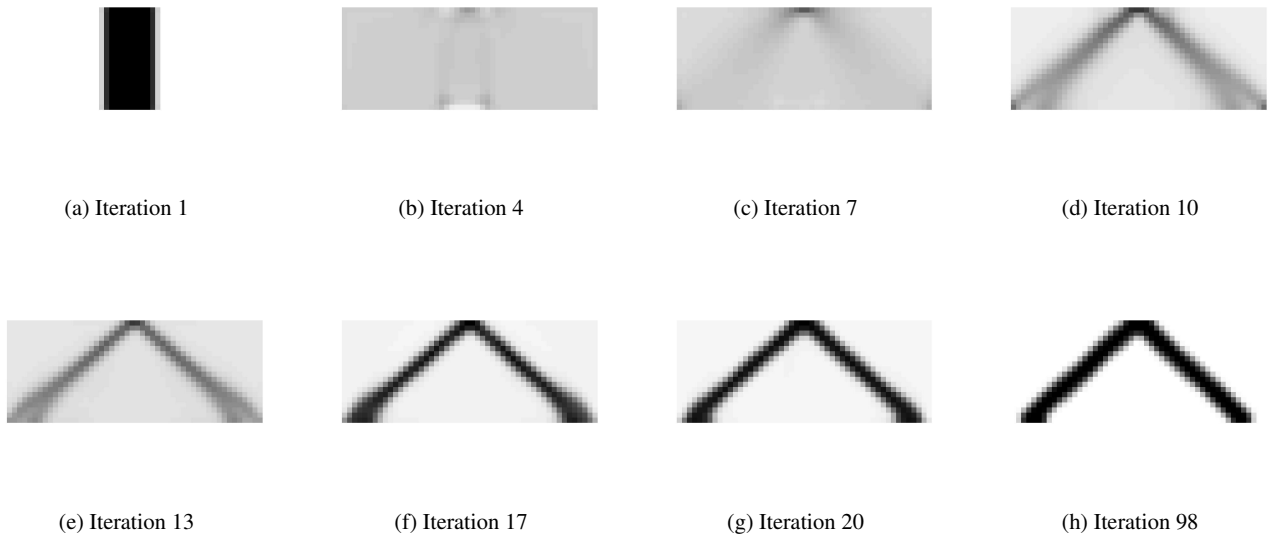
Fig. 5: Intermediate and final designs at different optimization iterations for IPOPT$_m$.

For the minimum compliance problem, the barrier function increases when the design variables tend to the bounds. Therefore, at the beginning of the optimization history, which normally has a relatively large barrier parameter value, the obtained designs are mostly grey. This same reasoning can be used to explain the difference in the design history when different barrier parameter update strategies are used. If the adaptive strategy is chosen (IPOPT$_a$), the barrier penalty decreases fast at the beginning of the optimization process.

Thus, it cannot blur the solid bar completely as it does in the monotone strategy (IPOPT$_m$). This behaviour indicates that the log-barrier term dominates the objective in the barrier problem despite the penalization term (SIMP) in the computation of compliance.

Therefore, it is possible to find a set of parameters such that interior point methods with a solid bar as initial point behave similar to the same method applied to the uniform starting guess. This explains why the number of iterations

for IPOPT$_m$ reported in Table 2 is almost the same for the different starting points.

Remark 1 *During the preliminary numerical experiments, we experienced that the robustness of the solvers is sensitive to the scaling of the problem. The scaling of the objective function is highly important in the behaviour of interior point methods since they balance the minimization of the objective function and the penalization of the bounds. For some poorly scaled problems, the behaviour observed in this article may not be visible. For the same reason IPOPT$_m$ starts with a relatively large barrier parameter equal to $\mu_0 = 10$. This parameter is responsible of penalizing the bounds.*

On the other hand, when linearly constrained nonlinear problems are considered, MMA and TopSQP belong to the same class of methods, i.e. sequential convex approximation methods and thus behave similarly. Both solvers produce intermediate designs which are almost black and white.

Remark 2 *The considered problem is in general nonconvex. The objective function values reported in Table 2 correspond to different local minima. We would like to emphasize that the purpose of this study is not to judge which method finds the best design, although this is of course highly relevant, but instead to illustrate the qualitative the behaviour of the algorithms. For a comparative study of the performance of the solvers we refer to Rojas-Labanda and Stolpe (2015).*

Since interior point methods generally start, or can be instructed to start, with a relatively large value of the barrier parameter, the behaviour of the solver is (relatively) not affected by the starting point. This observation does not mean that using interior point methods can avoid the slowly moving boundaries which are normally observed at the end of the optimization process. A disadvantage is its inability to take advantage of qualified starting guesses.

4.2 Slowly moving boundaries in the inverter mechanism problem

The second example presented in Sigmund and Maute (2013) is the compliant mechanism problem described in e.g. Sigmund (1997) and Sigmund (2009). Figure 3 shows the description of the design domain of the inverter problem. In particular, the design domain is discretized using 80×40 elements, the volume fraction limit is set to $V = 0.3$, and the input and output springs are set to $k_{\text{in}} = 1$ and $k_{\text{out}} = 0.001$, respectively. The radius filter is set to $r_{\text{min}} = 0.03 \times 80 = 2.4$.

The review article Sigmund and Maute (2013) includes an example in which the Optimality Criteria method (OC, (Rozvany and Zhou, 1991), (Zhou and Rozvany, 1991)) needs $1,562$ iterations before reaching the stopping criteria and in which the difference in objective function values between

Table 3: The table contains the objective function value and the number of iterations when the inverter problem is solved with MMA, SQP, and IPOPT$_a$.

Solver	u_{out}	Number of iterations
MMA	-1.9924	$1,000^*$
SQP	-1.9753	143
IPOPT$_a$	-1.9741	325

iteration 100 and $1,562$ is 5.5%. Additionally, it includes a density difference plot between those two designs. In this example, OC slowly moves the bars of the design.

We perform the same study using different optimization methods with the aim of indicating the difference between optimization solvers with respect to the slowly moving boundaries effect. More specifically MMA, SQP, and IPOPT$_a$ are used to solve the mechanism design problem.

Table 3 shows the objective function value (output displacement, u_{out}) and the number of iterations needed for the different solvers. MMA stopped due to the maximum number of iterations ($1,000$). In contrast, SQP and IPOPT$_a$ converge within relatively few iterations.

Remark 3 *The examples in Sigmund and Maute (2013) are solved by the MATLAB code described in Andreassen et al (2011). This code stops if the norm of the difference between two consecutive primal variable vectors is less than some threshold. This criterion thus stops the algorithm if it is not making primal progress. This criterion thus ignores the possibility that the algorithm may be adjusting the working set and the corresponding Lagrange multipliers (which is exactly what is in progress when the boundaries are slowly moving). The differences in stopping criteria, the choice of norms and the scale of the problem explain the difference in the number of iterations reported in Tables 2 and 3, and in Sigmund and Maute (2013).*

MMA relatively fast finds a "good" solid and void design. However, it needs a significant number of iterations to move the bars in order to find better designs. Similar behaviour is observed in SQP, but since it uses a more efficient second-order approximation, the movement of the bars requires fewer iterations. On the other hand, intermediate designs of IPOPT$_a$ contains more grey elements and the solver tries to find a good design without producing prematurely a solid and void design (see Figure 6). These grey designs do not disappear, as it occurs in MMA (and SQP), due to the barrier penalty parameter. The barrier function is penalized at the beginning of the optimization process, and with it, "black-and-white" designs.

IPOPT$_a$ spends most of the iterations in finding an optimal geometry but also in improving the objective function

(a) Iteration 10   (b) Iteration 40   (c) Iteration 70   (d) Iteration 100

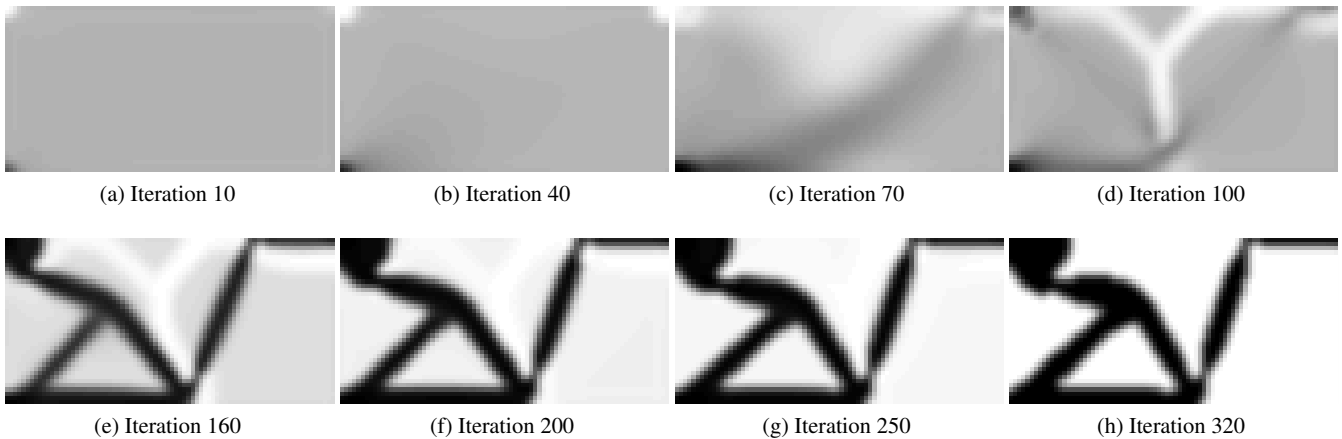(e) Iteration 160   (f) Iteration 200   (g) Iteration 250   (h) Iteration 320

Fig. 6: Intermediate designs at different optimization iterations of IPOPT$_a$.

value. This observation is clearly illustrated in Figures 7, 8, and 9. The objective function value at each iteration, the number of intermediate design variables (variables between 0.01 and 0.95), and the norm of the KKT conditions[3] are presented for each solver.

Figure 7 and 8 show that MMA and SQP quickly improve the objective function and then level out. In contrast, the objective function values of IPOPT$_a$ (see Figure 9) decreases slower and more gradually than in MMA and SQP. The same occurs for the number of grey elements. IPOPT$_a$ spends more iterations to remove all the grey elements, while MMA and SQP remove the intermediate design variables at the very beginning of the optimization process, and then they slowly move the bars.

From the density history of the solvers, it is possible to identify when the designs are "visually good". For MMA, the design is identified in a very early stage, between the 38-th and the 100-th iteration. SQP, in turn, needs about 60 to 70 iterations to define the design. In contrast, IPOPT$_a$ needs around $180 - 200$ iterations, and these intermediate designs still include grey areas.

In addition, Figures 10a, 11a, and 12a, show the ratio of improvement between the intermediate objective function value and the optimal one, i.e it is represented

$$r_{\mathrm{improv}} = \left(\frac{u^*_{out}}{u^i_{out}} - 1\right) \times 100.$$

Here, i indicates the outer optimization iteration. For those positive objective function values, the value represented is the maximum possible, i.e. 100. A design difference between a given intermediate iteration and the final iteration is also presented in Figures 10b, 11b, and 12b. In particular the choice of this intermediate iteration is chosen such as

the percentage of improvement between objective function values is less than or equal to 10% ($r_{\mathrm{improv}} = 10$).

The percentage of improvement in the objective function value throughout the optimization process is clearly differentiated between MMA, SQP, and IPOPT$_a$. Both MMA and SQP at the 40-th iteration, they obtain an objective function value that is "only" 10% higher than the optimal found (at most). However, IPOPT$_a$ needs 194 iterations to achieve this threshold. At this point, the visual difference in the designs of interior point methods are much smaller than the difference in the designs of MMA and SQP. In the two latter cases, the bars have been clearly moved (see Figure 10b and 11b).

These numerical experiments indicate how interior point methods and convex approximation type methods behave. The former spends more iterations without suggesting a good design. However when a "visually good" design is found, it is in fact much closer and accurate to the final design than the "visually similar" designs suggested by MMA and SQP methods.

These preliminary and limited numerical experiments indicate that the choice of optimization method affects slowly moving boundaries, at least visually. The objective function value in solvers such as MMA and SQP decreases too rapidly and the grey elements disappear in few iterations. Thus, the final design is obtained by slowly moving the boundaries. In contrast, IPOPT balances the improvement of objective function value with the reduction of grey elements in the early stage. This produces that the boundaries are properly located and therefore the solver does not need to move them. **These preliminary experiments suggests that the term "visually similar" for interior point methods can be related to "relatively close objective function value". This relationship is not found when using MMA or SQP.**

Remark 4 *Enforcing grey designs at the beginning of the optimization process is not suitable for all multiphysics prob-*

---

[3] The KKT condition of IPOPT cannot be obtained with the interface used in these numerical examples, and thus cannot be presented in Figure 9.

(a) Objective function value

(b) Number of intermediate design variables

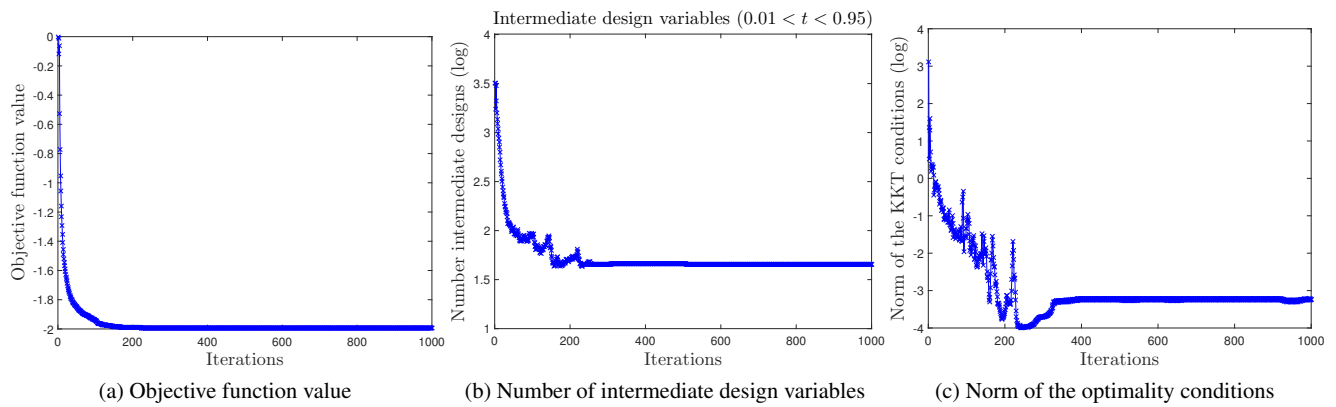(c) Norm of the optimality conditions

Fig. 7: Objective function value at each optimization iteration, number of intermediate design variables (logarithmic scale), and norm of the KKT conditions (logarithmic scale) using MMA as solver.
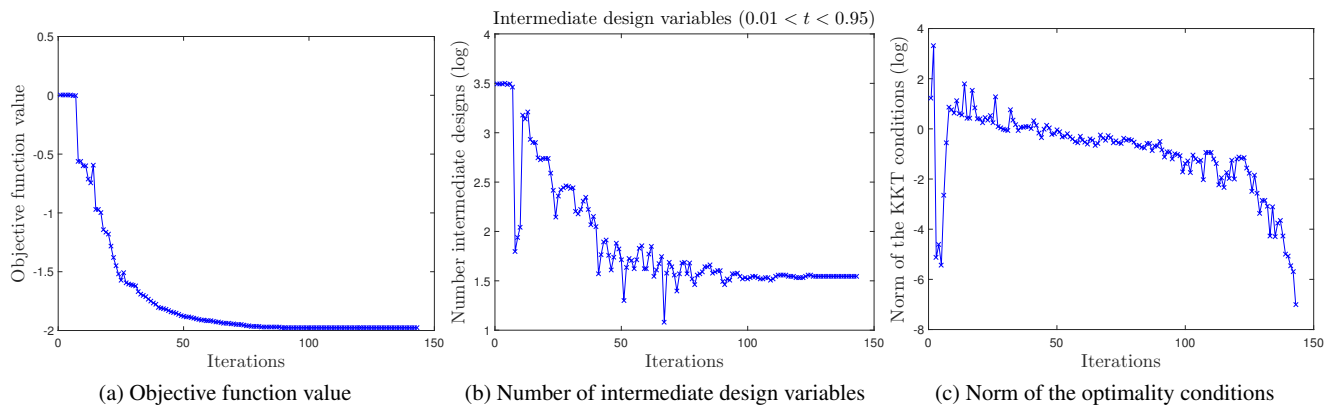

(a) Objective function value

(b) Number of intermediate design variables

(c) Norm of the optimality conditions

Fig. 8: Objective function value at each optimization iteration, number of intermediate design variables (logarithmic scale), and norm of the KKT conditions (logarithmic scale) using SQP as solver.
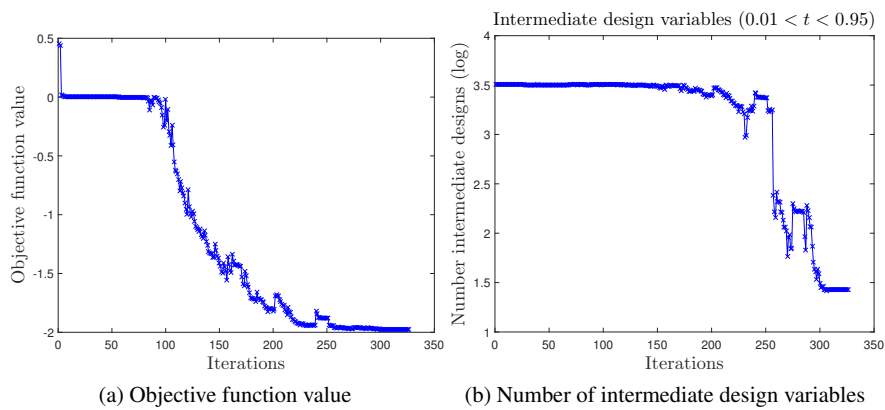

(a) Objective function value

(b) Number of intermediate design variables

Fig. 9: Objective function value at each optimization iteration and number of intermediate design variables (logarithmic scale) using IPOPT$_a$ as solver.

lems. Some physics situation require clearly defined boundaries, such as fluid-structure interaction problems.

5 Closing remarks and future directions

Several observations, interpretations, and open questions can be made based on the presented numerical results. Since the observations are supported by only a handful of numerical experiments they do, of course, not provide conclusive evidence. They do, however, give indications in which areas future theoretical and numerical investigations should be directed.

- It is still unclear how the slowly moving boundaries can be identified during the optimization process and how to properly and theoretically define them.
- The TopSQP method behaves similar to MMA for this type of problems. However, since TopSQP and SQP use efficient second-order approximations, they generally converge using fewer iterations.
- Interior point methods behave (if they are appropriately scaled and with appropriate parameters) different compared to sequential convex approximation methods. At the beginning of the optimization process, relatively large barrier parameter values in interior point methods can turn the initial point to a greyish design. Thus, they perform similarly independently of the starting point.
- Interior point methods (with proper scale and parameter selection) can keep intermediate design values for a longer period than MMA. Thus, they are able to properly identify the location of the structural boundaries, avoiding the expensive operation of moving members once the design is solid and void.
- The performance of interior point methods is little affected by the initial point. A direct consequence is that the solver cannot take advantage of qualified starting guesses.
- In the numerical experiments presented, visually similar designs relate to similar objective function values when IPOPT is used.
- In many practical applications, the optimization solvers are stopped after a relatively small number of iterations. Interior point methods might not produce solid-and-void designs since the barrier penalization parameter values could be relatively large for these intermediate iterations.
- The use of continuation methods to delay the solid-and-void designs in MMA and SQP methods might help to reduce the slowly moving boundaries but it would increase the number of iterations.

The authors of this article hope that the observations presented here can lead the field to further investigate the slowly moving boundaries issue. We hope these results will help and guide the community to answer some research questions such as how to properly define the slowly moving boundaries, how to identify them, and finally how to avoid them.

References

Amestoy PR, Duff IS, L'Excellent JY (2000) Multifrontal parallel distributed symmetric and unsymmetric solvers. Computer Methods in Applied Mechanics and Engineering 184(24):501–520

Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O (2011) Efficient topology optimization in MATLAB using 88 lines of code. Structural and Multidisciplinary Optimization 43(1):1–16

Bendsøe MP (1989) Optimal shape design as a material distribution problem. Structural Optimization 1(4):192–202

Bendsøe MP, Sigmund O (1999) Material interpolation schemes in topology optimization. Archive of Applied Mechanics 69(9–10):635–654

Bendsøe MP, Sigmund O (2003) Topology optimization: Theory, methods and applications. Springer

Boggs PT, Tolle JW (1995) Sequential Quadratic Programming. Acta Numerica 4:1–51

Bourdin B (2001) Filters in topology optimization. International Journal for Numerical Methods in Engineering 50(9):2143–2158

Byrd RH, Hribar ME, Nocedal J (1999) An interior point algorithm for large-scale nonlinear programming. SIAM Journal on Optimization 9(4):877–900

Dennis JE, Moré JJ (1977) Quasi-Newton Methods, Motivation and Theory. SIAM Review 19(1):46–89

Forsgren A, Gill PE (1998) Primal-dual interior methods for nonconvex nonlinear programming. SIAM Journal on Optimization 8(4):1132–1152

Morales JL, Nocedal J, Wu Y (2010) A sequential quadratic programming algorithm with an additional equality constrained phase. Journal of Numerical Analysis 32(2):553–579

Nocedal J (1980) Updating Quasi-Newton matrices with limited storage. Mathematics of Computation 35(151):773–782

Nocedal J, Wright SJ (1999) Numerical Optimization. Springer

Nocedal J, Wächter R, Waltz RA (2009) Adaptive barrier update strategies for nonlinear interior methods. SIAM Journal on Optimization 19(4):1674–1693

(a) Percentage between f^* and f_i (logarithmic scale)   (b) Density difference between an intermediate and the final design
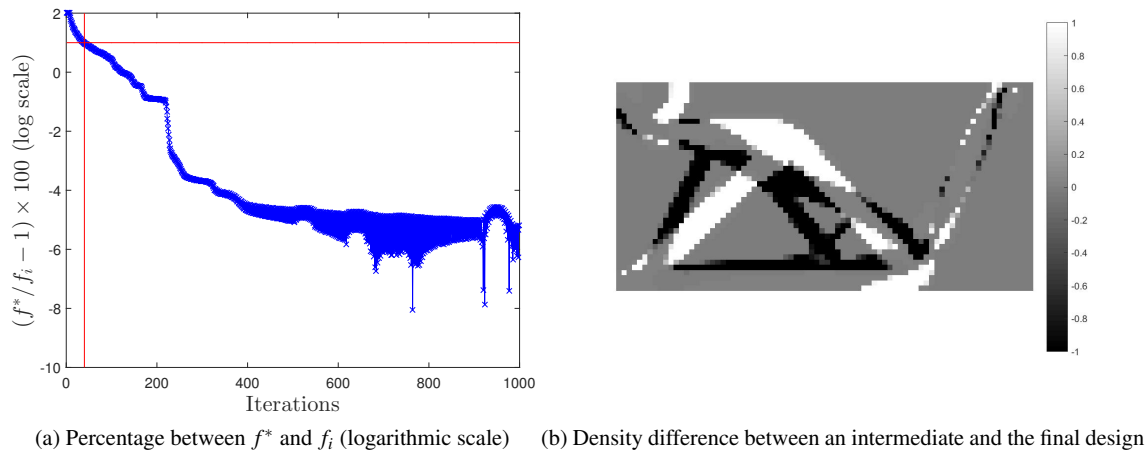
Fig. 10: The left figure presents the percentage between the objective function value at each optimization iteration and minimum objective function value found by the same solver (MMA). The right figure shows the design difference between the final design (at iteration 1000) and an intermediate design (at iteration 40). The difference in the objective function value for these two designs is less than or equal to 10%.

(a) Percentage between f^* and f_i (logarithmic scale)   (b) Density difference between an intermediate and the final design
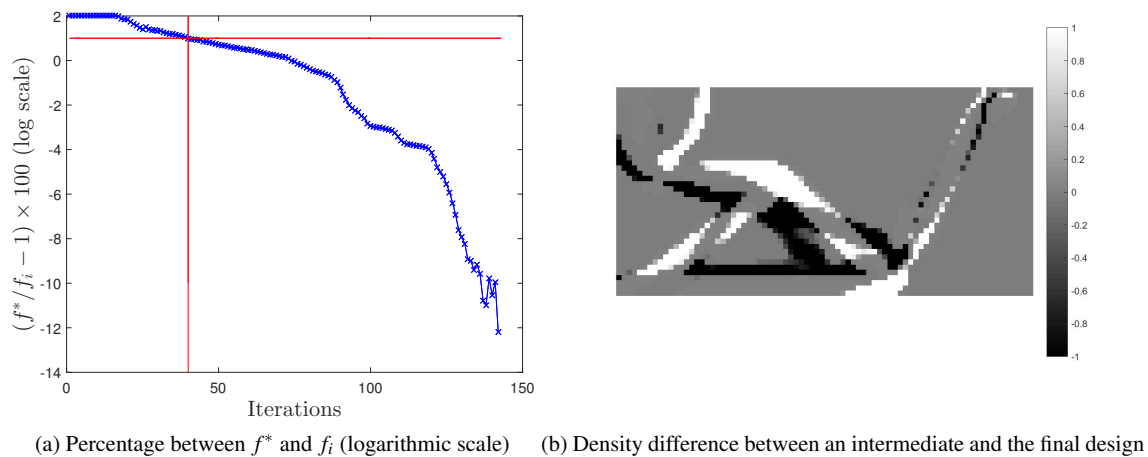
Fig. 11: The left figure presents the percentage between the objective function value at each optimization iteration and minimum objective function value found by the same solver (SQP). Figure 11b shows the design difference between the final design (at iteration 143) and an intermediate design (at iteration 40). The difference in the objective function value for these two designs is less than or equal to 10%.

Rojas-Labanda S, Stolpe M (2015) Benchmarking optimization solvers for structural topology optimization. Structural and Multidisciplinary Optimization 52(3):527–547

Rojas-Labanda S, Stolpe M (2016) An efficient second-order SQP method for structural topology optimization. Structural and Multidisciplinary Optimization 53(6):1315–1333

Rozvany GIN, Zhou M (1991) The COC algorithm, part I: Cross-section optimization or sizing. Computer Methods in Applied Mechanics and Engineering 89(1–3):281–308

Sigmund O (1997) On the design of compliant mechanisms using topology optimization. Journal of Structural Mechanics 25(4):492–526

Sigmund O (2009) Manufacturing tolerant topology optimization. Acta Mechanica Sinica 25(2):227–239

Sigmund O, Maute K (2013) Topology optimization approaches. Structural and Multidisciplinary Optimization 48(6):1031–1055

Svanberg K (1987) The method of moving asymptotes - A new method for structural optimization. Interna-
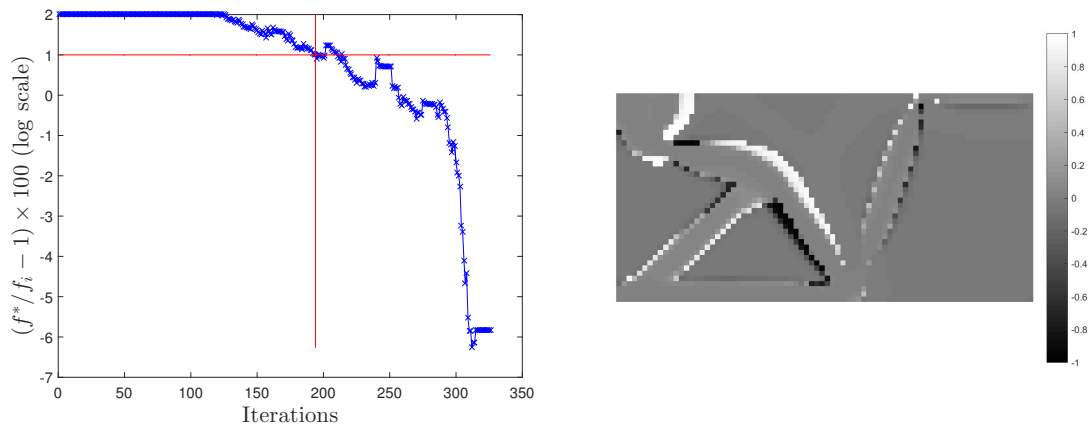
(a) Percentage between f^* and f_i (logarithmic scale)   (b) Density difference between an intermediate and the final design

Fig. 12: The left figure presents the percentage between the objective function value at each optimization iteration and minimum objective function value found by the same solver (IPOPT$_a$). The right figure shows the design difference between the final design (at iteration 325) and an intermediate design (at iteration 194). The difference in the objective function value for these two designs is less than or equal to 10%.

tional Journal for Numerical Methods in Engineering 24(2):359–373

Wächter A, Biegler LT (2006) On the implementation of an interior point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming 106(1):25–57

Yamashita H (1998) A globally convergent primal-dual interior point method for constrained optimization. Optimization Methods and Software 10(2):2–4

Zhou M, Rozvany GIN (1991) The COC algorithm, Part II: Topological, geometrical and generalized shape optimization. Computer Methods in Applied Mechanics and Engineering 89(1–3):309–336