SMO manuscript No.
(will be inserted by the editor)

An SLP algorithm for topology optimization

F. A. M. Gomes · T. A. Senne

Received: date / Accepted: date

Abstract Topology optimization problems, in general, and compliant mechanism design problems, in particular, are engineering applications that rely on nonlinear programming algorithms. Since these problems are usually huge, methods that do not require information about second derivatives are generally used for their solution. The most widely used of such methods are some variants of the method of moving asymptotes (MMA), proposed by Svanberg (1987), and sequential linear programming (SLP).

Although showing a good performance in practice, most of the SLP algorithms used in topology optimization lack a global convergence theory. This paper introduces a globally convergent SLP method for nonlinear programming. The algorithm is applied to the solution of classic compliance minimization problems, as well as to the design of compliant mechanisms. Our numerical results suggest that the new algorithm is faster than the globally convergent version of the MMA method.

Keywords Topology optimization · Compliant mechanisms · Sequential linear programming · Global convergence theory

Mathematics Subject Classification (2000) 65K05 · 90C55

1 Introduction

Topology optimization is a computational method originally developed with the aim of finding the stiffest structure that satisfies certain conditions, such as an upper limit for the amount of material.

This work is work was supported by CNPq and FAPESP (grant 2006/53768-0).

F. A. M. Gomes, T. A. Senne
Departamento de Matemática Aplicada, IMECC, Universidade Estatual de Campinas, 13083-859, Campinas, SP, Brazil.
E-mail: chico@ime.unicamp.br

The structure under consideration is under the action of external forces, and must be contained into a design domain Ω. Once the domain Ω is discretized, to each one of its elements we associate a variable χ that is set to 1 if the element belongs to the structure, or 0 if the element is void. Since it is difficult to solve a large nonlinear problem with discrete variables, χ is replaced by a continuous variable $\rho \in [0,1]$, called the element's "density".

However, in the final structure, ρ is expected to assume only 0 or 1. In order to eliminate the intermediate values of ρ, Bendsøe (1989) introduced the *Solid Isotropic Material with Penalization method* (SIMP for short), which replaces ρ by the function ρ^p that controls the distribution of material. The role of the penalty parameter $p > 1$ is to reduce of the occurrence of intermediate densities.

Topology optimization problems gained attention over the last two decades, due to their applicability in several engineering areas. One of the most successful applications of topology optimization is the design of compliant mechanisms. A compliant mechanism is a structure that is flexible enough to produce a maximum deflection at a certain point and direction, but is also sufficiently stiff as to support a set of external forces. Such mechanisms are used, for example, to build micro-eletrical-mechanical systems (MEMS).

Topology optimization problems are usually converted into nonlinear programming problems. Since the problems are huge, the iterations of the mathematical method used in its solution must be cheap. Therefore, methods that require the computation of second derivatives must be avoided. In this paper, we propose a new sequential linear programming algorithm for solving constrained nonlinear programming problems, and apply this method to the solution of topology optimization problems, including compliant mechanism design.

In the next section, we present the formulation adopted for the basic topology optimization problem, as well as to the compliant mechanism design problem. In Section 3, we introduce a globally convergent sequential linear programming algorithm for nonlinear programming. In Section 4, we discuss how to avoid the presence of checkerboard like material distribution in the structure. We devote Section 5 to our numerical experiments. Finally, Section 6 contains the conclusion and suggestions for future work.

2 Problem formulation

The simplest topology optimization problem is the compliance minimization of a structure (e.g. Bendsøe and Kikuchi 1988). The objective is to find the stiffest structure that fits into the domain, satisfies the boundary conditions and has a prescribed volume. After domain discretization, this problem becomes

$$\begin{aligned} \min_{\rho}\ & \mathbf{f}^T\mathbf{u} \\ \text{s.t.}\ & \mathbf{K}(\rho)\mathbf{u} = \mathbf{f} \\ & \sum_{i=1}^{n_{el}} v_i\rho_i \leq V \\ & \rho_{min} \leq \rho_i \leq 1, \qquad i = 1,\ldots,n_{el}, \end{aligned} \tag{1}$$

where n_{el} is the number of elements of the domain, ρ_i is the density and v_i is the volume of the i-th element, V is the upper limit for the volume of the structure, $\mathbf{f}$ is the vector of nodal forces associated to the external loads and $\mathbf{K}(\rho)$ is the stiffness matrix of the structure.

When the SIMP model is used to avoid intermediate densities, the global stiffness matrix is given by

$$\mathbf{K}(\rho) = \sum_{i=1}^{n_{el}} \rho_i^p \mathbf{K}_i,$$

where $\mathbf{K}_i$ is the stiffness matrix of the i-th element.

The parameter $\rho_{min} > 0$ is used to avoid zero density elements, that would imply in singularity of the stiffness matrix. Thus, for $\rho \geq \rho_{min}$, matrix $\mathbf{K}(\rho)$ is invertible, and it is possible to eliminate the u variables replacing $\mathbf{u} = \mathbf{K}(\rho)^{-1}\mathbf{f}$ in the objective function of problem (1). In this case, the problem reduces to

$$\begin{aligned} \min_{\rho}\ & \mathbf{f}^T\mathbf{K}(\rho)^{-1}\mathbf{f} \\ \text{s.t.}\ & \sum_{i=1}^{n_{el}} v_i\rho_i \leq V \\ & \rho_{min} \leq \rho_i \leq 1, \qquad i = 1,\ldots,n_{el} \end{aligned} \tag{2}$$

This problem has only one linear inequality constraint, besides the box constraints. However, the objective function is nonlinear, and its computation requires the solution of a linear systems of equations.

2.1 Compliant mechanisms

A more complex topology optimization problem is the design of a compliant mechanism. Some interesting formulations for this problem were introduced by Nishiwaki et al. (1998), Kikuchi et al. (1998), Lima (2002), Sigmund (1997), Pedersen et al. (2001), Min and Kim (2004), and Luo et al. (2005), to cite just a few.

No matter the author, each formulation eventually represents the physical structural problem by means of a nonlinear programming problem. The degree of nonlinearity of the objective function and of the problem constrains vary from one formulation to another. Besides, each one has its own idiosyncrasies that should be taken into account in the implementation of a specific algorithm for solving the optimization problem.

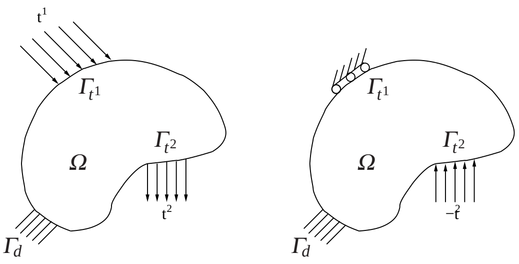
Therefore, an optimization method that works well with one formulation may be inefficient when applied to others. In this work, we adopt the formulation proposed by Nishiwaki et al. (1998), although some encouraging preliminary results were also obtained for the formulations of Sigmund (1997) and Lima (2002).

Nishiwaki et al. (1998) suggest to decouple the problem into two distinct load cases. In the first case, a load $\mathbf{t}^1$ is applied to the region Γ_{t^1} of the boundary of the domain Ω, and a fictitious load $\mathbf{t}^2$ is applied to the region Γ_{t^2} of the boundary of the domain Ω, as shown in Figure 1(a). This second load defines the desired direction of deformation of the Γ_{t^2} region.

To determine the optimal structure for this problem, we should maximize the mutual energy of the mechanism, satisfying the equilibrium and volume constraints. This problem represents the kinematic behavior of the compliant mechanism.

After the mechanism deformation, the Γ_{t^2} region eventually contacts a workpiece. In this case, the mechanism must be sufficiently rigid to resist the reaction force exerted by the workpiece and to keep its shape. This structural behavior of the mechanism is given by the second load case, shown in Figure 1(b). The objective is to minimize the mean compliance, supposing that a load is applied to Γ_{t^2}, and that there is no deflection at the region Γ_{t^1}.

The maximization of the mutual energy and the minimization of the mean compliance are combined into a single optimization problem. In the discretized form, this problem is defined by



(a) (b)

Fig. 1 The two load cases considered in the formulation of Nishiwaki et al. (1998).

$$
\begin{array}{ll}
\min\limits_{\rho} & -\dfrac{\mathbf{f}_b^T \mathbf{u}_a}{\mathbf{f}_c^T \mathbf{u}_c} \\
\text{s.t.} & \mathbf{K}_1(\rho)\mathbf{u}_a = \mathbf{f}_a \\
& \mathbf{K}_1(\rho)\mathbf{u}_b = \mathbf{f}_b \\
& \mathbf{K}_2(\rho)\mathbf{u}_c = -\mathbf{f}_b \\
& \sum\limits_{i=1}^{n_{el}} v_i \rho_i \leq V \\
& \rho_{min} \leq \rho_i \leq 1, \qquad i = 1, \ldots, n_{el}.
\end{array} \tag{3}
$$

In this problem, $\mathbf{f}_a$ and $\mathbf{f}_b$ are the vectors of nodal forces associated to the loads $\mathbf{t}_1$ and $\mathbf{t}_2$, respectively, while $\mathbf{K}_1(\rho)$ and $\mathbf{K}_2(\rho)$ are the stiffness matrices related to the load cases shown in Figure 1. The mutual energy is given by $\mathbf{f}_b^T \mathbf{u}_a$, and $\mathbf{f}_c^T \mathbf{u}_c$ represents the mean compliance that is to be minimized.

Since matrices $\mathbf{K}_1(\rho)$ and $\mathbf{K}_2(\rho)$ are invertible, it is possible to eliminate the $\mathbf{u}$ variables replacing $\mathbf{u}_a = \mathbf{K}_1(\rho)^{-1}\mathbf{f}_a$, $\mathbf{u}_b = \mathbf{K}_1(\rho)^{-1}\mathbf{f}_b$ and $\mathbf{u}_c = -\mathbf{K}_2(\rho)^{-1}\mathbf{f}_c$ in the objective function of (3). The new problem is

$$
\begin{array}{ll}
\min\limits_{\rho} & -\dfrac{\mathbf{f}_b^T \mathbf{K}_1(\rho)^{-1} \mathbf{f}_a}{\mathbf{f}_c^T \mathbf{K}_2(\rho)^{-1} \mathbf{f}_c} \\
\text{s.t.} & \sum\limits_{i=1}^{n_{el}} v_i \rho_i \leq V \\
& \rho_{min} \leq \rho_i \leq 1, \qquad i = 1, \ldots, n_{el}
\end{array} \tag{4}
$$

This problem has the same constraints of (2). However, the objective function is very nonlinear, and its computation requires the solution of two linear systems of equations.

Other formulations, such as the one proposed by Sigmund (1997), also include constraints on the displacements at certain points of the domain, so the optimization problem becomes larger and more nonlinear.

3 Sequential linear programming

Sequential linear programming (SLP) algorithms have been used successfully in structural design (e.g. Kikuchi et al. 1998; Nishiwaki et al. 1998; Lima 2002; Sigmund 1997). This class of methods is well suited for solving large nonlinear problems due to the fact that it does not require the computation of second derivatives, so the iterations are cheap. However, for most algorithms presented in the literature, global convergence results are not fully established.

In this section we describe a new SLP algorithm for the solution of constrained nonlinear programming problems. As it will become clear, our algorithm is not only globally convergent, but can also be easily adapted for solving topology optimization problems.

3.1 Description of the method

Consider the nonlinear programming problem

$$
\begin{array}{ll}
\min & f(\mathbf{x}) \\
\text{s.t.} & \mathbf{c}(\mathbf{x}) = \mathbf{0}, \\
& \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u,
\end{array} \tag{5}
$$

where the functions $f : \mathbb{R}^n \to \mathbb{R}$ and $\mathbf{c}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m$ have Lipschitz continuous first derivatives, and vectors $\mathbf{x}_l,\ \mathbf{x}_u \in \mathbb{R}^n$define the lower and upper bounds for the components of $\mathbf{x} = [x_1 \ldots x_n]^T$.

One should notice that, using slack variables, any nonlinear programming problem may be written in the form (5).

Since f_i and $\mathbf{c}$ have Lipschitz continuous first derivatives, it is possible to define a linear approximation for the objective function and for the equality constraints of (5) in the neighborhood of a point $\mathbf{x} \in \mathbb{R}^n$, so

$$f(\mathbf{x}+\mathbf{s}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{s} \equiv L(\mathbf{x}, \mathbf{s})$$

and

$$\mathbf{c}(\mathbf{x}+\mathbf{s}) \approx \mathbf{c}(\mathbf{x}) + \mathbf{A}(\mathbf{x})\mathbf{s},$$

where $\mathbf{A}(\mathbf{x}) = [\nabla f_1(\mathbf{x}) \ldots \nabla f_m(\mathbf{x})]^T$ is the Jacobian matrix of the constraints. Therefore, given a point $\mathbf{x}$, (5) can be approximated by the linear programming problem

$$
\begin{array}{ll}
\min\limits_{\mathbf{s}} & f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{s} \\
\text{s.t.} & \mathbf{A}(\mathbf{x})\mathbf{s} + \mathbf{c}(\mathbf{x}) = \mathbf{0} \\
& \mathbf{x}_l \leq \mathbf{x} + \mathbf{s} \leq \mathbf{x}_u.
\end{array} \tag{6}
$$

A sequential linear programming (SLP) algorithm is an iterative method that generates and solves a sequence of linear problems in the form (6). At each iteration k of the algorithm, a previously computed point $\mathbf{x}^{(k)}$ is used to generate the linear programming problem. After finding $\mathbf{s}_c$, an approximate solution for (6), the variables of the original problem (5) are updated according to

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}_c. \tag{7}$$

Unfortunately, this scheme has some pitfalls. First, problem (6) may be unlimited even in the case problem (5) has an optimal solution. Besides, the linear functions used to define (6) may be poor approximations of the actual functions f and $\mathbf{c}$ on a point $\mathbf{x}+\mathbf{s}$ that is too far from $\mathbf{x}$. To avoid these difficulties, it is an usual practice to require the step $\mathbf{s}$ to satisfy the following *trust region* constraint

$$\|\mathbf{s}\|_\infty \leq \delta, \tag{8}$$

where $\delta > 0$, the *trust region radius*, is updated at each iteration of the algorithm, to reflect the size of the neighborhood of $\mathbf{x}$ where the linear programming problem is a good approximation of (5).

Including the trust region in (6), we get the problem

$$\begin{array}{ll} \min & \nabla f(\mathbf{x})^T \mathbf{s} \\ \text{s.t.} & \mathbf{A}(\mathbf{x})\mathbf{s} + \mathbf{c}(\mathbf{x}) = \mathbf{0} \\ & \mathbf{s}_l \leq \mathbf{s} \leq \mathbf{s}_u \end{array} \tag{9}$$

where $\mathbf{s}_l = \max\{-\delta, \mathbf{x}-\mathbf{x}_l\}$ and $\mathbf{s}_u = \min\{\delta, \mathbf{x}_u - \mathbf{x}\}$.

However, unless $\mathbf{x}^{(k)}$ satisfies the constraints of (5), it is still possible that problem (9) has no feasible solution. In this case, we need not only to improve $f(\mathbf{x}+\mathbf{s})$, but also to find a point that reduces this infeasibility. This can be done, for example, solving the linear programming problem

$$\begin{array}{ll} \min & M(\mathbf{x},\mathbf{s}) = \frac{1}{2}\,||\mathbf{A}(\mathbf{x})\mathbf{s} + \mathbf{c}(\mathbf{x})||_2^2 \\ \text{s.t.} & \mathbf{s}_n^l \leq \mathbf{s} \leq \mathbf{s}_n^u \\ & \mathbf{s} \in \Omega \end{array} \tag{10}$$

where $\mathbf{s}_n^l = \max\{-0.8\delta, \mathbf{x}-\mathbf{x}_l\}$, $\mathbf{s}_n^u = \min\{0.8\delta, \mathbf{x}_u - \mathbf{x}\}$, and Ω is a set suitably chosen to simplify the search for $\mathbf{s}$. Clearly, $M(\mathbf{x},\mathbf{s})$ is an approximation for the true measure of the infeasibility, given by the function

$$\varphi(\mathbf{x}) = \frac{1}{2}\,||\mathbf{c}(\mathbf{x})||_2^2.$$

After solving (10), $\mathbf{x}$, f and $\mathbf{c}$ are updated, so (9) becomes feasible.

One should notice that the trust region used in (10) is slightly smaller that the region adopted in (9). This trick is used to give (9) a sufficiently large feasible region, so the objective function can be improved. As it will become clear in the next sections, the choice of 0.8 is quite arbitrary. However, we prefer to explicitly define a value for this and other parameters of the algorithm in order to simplify the notation.

Problems (9) and (10) reveal the two conflicting objectives we need to deal with at each iteration of the algorithm: the reduction of $f(\mathbf{x})$ and the reduction of $\varphi(\mathbf{x})$.

If $f(\mathbf{x}^{(k)}+\mathbf{s}_c) << f(\mathbf{x}^{(k)})$ and $\varphi(\mathbf{x}^{(k)}+\mathbf{s}_c) << \varphi(\mathbf{x}^{(k)})$, it is clear that $\mathbf{x}+\mathbf{s}_c$ is a better approximation than $\mathbf{x}^{(k)}$ for the optimal solution of problem (5). However, no straightforward conclusion can be drawn if one of these functions is reduced while the other is increased.

In such situations, we use a *merit function* to decide if $\mathbf{x}^{(k)}$ can be replaced by $\mathbf{x}^{(k)}+\mathbf{s}_c$. In this work, the merit function is defined as

$$\psi(\mathbf{x},\theta) = \theta f(\mathbf{x}) + (1-\theta)\varphi(\mathbf{x}), \tag{11}$$

where $\theta \in (0,1]$ is a penalty parameter used to balance the roles of f and φ. If the merit function is sufficiently reduced between $\mathbf{x}^{(k)}$ and $\mathbf{x}^{(k)}+\mathbf{s}_c$, then the step $\mathbf{s}_c$ is accepted.

However, it is not possible to define a fixed reduction for the merit function. Thus, the step acceptance is based on the comparison of the actual reduction of ψ with the reduction predicted by the linear model used to compute $\mathbf{s}_c$.

The actual reduction of ψ between $\mathbf{x}^{(k)}$ and $\mathbf{x}^{(k)}+\mathbf{s}_c$ is given by

$$A_{red} = \theta A_{red}^{opt} + (1-\theta)A_{red}^{fct},$$

where

$$A_{red}^{opt} = f(\mathbf{x}) - f(\mathbf{x}+\mathbf{s}_c)$$

is the actual reduction of the objective function, and

$$A_{red}^{fct} = \varphi(\mathbf{x}) - \varphi(\mathbf{x}+\mathbf{s}_c)$$

is the reduction of the infeasibility.

The predicted reduction of the merit function is defined as

$$P_{red} = \theta P_{red}^{opt} + (1-\theta)P_{red}^{fct},$$

where

$$P_{red}^{opt} = -\nabla f(\mathbf{x})^T \mathbf{s}_c$$

is the predicted reduction of f and

$$\begin{aligned} P_{red}^{fct} &= M(\mathbf{x},\mathbf{0}) - M(\mathbf{x},\mathbf{s}_c) \\ &= \frac{1}{2}\,||\mathbf{c}(\mathbf{x})||_2^2 - \frac{1}{2}\,||\mathbf{A}(\mathbf{x})\mathbf{s}_c + \mathbf{c}(\mathbf{x})||_2^2 \end{aligned}$$

is the predicted reduction of the infeasibility.

At the k-th iteration of the algorithm, the step $\mathbf{s}_c$ is accepted if the merit function is reduced at least by one tenth of the reduction predicted by the linear model, i.e.

$$A_{red} \geq 0.1 P_{red}.$$

If this condition is not verified, δ is reduced and the step is recomputed. On the other hand, the trust region radius may also be increased if the ratio A_{red}/P_{red} is sufficiently large.

The role of the penalty parameter is crucial for the acceptance of the step. Unfortunately, computing θ is also the trickiest part of the merit function definition. It is easy to see from (11) that it may be necessary to reduce θ along the execution of the algorithm to ensure feasibility. However, if this penalty parameter decays too quickly in the first iterations, the steps may become arbitrarily small.

In this work, we follow a suggestion given by Gomes et al. (1999) and define

$$\theta_k = \min\{\theta_k^{large}, \theta_k^{sup}\}.$$

where

$$\theta_k^{large} = \left[1 + \frac{N}{(k+1)^{1.1}}\right] \min\{1, \theta_0, \ldots, \theta_{k-1}\},$$

$$\begin{aligned}\theta_k^{sup} &= \sup\{\theta \in [0,1] \,|\, P_{red} \geq 0.5 P_{red}^{fct}\} \qquad (12)\\ &= \begin{cases} 0.5\left(\dfrac{P_{red}^{fct}}{P_{red}^{fct} - P_{red}^{opt}}\right), & \text{if } P_{red}^{opt} \leq \dfrac{1}{2} P_{red}^{fct} \\ 1, & \text{otherwise.}\end{cases}\end{aligned}$$

The parameter $N \geq 0$, used to compute θ_k^{large}, can be adjusted to allow a nonmonotone decrease of θ.

3.2 An SLP algorithm for nonlinear programming

Let us define $\theta_0 = \theta_{max} = 1$, and $k = 0$, and suppose that a starting point $\mathbf{x}^{(0)} \in X$ and an initial trust region radius $\delta_0 \geq \delta_{min} > 0$ are available.

A new SLP method for solving problem (5) is given by Algorithm 1, where we denote $\mathbf{A} \equiv \mathbf{A}(\mathbf{x})$ and $\mathbf{c} \equiv \mathbf{c}(\mathbf{x})$.

Steps 2 and 7 of this algorithm were intentionally left unspecified to make it more flexible. In Section 5, we describe a particular implementation of this SLP method for solving the topology optimization problem.

In the next subsections we prove that this algorithm is well defined and converges to the solution of (5) under mild conditions.

3.3 The algorithm is well defined

We say that a point $\mathbf{x} \in \mathbb{R}^n$ is *φ-stationary* if it satisfies the Karush-Kuhn-Tucker (KKT) conditions of the problem

$$\min_{x \in X} \varphi(\mathbf{x})$$

where

$$X = \{\mathbf{x} \in \mathbb{R}^n \,|\, \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u\}.$$

In this section, we show that after repeating the steps of Algorithm 1 a finite number of times, a new iterate $\mathbf{x}^{(k+1)}$ is obtained. In order to prove this well definiteness property, we consider three cases. In Lemma 1, we suppose that $\mathbf{x}^{(k)}$ is not φ-stationary and $\mathbf{s}_n$ could not be found. Lemma 2 deals with the case in which $\mathbf{x}^{(k)}$ is not φ-stationary, but there exists $\mathbf{s}_n$. Finally, in Lemma 3, we suppose that $\mathbf{x}^{(k)}$ is feasible and regular for (5), but does not satisfy the KKT conditions of this problem.

Algorithm 1 General SLP algorithm.

1: **while** a stopping criterion is not satisfied, **do**
2: Try to find a point $\mathbf{s}_n$ that satisfies
$$\mathbf{A}\mathbf{s}_n = -\mathbf{c}, \quad \mathbf{s}_n^l \leq \mathbf{s}_n \leq \mathbf{s}_n^u$$
3: **if** $\mathbf{s}_n$ could not be found, **then**
4: $\mathbf{d}_n \leftarrow -\nabla\varphi(\mathbf{x}^{(k)})$
5: Determine $\bar{\alpha}$, the solution of
$$\begin{aligned}\min\ & M(\mathbf{x}^{(k)}, \alpha\mathbf{d}_n)\\ \text{s.t.}\ & \mathbf{s}_n^l \leq \alpha\mathbf{d}_n \leq \mathbf{s}_n^u\\ & \alpha \geq 0\end{aligned}$$
6: $\mathbf{s}_n^d \leftarrow \bar{\alpha}\mathbf{d}_n$
7: Determine $\mathbf{s}_c$ such that $M(\mathbf{x}^{(k)}, \mathbf{s}_c) \leq M(\mathbf{x}^{(k)}, \mathbf{s}_n^d)$.
8: **else**
9: Starting from $\mathbf{s}_n$, determine $\mathbf{s}_c$, the solution of
$$\begin{aligned}\min\ & \nabla f(\mathbf{x})^T\mathbf{s}\\ \text{s.t.}\ & \mathbf{A}\mathbf{s} = -\mathbf{c}\\ & \mathbf{s}_l \leq \mathbf{s} \leq \mathbf{s}_u\end{aligned}$$
10: **end if**
11: Determine $\theta_k \in [0, \theta_{max}]$
12: **if** $A_{red} \geq 0.1 P_{red}$ **then**
13: $\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} + \mathbf{s}_c$
14: **if** $A_{red} \geq 0.5 P_{red}$, **then**
15: $\delta_{k+1} \leftarrow \min\{2.5\delta_k, \|\mathbf{x}_u - \mathbf{x}_l\|_\infty\}$
16: **else**
17: $\delta_{k+1} \leftarrow \delta_{min}$
18: **end if**
19: $\theta_{max} \leftarrow 1$
20: **else**
21: $\delta_{k+1} \leftarrow 0.25\|\mathbf{s}_c\|_\infty$
22: $\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)}$
23: $\theta_{max} \leftarrow \theta_k$
24: **end if**
25: **end while**

Lemma 1 *Suppose that $\mathbf{x}^{(k)}$ is not φ-stationary and that $\mathbf{s}_n$ could not be found. Then, after a finite number of step rejections, $\mathbf{x}^{(k)} + \mathbf{s}_c$ is accepted.*

Proof If $\mathbf{x}^{(k)}$ is not φ-stationary, then

$$\mathbf{d}_n = -\nabla\varphi(\mathbf{x}^{(k)}) = -\nabla M(\mathbf{x}^{(k)}, \mathbf{0}) = -\mathbf{A}^T\mathbf{c} \neq \mathbf{0}.$$

Besides, defining

$$\tilde{\alpha} = \frac{\mathbf{d}_n^T\mathbf{d}_n}{\mathbf{d}_n^T\mathbf{A}^T\mathbf{A}\mathbf{d}_n},$$

and noticing that $\mathbf{s}_n^l \leq \mathbf{0}$ and $\mathbf{s}_n^u \geq \mathbf{0}$, we get

$$\bar{\alpha} = \min\left\{\tilde{\alpha},\ \min_{\mathbf{d}_{n_i}>0}\left\{\frac{\mathbf{s}_{n_i}^u}{\mathbf{d}_{n_i}}\right\},\ \min_{\mathbf{d}_{n_i}<0}\left\{\frac{\mathbf{s}_{n_i}^l}{\mathbf{d}_{n_i}}\right\}\right\}. \qquad (13)$$

Since $\mathbf{x}^{(k)}$ is not φ-stationary, $\bar{\alpha} = \beta\tilde{\alpha}$, for some $\beta \in (0,1]$. Therefore,

$$\begin{aligned} M(\mathbf{0}) - M(\mathbf{s}_n^d) &= M(\mathbf{0}) - M(\beta\tilde{\alpha}\mathbf{d}_n) \\ &= \beta\tilde{\alpha}\mathbf{d}_n^T\mathbf{d}_n - \frac{\beta^2\tilde{\alpha}^2}{2}\mathbf{d}_n^T\mathbf{A}^T\mathbf{A}\mathbf{d}_n \\ &= \beta\frac{(\mathbf{d}_n^T\mathbf{d}_n)^2}{\mathbf{d}_n^T\mathbf{A}^T\mathbf{A}\mathbf{d}_n} - \frac{\beta^2}{2}\frac{(\mathbf{d}_n^T\mathbf{d}_n)^2}{\mathbf{d}_n^T\mathbf{A}^T\mathbf{A}\mathbf{d}_n} \\ &= \left(\beta - \frac{\beta^2}{2}\right)\tilde{\alpha}\mathbf{d}_n^T\mathbf{d}_n \\ &= \left(1 - \frac{\beta}{2}\right)\bar{\alpha}\|\mathbf{d}_n\|_2^2. \end{aligned}$$

From (13) and the definition of $\mathbf{s}_n^l$ and $\mathbf{s}_n^u$, there exists $\bar{\delta} \in (0, \tilde{\alpha}]$ such that, for all $\delta \in (0, \bar{\delta}]$, we have $\|\bar{\alpha}\mathbf{d}_n\|_\infty \le \delta$, and

$$\bar{\alpha}\|\mathbf{d}_n\|_2^2 \ge \|\bar{\alpha}\mathbf{d}_n\|_\infty\|\mathbf{d}_n\|_2 \ge \|\mathbf{d}_n\|_2\delta.$$

Thus,

$$\begin{aligned} P_{red}^{fct} &= M(\mathbf{0}) - M(\mathbf{s}_c) \\ &\ge M(\mathbf{0}) - M(\mathbf{s}_n^d) \ge \left(1 - \frac{\beta}{2}\right)\|\mathbf{d}_n\|_2\delta. \end{aligned} \tag{14}$$

Now, doing a Taylor expansion, we get

$$\mathbf{c}(\mathbf{x}^{(k)} + \mathbf{s}_c) = \mathbf{c}(\mathbf{x}^{(k)}) + \mathbf{A}(\mathbf{x}^{(k)})\mathbf{s}_c + O(\|\mathbf{s}_c\|_2^2),$$

so

$$\begin{aligned} \varphi(\mathbf{x}^{(k)} + \mathbf{s}_c) &= \frac{1}{2}\|\mathbf{c}(\mathbf{x}^{(k)} + \mathbf{s}_c)\|_2^2 \\ &= \frac{1}{2}\mathbf{c}(\mathbf{x}^{(k)})^T\mathbf{c}(\mathbf{x}^{(k)}) + \mathbf{c}(\mathbf{x}^{(k)})^T\mathbf{A}(\mathbf{x}^{(k)})\mathbf{s}_c \\ &\quad + \frac{1}{2}\mathbf{s}_c^T\mathbf{A}(\mathbf{x}^{(k)})^T\mathbf{A}(\mathbf{x}^{(k)})\mathbf{s}_c + O(\|\mathbf{s}_c\|_2^2) \\ &= M(\mathbf{x}^{(k)}, \mathbf{s}_c) + O(\|\mathbf{s}_c\|_2^2). \end{aligned}$$

Analogously, we have

$$f(\mathbf{x}^{(k)} + \mathbf{s}_c) = L(\mathbf{x}^{(k)}, \mathbf{s}_c) + O(\|\mathbf{s}_c\|_2^2).$$

Therefore, for δ sufficiently small,

$$A_{red}(\delta) = P_{red}(\delta) + O(\delta^2),$$

so

$$\lim_{\delta\to 0}\frac{|A_{red}(\delta) - P_{red}(\delta)|}{\delta} = 0. \tag{15}$$

Our choice of θ ensures that $P_{red} \ge 0.5P_{red}^{fct}$. Thus, from (14), we get

$$P_{red} \ge (2-\beta)\|\mathbf{d}_n\|_2\frac{\delta}{4}.$$

Since $\lim_{\delta\to 0}\beta(\delta) = 0$, we have $\beta < 1$ for δ sufficiently small, so

$$P_{red} \ge \|\mathbf{d}_n\|_2\frac{\delta}{4}. \tag{16}$$

From (15) and (16), we obtain

$$\lim_{\delta\to 0}\left|\frac{A_{red}(\delta)}{P_{red}(\delta)} - 1\right| = 0. \tag{17}$$

Therefore, $A_{red} \ge 0.1P_{red}$ for δ sufficiently small, and the step is accepted. ■

Lemma 2 *Suppose that $\mathbf{x}^{(k)}$ is not φ-stationary and that $\mathbf{s}_n$ satisfies the conditions stated in step 2 of Algorithm 1. Then, after a finite number of step rejections, $\mathbf{x}^{(k)} + \mathbf{s}_c$ is accepted.*

Proof Let $\mathbf{s}_n^m$ be the solution of

$$\begin{aligned} \min\ & \|\mathbf{s}\|_\infty \\ \text{s.t.}\ & \mathbf{A}\mathbf{s} = -\mathbf{c} \\ & \mathbf{s}_n^l \le \mathbf{s} \le \mathbf{s}_n^u \end{aligned}$$

Since $\mathbf{x}^{(k)}$ is not φ-stationary, $\|\mathbf{s}_n^m\|_\infty > 0$. Now, supposing that the step is rejected j times, we get $\delta_{k+j} \le 0.25^j\delta_k$. Thus, after $\left\lceil \log_2\sqrt{0.8\delta_k/\|\mathbf{s}_n^m\|_\infty}\right\rceil$ iterations, $\mathbf{s}_n$ is rejected and Lemma 1 applies. ■

Lemma 3 *Suppose that $\mathbf{x}^{(k)}$ is feasible and regular for (5), but does not satisfy the KKT conditions of this problem. If $\mathbf{s}_n$ always exists, then after a finite number of iterations $\mathbf{x}^{(k)} + \mathbf{s}_c$ is accepted.*

Proof If $\mathbf{x}^{(k)}$ is regular but not stationary for problem (5), then we have $\mathbf{d}_t = P_\Upsilon(-\nabla f(\mathbf{x}^{(k)})) \ne \mathbf{0}$, where P_Υ denotes the orthogonal projection onto the set

$$\Upsilon = \left\{\mathbf{s} \in \mathbb{R}^n \mid \mathbf{A}(\mathbf{x}^{(k)})\mathbf{s} = \mathbf{0},\ \mathbf{s}_n^l \le \mathbf{s} \le \mathbf{s}_n^u\right\}.$$

Let $\bar{\alpha}$ be the solution of the auxiliary problem

$$\begin{aligned} \min\ & \alpha\nabla f(\mathbf{x})^T\mathbf{d}_t \\ \text{s.t.}\ & \alpha\mathbf{d} \in \Upsilon \\ & \alpha > 0. \end{aligned} \tag{18}$$

Since (18) is a linear programming problem, $\bar{\alpha}\mathbf{d}$ belongs to the boundary of Υ. Therefore, if $\delta_k < \min\{\mathbf{x}_u - \mathbf{x}_l\}$, then $\|\bar{\alpha}\mathbf{d}_t\|_\infty = \delta_k$, which means that $\bar{\alpha} = \delta_k/\|\mathbf{d}_t\|_\infty$. Besides, $-\nabla f(\mathbf{x}^{(k)})^T\mathbf{d}_t/\|\mathbf{d}_t\|_\infty > 0$, so we have

$$\begin{aligned} L(\mathbf{x}^{(k)}, \mathbf{0}) - L(\mathbf{x}^{(k)}, \bar{\alpha}\mathbf{d}_t) &= -\bar{\alpha}\nabla f(\mathbf{x}^{(k)})^T\mathbf{d}_t \\ &= -\frac{\delta_k}{\|\mathbf{d}_t\|_\infty}\nabla f(\mathbf{x}^{(k)})^T\mathbf{d}_t \\ &= \eta\,\delta_k. \end{aligned} \tag{19}$$

Combining (19) and the fact that $\mathbf{s}_c$ is the solution of (9), we get

$$\begin{aligned}P_{red}^{opt} &= L(\mathbf{x}^{(k)},\mathbf{0}) - L(\mathbf{x}^{(k)},\mathbf{s}_c)\\ &\geq L(\mathbf{x}^{(k)},\mathbf{0}) - L(\mathbf{x}^{(k)},\bar{\alpha}\mathbf{d}_t) = \eta\,\delta_k.\end{aligned}$$

On the other hand, since $\mathbf{x}^{(k)}$ is feasible,

$$M(\mathbf{x}^{(k)},\mathbf{0}) = M(\mathbf{x}^{(k)},\mathbf{s}) = 0.$$

Thus, $\theta_k = \min\{1, \theta_k^{large}\}$ is not reduced along with δ, and

$$P_{red} = \theta_k P_{red}^{opt} \geq \theta_k \eta\,\delta_k. \tag{20}$$

Since (15) also applies in this case, we can combine it with (20) to obtain (17). Therefore, for δ sufficiently small, $A_{red} \geq 0.1 P_{red}$ and the step is accepted. ■

3.4 Every limit point of $\{\mathbf{x}^{(k)}\}$ is φ-stationary

As we have seen, Algorithm 1 stops when $\mathbf{x}^{(k)}$ is a stationary point for problem (5); or when $\mathbf{x}^{(k)}$ is φ-stationary, but infeasible; or even when $\mathbf{x}^{(k)}$ is feasible but not regular.

Following the steps adopted by Gomes et al. (1999), we will now investigate what happens when Algorithm 1 generates an infinite sequence of iterates. Our objective is to prove that the limit points of this sequence are φ-stationary.

Lemma 4 *Suppose that $\mathbf{x}^* \in X$ is not a φ-stationary point and that Algorithm 1 is applied to $\mathbf{x} \in X$ satisfying $\|\mathbf{x} - \mathbf{x}^*\|_2 \leq \varepsilon_1$. Then there exist ε_1, δ_1', $c_1 > 0$ such that $P_{red} \geq c_1 \min\{\delta, \delta_1'\}$.*

Proof If there exists $\mathbf{s}_n$, then $\theta_k^{sup} = 1$ and $P_{red} \geq \frac{1}{2} P_{red}^{fct} = \frac{1}{2} M(\mathbf{0}) = \frac{1}{2}\varphi(\mathbf{x})$, for all θ_k. Thus, defining $c_1 = \frac{1}{2}\varphi(\mathbf{x})$ and $\delta_1' = 1$, we get the desired result.

If $\mathbf{s}_n$ cannot be found, then the proof of Lemma 3 from the paper by Gomes et al. (1999) applies, replacing the constant 0.9 by 1. In this case, $c_1 = \frac{1}{32}\|\nabla\varphi(\mathbf{x}^*)\|_2$. ■

Lemma 5 *Suppose that $\mathbf{x}^* \in X$ is not φ-stationary and that Algorithm 1 is applied to $\mathbf{x} \in X$ satisfying $\|\mathbf{x} - \mathbf{x}^*\|_2 \leq \varepsilon_2$. Then there exist ε_2, $\delta_2' > 0$ such that, if $\delta \leq \delta_2'$, then $A_{red} \geq 0.1 P_{red}$.*

Proof See Lemma 4 of the paper by Gomes et al. (1999). ■

Lemma 6 *Suppose that $\mathbf{x}^*$ is not φ-stationary and K_1 is an infinite set of indices such that*

$$\lim_{k \in K_1} \mathbf{x}^{(k)} = \mathbf{x}^*.$$

Then $\{\delta_k \,|\, k \in K_1\}$ is bounded away from zero. Moreover, there exists $c_2 > 0$ such that, for $k \in K_1$ sufficiently large, we have $A_{red} \geq c_2$.

Proof See Lemma 5 of the paper by Gomes et al. (1999). ■

Lemma 7 *Suppose that Algorithm 1 generates an infinite sequence $\{\mathbf{x}^{(k)}\}$. Then the sequence θ_k is convergent.*

Proof See Lemma 6 of the paper by Gomes et al. (1999). ■

In order to prove the main theorem of this section, we need an additional compactness hypothesis, trivially verified when dealing with bound constrained problems such as (5).

Hypothesis H1. The sequence $\{\mathbf{x}^{(k)}\}$ generated by Algorithm 1 is bounded.

Theorem 1 *Let $\{\mathbf{x}^{(k)}\}$ be an infinite sequence generated by Algorithm 1. Suppose that H1 holds. Then every limit point of $\{\mathbf{x}^{(k)}\}$ is φ-stationary.*

Proof This result can be easily obtained from the proof of Theorem 1 presented in the paper by Gomes et al. (1999), replacing l_k by $f(x_k)$.

3.5 The algorithm finds a critical point

In this section, we show that there exists a limit point of the sequence of iterates generated by Algorithm 1 that is a stationary point of (5). Most lemmas presented below are based on the following hypothesis.

Hypothesis H2. Let $\widetilde{\mathbf{s}}_n$ be the step generated by Algorithm 1 to reduce the infeasiblity, which means that $\widetilde{\mathbf{s}}_n = \mathbf{s}_n$ if this vector exists, or $\widetilde{\mathbf{s}}_n = \mathbf{s}_c$ (see line 7 of the algorithm) if $\mathbf{s}_n$ could not be found. Then, $\|\widetilde{\mathbf{s}}_n\|_2 \leq O(\|\mathbf{c}(\mathbf{x}^{(k)})\|_2)$.

This hypothesis holds if, for example, $\widetilde{\mathbf{s}}_n$ is obtained as the solution of the linear programming problem

$$\begin{array}{ll}\min & \sum_{i=1}^{m}(z_i^- + z_i^+)\\ \text{s.t.} & \mathbf{A}\mathbf{s}_n - \mathbf{z}^- + \mathbf{z}^+ = -\mathbf{c}\\ & \mathbf{s}_n^l \leq \mathbf{s}_n \leq \mathbf{s}_n^u\\ & \mathbf{z}^-, \mathbf{z}^+ \geq \mathbf{0}\end{array} \tag{21}$$

starting from $\mathbf{s}_n^d$.

Let $\widetilde{\mathbf{s}}_n$ be an optimal basic feasible solution of problem (21). If $\mathbf{z}^- = \mathbf{z}^+ = \mathbf{0}$, then $\widetilde{\mathbf{s}}_n = -\mathbf{B}^{-1}\mathbf{c}$, where $\mathbf{B}$ is a nonsingular matrix formed by a subset of the columns of $\mathbf{A}$. Thus, Hypothesis H2 is trivially satisfied.

On the other hand, if $\mathbf{z}^-$ or $\mathbf{z}^+$ have nonzero components, we may write $\widetilde{\mathbf{s}}_n = \mathbf{B}^{-1}(-\mathbf{c} + \mathbf{z}^- - \mathbf{z}^+)$, so

$$\begin{aligned}\|\widetilde{\mathbf{s}}_n\|_2 &\leq \|\mathbf{B}^{-1}\|_2\,\| - \mathbf{c} + \mathbf{z}^- - \mathbf{z}^+\|_2\\ &\leq \|\mathbf{B}^{-1}\|_2\,(\|\mathbf{c}\|_2 + \|\mathbf{z}^- - \mathbf{z}^+\|_2)\\ &\leq \|\mathbf{B}^{-1}\|_2\,(\|\mathbf{c}\|_2 + \|\mathbf{z}^- - \mathbf{z}^+\|_1).\end{aligned}$$

Noting that $\|\mathbf{z}^- + \mathbf{z}^+\|_1$ is just the objective function of problem (21) and that $\|\mathbf{z}^- - \mathbf{z}^+\|_1 = \|\mathbf{c}\|_1$ if $\mathbf{s}_n = \mathbf{0}$, we may write

$$\begin{aligned}\|\mathbf{s}_n\|_2 &\leq \|\mathbf{B}^{-1}\|_2(\|\mathbf{c}\|_2 + \|\mathbf{c}\|_1)\\ &\leq (1+\sqrt{m})\|\mathbf{B}^{-1}\|_2\|\mathbf{c}\|_2,\end{aligned}$$

so H2 also holds in this case. ■

Lemma 8 *Let* $\mathbf{x}$ *be a point satisfying* $\|\mathbf{c}(\mathbf{x})\|_2 \geq \beta\delta$. *In this case, there exists* $c_0 > 0$ *such that*

$$M(\mathbf{x},\mathbf{0}) - M(\mathbf{x},\widetilde{\mathbf{s}}_n(\mathbf{x},\delta)) \geq c_0\delta\|\mathbf{c}(\mathbf{x})\|_2$$

whenever we perform an iteration of Algorithm 1 starting from $\mathbf{x}$.

Proof If $\widetilde{\mathbf{s}}_n = \mathbf{s}_n$, then $M(\widetilde{\mathbf{s}}_n) = 0$, so

$$M(\mathbf{0}) - M(\widetilde{\mathbf{s}}_n) = \frac{1}{2}\|\mathbf{c}(\mathbf{x})\|_2^2 \geq \frac{1}{2}\beta\delta\|\mathbf{c}(\mathbf{x})\|_2 .$$

Therefore, defining $c_0 = \beta/2$, we obtain the desired result.

On the other hand, if $\mathbf{s}_n$ does not exist and $\widetilde{\mathbf{s}}_n = \mathbf{s}_c$, then

$$M(\mathbf{0}) - M(\widetilde{\mathbf{s}}_n) \geq M(\mathbf{0}) - M(\mathbf{s}_n^d). \tag{22}$$

Let $\widetilde{\alpha}$ be the solution of the unrestricted problem

$$\min M(-\alpha\nabla\varphi(\mathbf{x})).$$

In this case,

$$\widetilde{\alpha} = \frac{\nabla\varphi(\mathbf{x})^T\nabla\varphi(\mathbf{x})}{\nabla\varphi(\mathbf{x})^T\mathbf{A}(\mathbf{x})^T\mathbf{A}(\mathbf{x})\nabla\varphi(\mathbf{x})}. \tag{23}$$

From the definition of $\mathbf{s}_n^d$, we have $\mathbf{s}_n^d = -\gamma\widetilde{\alpha}\nabla\varphi(\mathbf{x})$, where $\gamma \in (0,1]$. Thus, from (22) and (23), we get

$$\begin{aligned}
M(\mathbf{0}) - M(\widetilde{\mathbf{s}}_n) &\geq M(\mathbf{0}) - M(-\gamma\widetilde{\alpha}\nabla\varphi(\mathbf{x})) \\
&= -\frac{1}{2}\gamma^2\widetilde{\alpha}^2\nabla\varphi(\mathbf{x})^T\mathbf{A}(\mathbf{x})^T\mathbf{A}(\mathbf{x})\nabla\varphi(\mathbf{x}) \\
&\quad +\gamma\widetilde{\alpha}\mathbf{c}(\mathbf{x})^T\mathbf{A}(\mathbf{x})\nabla\varphi(\mathbf{x}) \\
&= \gamma\left(1-\frac{\gamma}{2}\right)\frac{(\nabla\varphi(\mathbf{x})^T\nabla\varphi(\mathbf{x}))^2}{\nabla\varphi(\mathbf{x})^T\mathbf{A}(\mathbf{x})^T\mathbf{A}(\mathbf{x})\nabla\varphi(\mathbf{x})} \\
&= \gamma\left(1-\frac{\gamma}{2}\right)\frac{\|\nabla\varphi(\mathbf{x})\|_2^4}{\|\mathbf{A}(\mathbf{x})\nabla\varphi(\mathbf{x})\|_2^2} \\
&\geq \gamma\left(1-\frac{\gamma}{2}\right)\frac{\|\mathbf{A}(\mathbf{x})^T\mathbf{c}(\mathbf{x})\|_2^2}{\|\mathbf{A}(\mathbf{x})\|_2^2}.
\end{aligned}$$

Let σ_1 and σ_m be, respectively, the greatest and smallest singular values of $\mathbf{A}(\mathbf{x})$, and suppose that this matrix has full row rank (the extension to the case $rank(\mathbf{A}(\mathbf{x})) < m$ is straightforward). In this case,

$$\begin{aligned}
M(\mathbf{0}) - M(\widetilde{\mathbf{s}}_n) &\geq \gamma\left(1-\frac{\gamma}{2}\right)\frac{\sigma_m^2}{\sigma_1^2}\|\mathbf{c}\varphi(\mathbf{x})\|_2^2 \\
&\geq \gamma\left(1-\frac{\gamma}{2}\right)\frac{\sigma_m^2}{\sigma_1^2}\beta\delta\|\mathbf{c}(\mathbf{x})\|_2.
\end{aligned}$$

Thus, defining $c_0 = \gamma\left(1-\frac{\gamma}{2}\right)\frac{\sigma_m^2}{\sigma_1^2}\beta$, we prove the lemma. ∎

Lemma 9 *Let* $\{\mathbf{x}^{(k)}\}$ *be an infinite sequence generated by Algorithm 1. Suppose that* $\{\mathbf{x}^{(k)}\}_{k\in K_1}$ *is a subsequence that converges to the feasible and regular point* $\mathbf{x}^*$ *that is not stationary for problem (5). Then, there exist* $c_1,k_1,\delta' > 0$ *such that, for* $\mathbf{x} \in \{\mathbf{x}^{(k)}\,|\,k\in K_1,\,k\geq k_1\}$, *we have*

$$L(\mathbf{x},\mathbf{s}_n) - L(\mathbf{x},\mathbf{s}_c) \geq c_1\min\{\delta,\delta'\}.$$

Proof Since the subsequence $\{\mathbf{x}^{(k)}\}_{k\in K_1}$ converges to a feasible and regular point, there exists $k_0 > 0$ such that, for $\mathbf{x} \in \{\mathbf{x}^{(k)}\,|\,k\in K_1,\,k\geq k_0\}$, step $\mathbf{s}_n$ is defined.

Analogously to what was done in Lemma 3, let us define $\mathbf{d}_t = P_\Gamma(-\nabla f(\mathbf{x}^{(k)}))$, where

$$\Gamma = \{\mathbf{s}\in N(\mathbf{A}(\mathbf{x}))\,|\,\mathbf{x}_l \leq \mathbf{x}+\mathbf{s}_n+\mathbf{s} \leq \mathbf{x}_u\},$$

and $N(\mathbf{A}(\mathbf{x}))$ denotes the null space of $\mathbf{A}(\mathbf{x})$. Let us also denote $\mathbf{s}_t^d$ the solution of

$$\begin{array}{ll}
\min & L(\mathbf{x},\mathbf{s}_n+\mathbf{s}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{s}_n+\mathbf{s}) \\
\text{s.t.} & \mathbf{s} = t\mathbf{d}_t,\;\; t\geq 0 \\
& \|\mathbf{s}_n+\mathbf{s}\|_\infty \leq \delta \\
& \mathbf{x}_l \leq \mathbf{x}+\mathbf{s}_n+\mathbf{s} \leq \mathbf{x}_u
\end{array} \tag{24}$$

After some algebra, we see that $\mathbf{s}_t^d = \widetilde{t}\mathbf{d}_t$ is also the solution of

$$\begin{array}{ll}
\min & (\nabla f(\mathbf{x})^T\mathbf{d}_t)t \\
\text{s.t.} & 0 \leq t \leq \overline{t},
\end{array}$$

where

$$\overline{t} = \min\{1,\Delta_1,\Delta_2\},$$

$$\Delta_1 = \min_{d_{t_i}<0}\left\{\frac{\delta+s_{n_i}}{-d_{t_i}},\frac{x_i+s_{n_i}-x_{l_i}}{-d_{t_i}}\right\},$$

$$\Delta_2 = \min_{d_{t_i}>0}\left\{\frac{\delta-s_{n_i}}{d_{t_i}},\frac{x_{u_i}-x_i-s_{n_i}}{d_{t_i}}\right\}.$$

Now, since (24) is a linear programming problem and $\nabla f(\mathbf{x})^T\mathbf{d}_t < 0$, we conclude that $\widetilde{t} = \overline{t}$. Besides, $t = 1$ satisfies $\mathbf{x}_l \leq \mathbf{x}+\mathbf{s}_n+\mathbf{s} \leq \mathbf{x}_u$, so

$$\overline{t} \geq \min\left\{1,\,\min_{d_{t_i}<0}\left\{\frac{\delta+s_{n_i}}{-d_{t_i}}\right\},\,\min_{d_{t_i}>0}\left\{\frac{\delta-s_{n_i}}{d_{t_i}}\right\}\right\}. \tag{25}$$

Remembering that $\mathbf{s}_c$ is the solution of (9), we obtain

$$L(\mathbf{s}_n) - L(\mathbf{s}_c) \geq L(\mathbf{s}_n) - L(\mathbf{s}_n+\mathbf{s}_t^d) = -\overline{t}\nabla f(\mathbf{x})^T\mathbf{d}_t. \tag{26}$$

Since $P_\Gamma(-\nabla f(\mathbf{x}))$ is a continuous function on $\mathbf{x}$, and $\mathbf{x}^*$ is regular and feasible, there exist $c_1',\; c_2' > 0$ and $k_1 \geq k_0$ such that, for all $\mathbf{x} \in \{\mathbf{x}^{(k)}\,|\,k\in K_1,\,k\geq k_1\}$,

$$\|\mathbf{d}_t\|_\infty \leq c_1' \tag{27}$$

and

$$-\nabla f(\mathbf{x})^T\mathbf{d}_t \geq c_2'. \tag{28}$$

From (25) and the fact that $\|\mathbf{s}_n\|_\infty \leq 0.8\delta_k$, we have that

$$t \geq \min\left\{1,\,\frac{0.2\delta}{\|\mathbf{d}_t\|_\infty}\right\}.$$

Thus, from (27) we obtain

$$t \geq \min\left\{1,\,\frac{0.2\delta}{c_1'}\right\} = \frac{0.2}{c_1'}\min\left\{\frac{c_1'}{0.2},\,\delta\right\}. \tag{29}$$

Combining (26), (27), (28) and (29), we get, for all $x \in \{\mathbf{x}^{(k)} \,|\, k \in K_1, k \geq k_0\}$,

$$L(\mathbf{s}_n) - L(\mathbf{s}_c) \geq \frac{0.2c_2'}{c_1'} \min\left\{\frac{c_1'}{0.2}, \delta\right\}.$$

The desired results is obtained taking $c_1 = \dfrac{0.2c_2'}{c_1'}$ and $\delta' = \dfrac{c_1'}{0.2}$. ■

Lemma 10 *Suppose that H2 holds, as well as the hypotheses of Lemma 9. Then there exist $\beta, c_2, k_2 > 0$ such that, whenever $\mathbf{x} \in \{\mathbf{x}^{(k)} \,|\, k \in K_1, k \geq k_2\}$ and $\|\mathbf{c}(\mathbf{x})\|_2 \leq \beta\delta_k$,*

$$L(\mathbf{x}, \mathbf{0}) - L(\mathbf{x}, \mathbf{s}_c) \geq c_2 \min\{\delta, \delta'\}$$

and

$$\theta^{sup}(\mathbf{x}, \delta) = 1,$$

where θ^{sup} is given by (12) and δ' is defined in Lemma 9.

Proof As in Lemma 9, let us suppose that $\mathbf{s}_n$ is defined. In this case, we have

$$L(\mathbf{0}) - L(\mathbf{s}_c) \geq (L(\mathbf{s}_n) - L(\mathbf{s}_c)) - |L(\mathbf{0}) - L(\mathbf{s}_n)|.$$

Now, from Lemma 9 and Hypothesis H2, we get

$$L(\mathbf{0}) - L(\mathbf{s}_c) \geq c_1 \min\{\delta, \delta'\} - O(\|\mathbf{c}(\mathbf{x})\|_2), \quad (30)$$

for all $\mathbf{x} \in \{\mathbf{x}^{(k)} \,|\, k \in K_1, k \geq k_2\}$.

Thus, choosing β conveniently, we prove the first statement of the Lemma.

To prove the second part of the lemma, we note that, from Hypothesis H2,

$$P_{red}^{fct} = M(\mathbf{0}) - M(\mathbf{s}_c) = M(\mathbf{0}) - M(\widetilde{\mathbf{s}}_n) \leq O(\|\mathbf{c}(\mathbf{x})\|_2).$$

Thus, for $\theta^{sup} = 1$, we have

$$P_{red} - 0.5P_{red}^{fct} = P_{red}^{opt} - 0.5P_{red}^{fct} \geq c_1 \min\{\delta, \delta'\} - O(\|\mathbf{c}(\mathbf{x})\|_2).$$

Therefore, for an appropriate choice of β, we get the desired result. ■

Lemma 11 *Suppose that H1 and H2 hold, as well as the hypotheses of Lemma 9. Then $\lim_{k\to\infty} \theta_k = 0$.*

Proof Suppose, for the purpose of obtaining a contradiction, that the infinite sequence $\{\theta_k\}$ does not converge to 0. Since, from Lemma 7, $\{\theta_k\}$ converges, there must exist $k_3 \geq k_2$ and $\widehat{\theta} > 0$ such that $\theta_k \geq \widehat{\theta}$ for $k \geq k_3$.

Now, suppose that $\mathbf{x} \in \{\mathbf{x}^{(k)} \,|\, k \in K_1, k \geq k_3\}$. Once

$$M(\mathbf{x}, \mathbf{0}) - M(\mathbf{x}, \mathbf{s}_c) \geq 0,$$

we get

$$P_{red} \geq \theta[L(\mathbf{x}, \mathbf{0}) - L(\mathbf{x}, \mathbf{s}_c)].$$

Thus, from (30), we obtain

$$P_{red} \geq \theta c_1 \min\{\delta, \delta'\} - O(\|\mathbf{c}(\mathbf{x})\|_2).$$

Since θ is not increased if the step is rejected, we can say that, while s_c is not accepted,

$$P_{red} \geq \widehat{\theta} c_1 \min\{\delta, \delta'\} - O(\|\mathbf{c}(\mathbf{x})\|_2). \quad (31)$$

On the other hand, using a Taylor expansion and the fact that ∇f and $\mathbf{A}$ are Lipschitz continuous, we obtain

$$|A_{red} - P_{red}| \leq O(\delta^2).$$

Thus, there exists $\widetilde{\delta} \in (0, \delta) \subset (0, \delta_{min})$ such that, if $\delta \in (0, \widetilde{\delta})$ and $\mathbf{x} \in \{\mathbf{x}^{(k)} \,|\, k \in K_1, k \geq k_3\}$,

$$|A_{red} - P_{red}| \leq \widehat{\theta} c_1 \frac{\widetilde{\delta}}{40}.$$

Let us define $k_4 \geq k_3$ such that, for all $\mathbf{x} \in \{\mathbf{x}^{(k)} | k \in K_1, k \geq k_4\}$, the term $O(\|\mathbf{c}(\mathbf{x})\|_2)$ of (31) satisfies

$$O(\|\mathbf{c}(\mathbf{x})\|_2) \leq \widehat{\theta} c_1 \frac{\widetilde{\delta}}{20}.$$

In this case,

$$P_{red} \geq \widehat{\theta} c_1 \min\{\delta, \delta'\} - \widehat{\theta} c_1 \frac{\widetilde{\delta}}{20}.$$

Besides, if $\delta \in [\widetilde{\delta}/10, \widetilde{\delta})$, then

$$P_{red} \geq \widehat{\theta} c_1 \frac{\widetilde{\delta}}{10} - \widehat{\theta} c_1 \frac{\widetilde{\delta}}{20} = \widehat{\theta} c_1 \frac{\widetilde{\delta}}{20}.$$

Therefore, for all $\delta \in [\widetilde{\delta}/10, \widetilde{\delta})$ and all $\mathbf{x} \in \{\mathbf{x}^{(k)} \,|\, k \in K_1, k \geq k_4\}$, we have

$$\frac{|A_{red} - P_{red}|}{P_{red}} \leq 0.5,$$

which implies that, for some $\delta \in [\widetilde{\delta}/10, \widetilde{\delta})$, the step is accepted. Thus, δ_k is bounded away from zero for $k \in K_1$, $k \geq k_4$, so P_{red} is also bounded away from zero.

Since $A_{red} \geq 0.1P_{red}$, the sequence $\{x^{(k)}\}$ is infinite and the sequence $\{\theta_k\}$ is convergent, we conclude that $\psi(\mathbf{x}, \theta)$ is unbouded, which contradicts Hypothesis H1, proving the lemma. ■

Lemma 12 *Suppose that the hypotheses of Lemmas 8 to 10 hold. Then, if $\mathbf{x} \in \{\mathbf{x}^{(k)} \,|\, k \in K_1, k \geq k_2\}$ and $\|\mathbf{c}(\mathbf{x})\|_2 \geq \beta\delta$, the ratio δ/θ^{sup} is uniformly bounded.*

Proof Observe that, when $\theta^{sup} \neq 1$,

$$\theta^{sup} = \frac{P_{red}}{2(P_{red}^{fct} - P_{red}^{opt})} = \frac{M(\mathbf{0}) - M(\mathbf{s}_n)}{2[M(\mathbf{0}) - M(\mathbf{s}_n) - L(\mathbf{0}) + L(\mathbf{s}_c)]}.$$

From Hypothesis H2 and Lemma 8, if $\mathbf{x} \in \{\mathbf{x}^{(k)} \mid k \in K_1, k \geq k_2\}$, we have that

$$\frac{1}{2\theta^{sup}} = 1 + \frac{L(\mathbf{s}_c) - L(\mathbf{s}_n)}{M(\mathbf{0}) - M(\mathbf{s}_n)} + \frac{L(\mathbf{s}_n) - L(\mathbf{0})}{M(\mathbf{0}) - M(\mathbf{s}_n)} \leq 1 + \frac{|L(\mathbf{0}) - L(\mathbf{s}_n)|}{M(\mathbf{0}) - M(\mathbf{s}_n)} \leq 1 + \frac{O(\|\mathbf{c}(\mathbf{x})\|_2)}{c_0\delta\|\mathbf{c}(\mathbf{x})\|_2} \leq 1 + O(1/\delta).$$

Therefore, δ/θ^{sup} is bounded. ■

Lemma 13 *Suppose that the hypotheses of Lemmas 8 to 10 hold. Then there exist $k_5 \geq k_2$, $\widetilde{\theta} \in (0,1]$ such that, if $\mathbf{x} \in \{\mathbf{x}^{(k)} \mid k \in K_1, k \geq k_5\}$, $\|\mathbf{c}(\mathbf{x})\|_2 \geq \beta\delta$ and $\theta \leq \widetilde{\theta}$, then $A_{red} \geq 0.1P_{red}$.*

Proof From the fact that $\nabla f(\mathbf{x})$ is Lipschitz continuous, we may write

$$A_{red} = \theta[f(\mathbf{x}) - f(\mathbf{x}+\mathbf{s}_c)] + (1-\theta)[\varphi(\mathbf{x}) - \varphi(\mathbf{x}+\mathbf{s}_c)] = \theta[L(\mathbf{0}) - L(\mathbf{s}_c) + O(\delta^2)] + (1-\theta)[\|\mathbf{c}(\mathbf{x})\|_2^2 - \|\mathbf{c}(\mathbf{x}+\mathbf{s}_n)\|_2^2]/2.$$

Since $A(\mathbf{x})$ is also Lipschitz continuous, we have

$$\|\mathbf{c}(\mathbf{x})\|_2^2 - \|\mathbf{c}(\mathbf{x}+\mathbf{s}_n)\|_2^2 = -\mathbf{s}_c^T\mathbf{A}(\mathbf{x})^T\mathbf{A}(\mathbf{x})\mathbf{s}_c - 2\mathbf{s}_c^T\mathbf{A}(\mathbf{x})^T\mathbf{c}(\mathbf{x}) + \|\mathbf{c}(\mathbf{x})\|_2 O(\delta^2) + O(\delta^3) = 2[M(\mathbf{0}) - M(\mathbf{s}_c)] + \|\mathbf{c}(\mathbf{x})\|_2 O(\delta^2) + O(\delta^3).$$

Thus,

$$A_{red} = \theta[L(\mathbf{0}) - L(\mathbf{s}_c) + O(\delta^2)] + (1-\theta)[M(\mathbf{0}) - M(\mathbf{s}_c) + \|\mathbf{c}(\mathbf{x})\|_2 O(\delta^2) + O(\delta^3)] = P_{red} + \theta O(\delta^2) + (1-\theta)[\|\mathbf{c}(\mathbf{x})\|_2 O(\delta^2) + O(\delta^3)].$$

Now, supposing that $\|\mathbf{c}(\mathbf{x})\|_2 \geq \beta\delta$, we have

$$|A_{red} - P_{red}| \leq \theta\|\mathbf{c}(\mathbf{x})\|_2 O(\delta) + \|\mathbf{c}(\mathbf{x})\|_2 O(\delta^2). \quad (32)$$

Since our choice of θ ensures that $P_{red} \geq 0.5[M(\mathbf{0}) - M(\mathbf{s}_c)]$, Lemma 8 implies that, for $k \in K_1$ sufficiently large,

$$P_{red} \geq \frac{c_0}{2}\|\mathbf{c}(\mathbf{x})\|_2\delta,$$

so $\delta\|\mathbf{c}(\mathbf{x})\|_2/P_{red}$ is uniformly bounded. Then, dividing both sides of (32) by P_{red}, we get

$$\left|\frac{A_{red}}{P_{red}} - 1\right| \leq O(\theta) + O(\delta) \leq O(\theta) + O(\|\mathbf{c}(\mathbf{x})\|_2/\beta), \quad (33)$$

which yields the desired result. ■

Lemma 14 *Let $\{\mathbf{x}^{(k)}\}$ be an infinite sequence generated by Algorithm 1. Suppose that all of the limit points of $\{\mathbf{x}^{(k)}\}$ are feasible and regular and that Hypotheses H1 and H2 hold. Then, there exists a limit point of $\{\mathbf{x}^{(k)}\}$ that is a stationary point of problem (5).*

Proof See Lemma 13 of the paper by Gomes et al. (1999).

Theorem 2 *Let $\{\mathbf{x}^{(k)}\}$ be an infinite sequence generated by Algorithm 1. Suppose that hypotheses H1 and H2 hold. Then all of the limit points of $\{\mathbf{x}^{(k)}\}$ are φ-stationary. Moreover, if all of these limit points are feasible and regular, there exists a limit point $\mathbf{x}^*$ that is a stationary point of problem (5). In particular, if all of the φ-stationary points are feasible and regular, there exists a subsequence of $\{\mathbf{x}^{(k)}\}$ that converges to feasible and regular stationary point of (5).*

Proof This result follows from Theorem 1 and Lemma 14. ■

4 Filtering

It is well known that the direct application of the SIMP method for solving a topology optimization problem may result in a structure containing a checkerboard-like material distribution (e.g. Díaz and Sigmund 1995). To circumvent this problem, several regularization schemes were proposed. The most commonly used schemes are based on density or sensitivity filters, due to their simplicity and ease of implementation (e.g. Bruns and Tortorelli 2003; Sigmund 1997). However, more elaborate approaches, such as the the Sinh method of Bruns (2005) and the morphology-based filters proposed by Sigmund (2007), are also gaining attention.

In this section, we review some of the filters that can be used in conjunction with our SLP method to solve topology optimization problems.

4.1 Sensitivity filter

Perhaps, the most widely used method for avoiding checkerboard patterns is the sensitivity filter proposed by Sigmund (1997/2001). In this filter, each component $\partial f/\partial\rho_i$ of the gradient of f (the objective function in (5)) is replaced by a weighted mean of the derivatives of f with respect to the densities of the elements that belong to a fixed neighborhood B_i of element i. Mathematically, $\partial f/\partial\rho_i$ is replaced by

$$\widehat{\frac{\partial f}{\partial\rho_i}} = \frac{\sum_{j\in B_i}\widehat{H}_j\,\rho_j\,\frac{\partial f}{\partial\rho_j}}{\rho_i\sum_{j\in B_i}\widehat{H}_j}, \qquad i = 1,\ldots,n_{el}, \quad (34)$$

where

$$\widehat{H}_j = \begin{cases} r_{min} - s_{ij}, & \text{if } j \in B_i, \\ 0, & \text{otherwise,} \end{cases} \quad (35)$$

is a weight factor and s_{ij} is the Euclidean distance between the centroids of the elements i and j.

Although having a good performance in practice, the sensitivity filter has one serious disadvantage: the incompatibility between ∇f and f prevent us from using f to measure the progress of the algorithm used to solve (5). To circumvent this problem, we can replace the objective function f of the topology optimization problem by

$$\widetilde{f}(\rho) = \zeta f(\rho) + \sum_{i=1}^{n_{el}} g_i(\rho^{(k)})\rho_i + \sum_{i=1}^{n_{el}} h_i(\rho^{(k)}) \log(\rho_i), \quad (36)$$

where

$$g_i(\rho^{(k)}) = \left[\frac{\widehat{H}_i}{\sum_{j\in B_i} \widehat{H}_j} - \zeta \right] \frac{\partial f(\rho^{(k)})}{\partial \rho_i}$$

and

$$h_i(\rho^{(k)}) = \frac{\sum_{j\in B_i,\, j\neq i} \widehat{H}_j \rho_j^{(k)} \frac{\partial f(\rho^{(k)})}{\partial \rho_j^{(k)}}}{\sum_{j\in B_i} \widehat{H}_j}.$$

and $\zeta \geq 1$ is a penalty factor used to balance the original function f and the two terms introduced in $\widetilde{f}$.

It must be noticed that, in (36), both $g_i(\rho^{(k)})$ and $h_i(\rho^{(k)})$ are updated only at the beginning of iteration k of the SLP algorithm, so they are treated as constants during the computation of s_c.

It is not difficult to show that, if $\rho = \rho^{(k)}$, then

$$\frac{\partial \widetilde{f}(\rho^{(k)})}{\partial \rho_i} = \frac{\widehat{\partial f}(\rho^{(k)})}{\partial \rho_i}.$$

Besides, if the sequence of iterates $\rho^{(k)}$ converges to ρ^*, the optimal solution of the problem, then

$$\lim_{k\to\infty} \frac{\partial \widetilde{f}(\rho^{(k)})}{\partial \rho_i} = \frac{\widehat{\partial f}(\rho^*)}{\partial \rho_i},\ i = 1, \ldots, n_{el}.$$

We also observe that the term $\sum_{i=1}^{n_{el}} h_i(\rho^{(k)}) \log(\rho_i)$ in (36) pushes ρ_i down to zero, reducing the occurrence of intermediate densities.

4.2 Density filter

Another very simple filter was proposed by Bruns and Tortorelli (2003) and works directly on the densities ρ. For each element i, this filter replaces ρ_i by a weighted mean of the densities of the elements belonging to a neighborhood B_i. The new density is given by

$$\phi_i \equiv \phi_i(\rho) = \sum_{j\in B_i} \frac{\omega_j(s_{ij})}{\omega_i} \rho_j, \quad (37)$$

where

$$\omega_j(s_{ij}) = \begin{cases} \frac{\exp(-s_{ij}^2/2(r/3)^2)}{2\pi(r/3)} & \text{if}\, s_{ij} \leq r, \\ 0 & \text{if}\, s_{ij} > r, \end{cases} \quad (38)$$

s_{ij} is the Euclidean distance between the centroids of elements i and j, and

$$\omega_i = \sum_{j\in B_i} \omega_j(s_{ij}). \quad (39)$$

The filtered densities must be used both in the objective function and in the constraints.

4.3 Morphology-based filters

Sigmund (2007) also introduced a family of filters based on the dilation and the erosion image morphology operators.

The idea behind the dilation operator is to replace the density of an element i by the maximum of the densities of the elements that belong to a neighborhood B_i. To avoid the discontinuities produced by the max function, Sigmund uses a continuous version of the operator, replacing ρ_i by

$$\widetilde{\rho}_i = \frac{1}{\beta} \log \left(\frac{\sum_{j\in B_i} \exp(\beta \rho_j)}{\sum_{j\in B_i} 1} \right), \quad (40)$$

for $i = 1, \ldots, n_{el}$.

The effect of the erosion operator is opposite to the one produced by dilation. In its discrete form, the density ρ_i is replaced by the minimum of the densities of the elements in B_i. Again, to allow the use of this operator in conjunction with an gradient-based optimization algorithm, a continuous version was proposed by Sigmund (2007), so ρ_i is replaced by

$$\overline{\rho}_i = 1 - \frac{1}{\beta} \log \left(\frac{\sum_{j\in B_i} \exp(\beta(1-\rho_j))}{\sum_{j\in B_i} 1} \right), \quad (41)$$

for $i = 1, \ldots, n_{el}$.

It is easy to see that

$$\lim_{\beta\to\infty} \widetilde{\rho}_i = \max_{j\in B_i} \rho_j, \qquad \text{and} \qquad \lim_{\beta\to\infty} \overline{\rho}_i = \min_{j\in B_i} \rho_j.$$

Unfortunately, choosing a large β may result in numerical instabilities. Thus, Sigmund (2007) suggests to start with a small β and increase this parameter gradually.

Sigmund also combines these two operators to generate other filters. The open operator, for example, is obtained applying erosion after dilation, while the close operator is generated using dilation after erosion.

The main inconvenience of these filters is that they turn the volume constraint into a nonlinear inequality constraint.

4.4 Sinh filter

The Sinh method of Bruns (2005) combines the density filter with a new scheme for avoiding intermediate densities, replacing the power function of the SIMP model by the hyperbolic sine function.

In the Sinh method, two density measures are used. The first one, $\eta_1(\rho)$, is employed in the objective function of the topological optimization problem, while the second, $\eta_2(\rho)$, replaces the true density in the constraints.

Bruns (2005) has proposed several definitions for $\eta_1(\rho)$ and $\eta_2(\rho)$. The basic Sinh method is obtained combining

$$\eta_{1_i}(\rho) = \rho_i, \ \ i = 1,\ldots,n_{el}, \tag{42}$$

and

$$\eta_{2_i}(\rho) = 1 - \frac{\sinh\{p[1-\phi_i(\rho)]\}}{\sinh(p)}, \ \ i = 1,\ldots,n_{el}, \tag{43}$$

where $\phi_i(\rho)$ is computed according to (37)–(39), and p is a penalty factor.

One disadvantage of this approach is that, due to the presence of the sinh function in (43), the volume constraint becomes nonlinear.

5 Computational results

In this section, we present one possible implementation for our SLP algorithm, and discuss its numerical behavior when applied to the solution of some standard topology optimization problems.

5.1 Algorithm details

Steps 2, 5, 7 and 9 constitute the core of the SLP algorithm. The implementation of the remaining steps is straightforward.

Step 5 is just a one-dimensional quadratic convex optimization problem. The solution of this problem is given by

$$\bar{\alpha} = \min\left\{ -\frac{\mathbf{c}^T\mathbf{A}\mathbf{d}_n}{\mathbf{d}_n^T\mathbf{A}^T\mathbf{A}\mathbf{d}_n}, \min_{d_{n_i}>0}\left\{\frac{s^u_{n_i}}{d_{n_i}}\right\}, \max_{d_{n_i}<0}\left\{\frac{s^l_{n_i}}{d_{n_i}}\right\}\right\}.$$

Step 2 of the SLP algorithm corresponds to the standard phase 1 of the two-phase method for linear programming.

If a simplex based linear programming function is available, then s_n may be defined as the feasible solution obtained at the end of phase 1, supposing that the algorithm succeeds in finding such a feasible solution. In this case, we can proceed to the second phase of the simplex method and solve the linear programming problem stated at Step 9[1].

On the other hand, when the constraints given in Step 2 are incompatible, we need to compute a point s_c satisfying $M(\mathbf{x}^{(k)}, \mathbf{s}_c) \le M(\mathbf{x}^{(k)}, \mathbf{s}_n^d)$ at Step 7. If the solution obtained by the simplex algorithm at the end of phase 1 satisfies this condition, it can be defined as $\mathbf{s}_c$. Otherwise, we can simply set $\mathbf{s}_c = \mathbf{s}_n^d$. Therefore, if the two-phase simplex method is used, the computation effort spent at each iteration corresponds to the solution of a single linear programming problem.

If an interior point method is used as the linear programming solver instead, then some care must be taken to avoid solving two linear problems per iteration. A good alternative is to try to compute Step 9 directly. In case the algorithm fails to obtain a feasible solution, then Steps 5 and 7 need to be performed. Fortunately, in the solution of topology optimization, the feasible region of (9) is usually not empty, so this scheme performs well in practice.

5.2 Description of the tests

In order to confirm the efficiency and robustness of the new algorithm, we compare it to the globally convergent version of the Method of Moving Asymptotes, the so called Conservative Convex Separable Approximations algorithm (CCSA for short), proposed by Svanberg (2002).

We solve four topology optimization problems. The first two are compliance minimization problems easily found in the literature: the cantilever and the MBB beams. The last two are compliant mechanism design problems: the gripper and the force inverter. All of them were discretized into 4-node rectangular finite elements, using bilinear interpolating functions to approximate the displacements.

In our experiments with compliant mechanisms, we use the Nishiwaki et al. (1998) formulation mentioned in section 2. Some preliminary results with the formulations of Lima (2002) and Sigmund (1997) gave similar results.

We also analyze the effect of the application of the filters presented in Section 4, to reduce the formation of checkerboard patterns in the structures.

The SIMP strategy was used in combination with the sensitivity, the density, the dilation and the erosion filters. In all cases, the penalty parameter p was set to 1, 2 and 3, consecutively. For the sinh method, the parameter p given in (43) was set to 1 to 6, consecutively.

The constant $\zeta = 100$ was used in (36) to define the objective function when the sensitivity filter is adopted. For the dilation and erosion filters, we apply $\beta = 0.2$, 0.4, 0.8 and 1.6, consecutively, for each value of p (see equations (40) and (41)).

[1] One should note, however, that the bounds on the variables defined at Steps 2 and 9 are not the same. Thus, some control over the simplex routine is necessary to ensure that not only the objective function, but also the upper and lower bounds on the variables are changed between phases.

When the SIMP method is used and $p = 1$ or 2, the algorithm stops if Δf, the difference between the objective function of two consecutive iterations, falls below 10^{-3}. For $p = 3$, the algorithm is halted if $\Delta f < 10^{-3}$ for three successive iterations. For the sinh method, we stop the algorithm whenever Δf falls below 10^{-3} if $p = 1$, 2 or 3, and require that $\Delta f < 10^{-3}$ for three successive iterations if $p = 4$, 5 or 6. Besides, we also define a limit of 500 iterations for each value of the penalty parameter p, that is used by both the SIMP and the sinh methods.

All of the tests were performed on a personal computer, with an Intel Pentium D 935 processor (3.2GHz, 512 MB RAM), under the Windows XP operating system. The algorithms were implemented in Matlab.

5.3 Cantilever beam design

The first problem we consider is the cantilever beam presented in Fig. 2.

We suppose that the structure's thickness is $e = 1cm$, the Poisson's coefficient is $\sigma = 0.3$ and the Young's modulus of the material is $E = 1N/cm^2$. The volume of the optimal structure is limited by 40% of the design domain. A force $f = 1N$ is applied downwards in the center of the right edge of the beam.

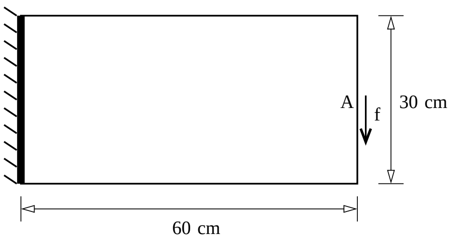


Fig. 2 Design domain for the cantilever beam.

The domain was discretized into 1800 square elements with $1\,mm^2$ each. The optimal structures for all of the combinations of methods and filters are shown in Figure 3.

Table 1 contains the initial trust region radius (δ_0) used to solve this problem, as well as the numerical results obtained, including the optimal value of the objective function, the total number of iterations and the execution time. In this table, the rows labeled *Ratio* contain the ratio between the values obtained by the SLP and the CCSA algorithms. A ratio marked in bold indicates the superiority of SLP over CCSA. The radius of each filter, r_{min}, is given in parentheses, after the filter's name.

The cantilever beams shown in Figure 3 are quite similar, suggesting that all of the filters efficiently reduced the formation of checkerboard patterns, as expected.

Table 1 Results for the cantilever beam

Method	δ_0	Objective	Iterations	Time (s)
no filter				
SLP	0.10	70.3013	298	109.27
CCSA	0.15	71.8734	521	866.65
Ratio	-	**0.978**	**0.572**	**0.126**
Sensitivity filter ($r_{min} = 1.5$)				
SLP	0.05	179.5024	105	49.41
CCSA	0.15	178.5657	352	546.31
Ratio	-	1.005	**0.298**	**0.090**
Density filter ($r_{min} = 2.0$)				
SLP	0.05	81.6859	381	180.80
CCSA	0.15	81.6914	947	2171.00
Ratio	-	1.000	**0.402**	**0.083**
Dilation filter ($r_{min} = 1.0$)				
SLP	0.10	87.6386	1058	691.20
CCSA	0.05	87.7092	1500	8533.30
Ratio	-	**0.999**	**0.705**	**0.081**
Erosion filter ($r_{min} = 1.0$)				
SLP	0.10	85.2952	953	557.08
CCSA	0.05	85.5921	1416	2168.50
Ratio	-	**0.997**	**0.673**	**0.257**
Sinh filter ($r_{min} = 2.0$)				
SLP	0.05	96.0394	818	467.96
CCSA	0.15	96.0574	2216	6019.20
Ratio	-	1.000	**0.369**	**0.078**

The results presented in Table 1 show a clear superiority of the SLP algorithm. Although both methods succeeded in obtaining the optimal structure with all of the filters (with minor differences in the objective function values), the CCSA algorithm spent much more time and took more iterations to converge.

5.4 MBB beam design

The second problem we consider is the MBB beam presented in Fig. 4. The structure's thickness, the Young's modulus of the material and the Poisson's coefficient are the same used for the cantilever beam. The volume of the optimal structure is limited by 50% of the design domain. A force $f = 1N$ is applied downwards in the center of the top edge of the beam.

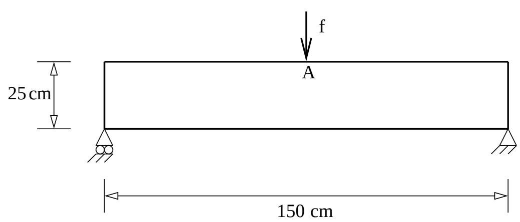


Fig. 4 Design domain for the MBB beam.

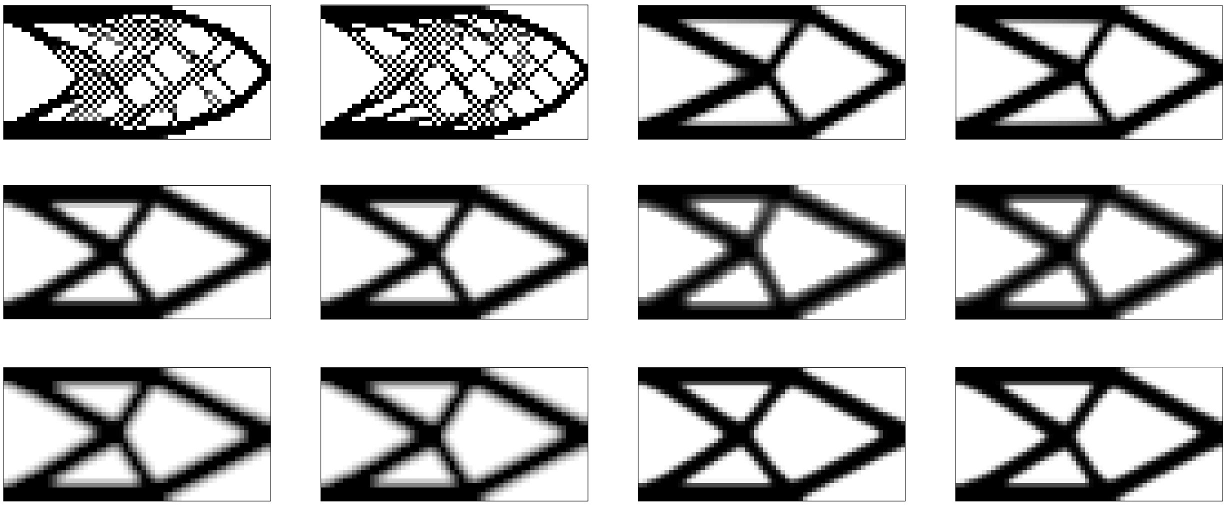


Fig. 3 The cantilever beams obtained using various filter and method combinations. The odd columns present the topologies generated by the SLP method, while the even columns present the topologies found by CCSA. The two columns on the left were obtained using no filter, the density, and the erosion filters. The last two columns present the mechanisms obtained using the sensitivity, the dilation, and the sinh filters

The domain was discretized into 3750 square elements with $1\,mm^2$ each. The optimal structures for all of the combinations of methods and filters are shown in Figure 5. Due to symmetry, only the right half of the domain is shown. Table 2 contains the numerical results obtained for this problem.

Table 2 Results for the MBB beam

Method	δ_0	Objective	Iterations	Time (s)
no filter				
SLP	0.05	166.6435	313	107.36
CCSA	0.15	166.8490	362	602.39
Ratio	-	**0.999**	**0.865**	**0.178**
Sensitivity filter ($r_{min} = 4.0$)				
SLP	0.05	980.2053	76	45.39
CCSA	0.05	958.5473	573	968.66
Ratio	-	1.023	**0.133**	**0.047**
Density filter ($r_{min} = 5.0$)				
SLP	0.05	236.2687	921	1046.00
CCSA	0.10	236.2687	1500	5339.40
Ratio	-	1.000	**0.614**	**0.196**
Dilation filter ($r_{min} = 2.0$)				
SLP	0.10	216.7414	1293	1094.20
CCSA	0.05	226.8034	1500	8394.70
Ratio	-	**0.956**	**0.862**	**0.130**
Erosion filter ($r_{min} = 2.0$)				
SLP	0.10	219.2267	1348	971.11
CCSA	0.05	219.5075	1500	2344.60
Ratio	-	**0.999**	**0.899**	**0.414**
Sinh filter ($r_{min} = 3.0$)				
SLP	0.05	240.3675	1014	673.23
CCSA	0.15	229.2998	2688	7043.90
Ratio	-	1.048	**0.377**	**0.096**

Again, the structures obtained by both methods are similar. The same happens to the values of the objective function, as expected. Table 2 shows that the SLP algorithm performs much better than the CCSA method for the MBB beam. In fact, the CCSA algorithm fails to converge in 1500 iterations for three filters (although the solutions found in these cases are equivalent to those obtained by the SLP method).

5.5 Gripper mechanism design

Our third problem is the design of a gripper, whose domain is presented in Fig. 6. In this compliant mechanism, a force f_a is applied to the center of the left side of the domain, and the objective is to generate a pair of forces with magnitude f_b at the right side. For this problem, we consider that the structure's thickness is $e = 1\,mm$, the Young's modulus of the material is $E = 210000\,N/mm^2$ and the Poisson's coefficient is $\sigma = 0.3$. The volume of the optimal structure is limited by 20% of the design domain. The input and output forces are $f_a = f_b = 1\,N$. The domain was discretized into 3300 square elements with $1\,mm^2$.

Figure 7 shows the grippers obtained. Due to symmetry, only the upper half of the domain is shown. Table 3 summarizes the numerical results.

The grippers shown in Figure 7 and the results presented in Table 3 suggest that there exists a strong correlation between the length of the vertical bar at the left of the gripper's mouth (the vertical bar at the bottom right part of the figure) and the quality of the solution. In fact, the longer the bar, the better is the objective function value found. Unfortunately, it seems that each type of structure obtained corresponds to a local minimum of the nonlinear programming problem, so

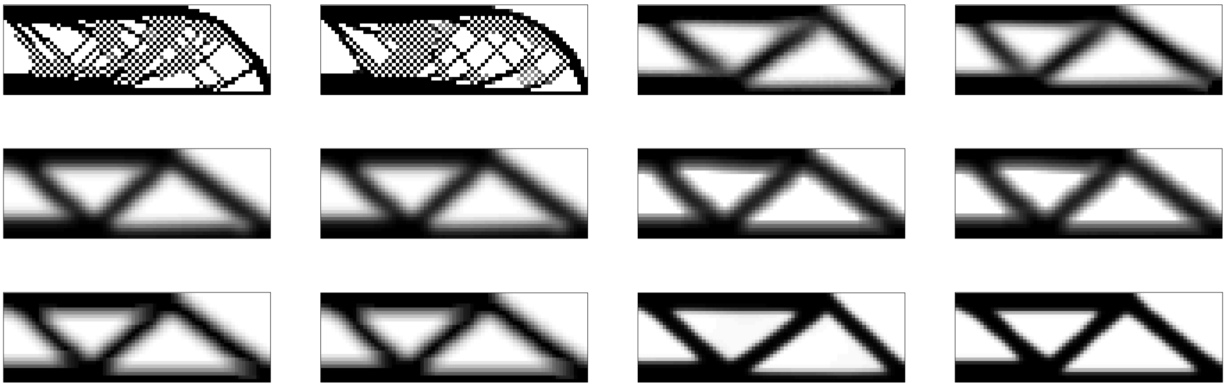


Fig. 5 The MBB beams obtained using various filter and method combinations. The odd columns present the topologies generated by the SLP method, while the even columns present the topologies found by CCSA. The two columns on the left were obtained using no filter, the density, and the erosion filters. The last two columns present the mechanisms obtained using the sensitivity, the dilation, and the sinh filters

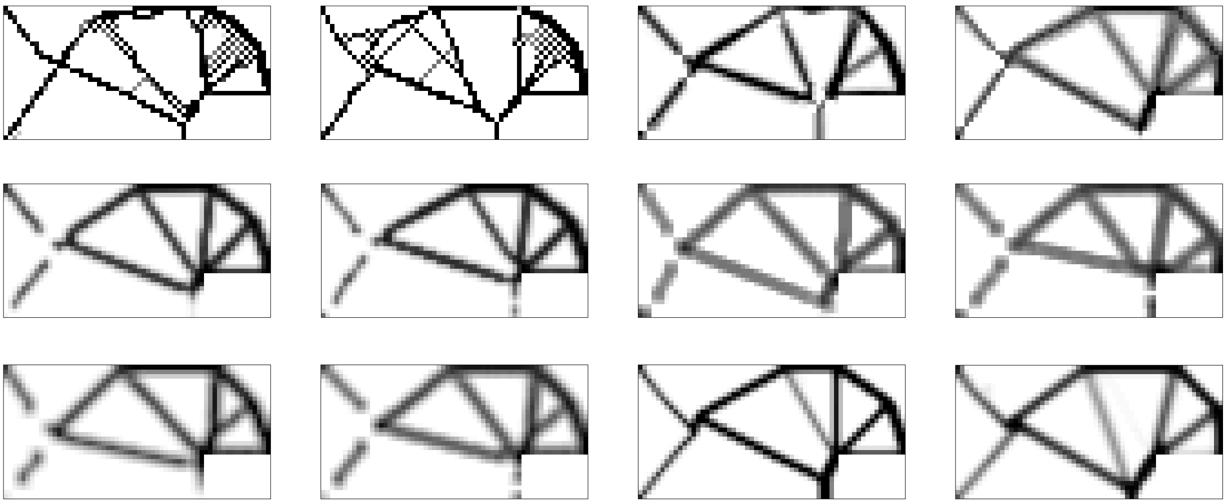


Fig. 7 Grippers obtained using various filter and method combinations. The odd columns present the topologies generated by the SLP method, while the even columns present the topologies found by CCSA. The two columns on the left were obtained using no filter, the density, and the erosion filters. The last two columns present the mechanisms obtained using the sensitivity, the dilation, and the sinh filters

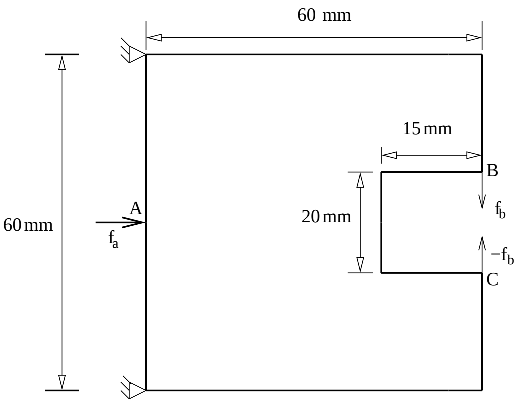


Fig. 6 Design domain for the gripper.

in some cases the algorithms are attracted to points that satisfy the KKT conditions but are not global minima.

Although the SLP algorithm has obtained the best solution for only three of the filters, it spent much less time to obtain the optimal solution in all cases. In fact, the SLP routine always took less than 1/5 of the time spent by the CCSA method.

5.6 Force inverter design

Our last problem is the design of a compliant mechanism known as force inverter. The domain is shown in Fig. 8. In this example, an input force f_a is applied to the center of the left side of the domain and the mechanism should generate

Table 3 Results for the gripper mechanism

Method	δ_0	Objective	Iterations	Time (s)
no filter				
SLP	0.10	-4.6685×10^6	141	98.32
CCSA	0.05	-2.2525×10^6	703	2679.70
Ratio	-	**2.073**	**0.201**	**0.037**
Sensitivity filter ($r_{min} = 1.5$)				
SLP	0.05	-7.3034×10^5	52	43.93
CCSA	0.05	-4.1948×10^5	1044	7067.50
Ratio	-	**1.741**	**0.050**	**0.006**
Density filter ($r_{min} = 2.0$)				
SLP	0.20	-5.7707×10^2	683	459.72
CCSA	0.15	-9.0467×10^2	1444	5943.40
Ratio	-	0.638	**0.473**	**0.077**
Dilation filter ($r_{min} = 1.0$)				
SLP	0.15	-1.3457×10^3	1164	933.41
CCSA	0.05	-1.6534×10^3	1500	6258.20
Ratio	-	0.814	**0.776**	**0.149**
Erosion filter ($r_{min} = 1.0$)				
SLP	0.25	-1.4203×10^3	1328	1058.70
CCSA	0.05	-2.6901×10^3	1500	4837.40
Ratio	-	0.528	**0.885**	**0.219**
Sinh filter ($r_{min} = 1.5$)				
SLP	0.10	-4.2026×10^0	614	382.51
CCSA	0.10	-3.2741×10^0	2389	8224.40
Ratio	-	**1.284**	**0.257**	**0.047**

an output force f_b on the right side of the structure. Note that both f_a and f_b are horizontal, but have opposite directions.

For this problem, we also use $e = 1\,mm$, $\sigma = 0.3$ and $E = 210000\,N/mm^2$. The volume is limited by 20% of the design domain, and the input and output forces are given by $f_a = f_b = 1\,N$. The domain was discretized into 3600 square elements with $1\,mm^2$.

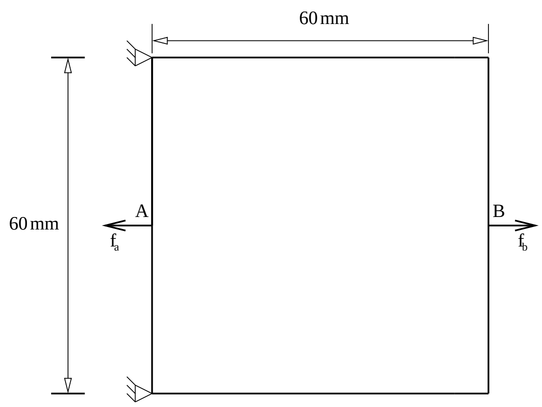


Fig. 8 Design domain for the force inverter.

Figure 9 shows the mechanisms obtained. Again, only the upper half of the structure is shown, due to its symmetry. Table 4 contains the numerical results.

Table 4 Results for the force inverter

Method	δ_0	Objective	Iterations	Time (s)
no filter				
SLP	0.05	-4.8722×10^6	164	93.02
CCSA	0.10	-4.1017×10^6	334	773.86
Ratio	-	**1.188**	**0.491**	**0.120**
Sensitivity filter ($r_{min} = 1.5$)				
SLP	0.20	-1.2081×10^8	51	47.32
CCSA	0.15	-5.6876×10^8	298	1030.40
Ratio	-	0.212	**0.171**	**0.046**
Density filter ($r_{min} = 3.0$)				
SLP	0.05	-8.6923×10^1	618	638.91
CCSA	0.10	-7.6925×10^1	1205	4372.00
Ratio	-	**1.130**	**0.513**	**0.146**
Dilation filter ($r_{min} = 1.0$)				
SLP	0.10	-2.3795×10^5	918	845.14
CCSA	0.20	-2.2690×10^5	1463	6160.30
Ratio	-	**1.049**	**0.627**	**0.137**
Erosion filter ($r_{min} = 1.0$)				
SLP	0.05	-4.1110×10^3	902	840.34
CCSA	0.10	-4.0075×10^5	1424	4517.10
Ratio	-	0.010	**0.633**	**0.186**
Sinh filter ($r_{min} = 1.5$)				
SLP	0.10	-4.7174×10^0	663	517.81
CCSA	0.05	-4.7698×10^0	534	1103.30
Ratio	-	0.989	1.242	**0.469**

According to Table 4, both algorithms found the best solution for exactly three types of filter. However, the CCSA method attained a much better solution for the erosion filter. Curiously, the structures obtained by the algorithms for this filter are fairly similar and do not reflect the difference in the objective function.

As in the previous examples, the SLP method took much less time to converge than the CCSA algorithm.

6 Conclusions and future work

In this paper, we have presented a new globally convergent SLP method. Our algorithm was used to solve some standard topology optimization problems. The results obtained show that it is fast and reliable, and can be used in combination with several filters for removing checkerboards.

The new algorithm seems to be faster than the globally convergent version of the MMA method, while the structures obtained by both methods seem to be comparable.

As we can observe, the filters have avoided the occurrence of checkerboards. However, some of them allowed

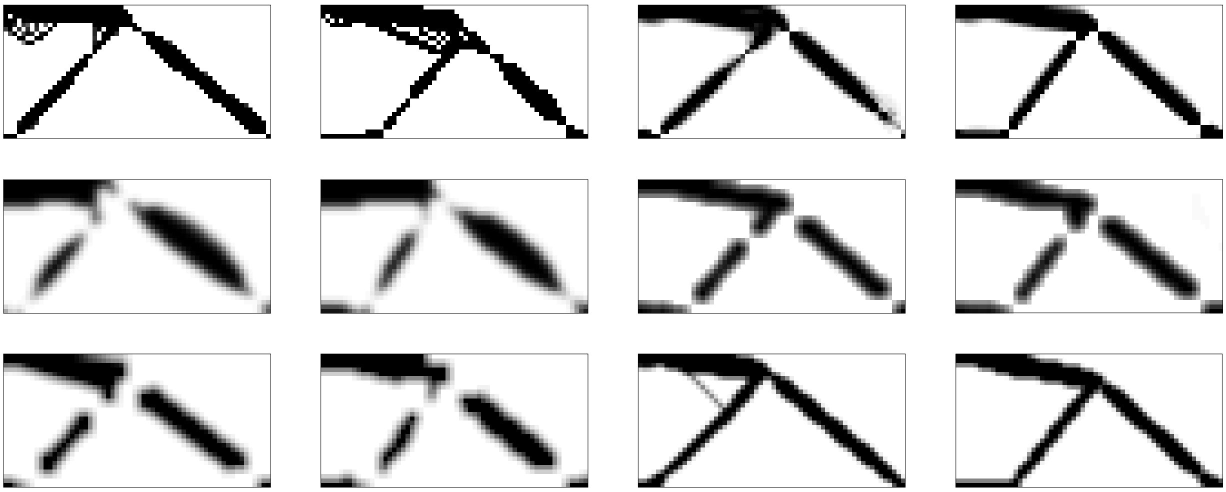


Fig. 9 Force inverters obtained using various filter and method combinations. The odd columns present the topologies generated by the SLP algorithm, while the even columns present the topologies found by CCSA. The two columns on the left were obtained using no filter, the density, and the erosion filters. The last two columns present the mechanisms obtained using the sensitivity, the dilation, and the sinh filters

the formation of one node hinges. The implementation of hinge elimination strategies, following the suggestions of Silva (2007), is one possible extension of this work.

We also plan to analyze the behavior of the SLP algorithm in combination with other compliant mechanism formulations, such as those proposed by Pedersen et al. (2001), Min and Kim (2004), and Luo et al. (2005).

Acknowledgements We would like to thank Prof. Svanberg for supplying the source code of his algorithm and Talita for revising the manuscript.

References

1. Bendsøe MP (1989) Optimal shape design as a material distribution problem. Struct Optim 1: 193-202
2. Bendsøe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. Comput Methods Appl Mech Eng 71: 197-224
3. Bruns TE, Tortorelli DA (2003) An element removal and reintroduction strategy for the topology optimization of structures and compliant mechanisms. Int J Numer Methods Eng 57: 1413-1430
4. Bruns TE (2005) A reevaluation of the SIMP method with filtering and an alternative formulation for solid-void topology optimization. Struct Multidiscipl Optim 30: 428-436
5. Díaz AR, Sigmund O (1995) Checkerboard patterns in layout optimization. Struct and Multidiscipl Optim 10: 40-45
6. Gomes FAM, Maciel MC, Martínez JM (1999) Nonlinear programming algorithms using trust regions and augmented Lagrangians with nonmonotone penalty parameters. Math Program 84: 161-200
7. Kikuchi N, Nishiwaki S, Ono JSF, Silva ECS (1998) Design optimization method for compliant mechanisms and material microstructure. Comput Methods Appl Mech Eng 151: 401-417
8. Lima CR (2002) Design of compliant mechanisms using the topology optimization method (in portuguese). Dissertation, University of São Paulo
9. Luo Z, Chen L, Yang J, Zhang Y, Abdel-Malek, K (2005) Compliant mechanism design using multi-objective topology optimization scheme of continuum structures. Struct Multidiscipl Optim 30: 142-154
10. Min S, Kim Y (2004) Topology Optimization of Compliant Mechanism with Geometrical Advantage. JSME Int J Ser C 47(2): 610-615
11. Nishiwaki S, Frecker MI, Seungjae M, Kikuchi N (1998) Topology optimization of compliant mechanisms using the homogenization method. Int J Numer Methods Eng 42: 535-559.
12. Pedersen CBW, Buhl T, Sigmund O (2001) Topology synthesis of large-displacement compliant mechanisms. Int J Num Methods Eng 50: 2683-2705
13. Sigmund O (1997) On the design of compliant mechanisms using topology optimization. Mech Struct Mach 25: 493-524
14. Sigmund O (2001) A 99 line topology optimization code written in Matlab. Struct Multidiscipl Optim 21: 120-127
15. Sigmund O (2007) Morphology-based black and white filters for topology optimization. Struct Multidiscipl Optim 33(4-5): 401-424
16. Silva MC (2007) Topology optimization to design hinge-free compliant mechanisms (in portuguese). Dissertation, University of São Paulo
17. Svanberg K (1987) The Method of Moving Asymptotes - A new method for structural optimization. Int J Numer Methods Eng 24: 359-373
18. Svanberg K (2002) A class of globally convergent optimization methods based on conservative convex separable approximations. SIAM J Optim 12(2): 555-573.