AN SLP ALGORITHM AND ITS APPLICATION TO TOPOLOGY OPTIMIZATION

FRANCISCO A. M. GOMES AND THADEU A. SENNE

Departamento de Matemática Aplicada, IMECC, Universidade Estadual de Campinas, 13083-859, Campinas, SP, Brazil.

E-mails: chico@ime.unicamp.br / tsenne@ime.unicamp.br

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 [19], 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.

Mathematical subject classification: Primary: 65K05; Secondary: 90C55.

Key words: Topology optimization – Compliant mechanisms – Sequential linear programming – Global convergence theory.

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

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 [1] 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 [2]). 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

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

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 $\mathbf{K}(\boldsymbol{\rho})$ is the stiffness matrix of the structure.

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

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

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 $\mathbf{K}(\boldsymbol{\rho})$ is invertible, and it is possible to eliminate the u variables replacing $\mathbf{u} = \mathbf{K}(\boldsymbol{\rho})^{-1}\mathbf{f}$ in the objective function of problem (1). In this case, the problem reduces to

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

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 optimization problem is the design of a compliant mechanism. Some interesting formulations for this problem were introduced by Nishiwaki et al. [14], Kikuchi et al. [10], Lima [11], Sigmund [16], Pedersen et al. [15], Min and Kim [13], and Luo et al. [12], 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. [14], although some encouraging preliminary results were also obtained for the formulations of Sigmund [16] and Lima [11].

Nishiwaki et al. [14] 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}.

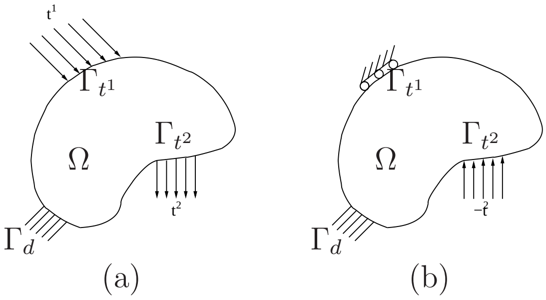

FIGURE 1. The two load cases considered in the formulation of Nishiwaki et al. [14].

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

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

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

$$\begin{aligned} \min_{\boldsymbol{\rho}} \quad & -\frac{\mathbf{f}_b^T\, \mathbf{K}_1(\boldsymbol{\rho})^{-1}\, \mathbf{f}_a}{\mathbf{f}_c^T\, \mathbf{K}_2(\boldsymbol{\rho})^{-1}\, \mathbf{f}_c} \\ \text{s.t.} \quad & \sum_{i=1}^{n_{el}} v_i\, \rho_i \le V \\ & \rho_{min} \le \rho_i \le 1, \qquad i = 1, \ldots, n_{el} \end{aligned}$$

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 [16], 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. [10]; Nishiwaki et al. [14]; Lima [11]; Sigmund [16]). 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 actually used in the literature, global convergence results are not fully established. In part, this problem is due to the fact that classical SLP algorithms, such as those presented in [21] and [8], have practical drawbacks. Besides, recent algorithms that rely on linear programming also include some sort of tangent step that use second order information (e.g. [5] and [6]).

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. Moreover, it is quite simple to implement, and depends only on a good LP library.

3.1. **Description of the method.** Consider the nonlinear programming problem

$$\begin{array}{ll} \min & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{c}(\mathbf{x}) = \mathbf{0}, \\ & \mathbf{x} \in \mathbf{X}, \end{array} \tag{4}$$

where the functions $f : \mathbb{R}^n \to \mathbb{R}$ and $\mathbf{c} : \mathbb{R}^n \to \mathbb{R}^m$ have Lipschitz continuous first derivatives,

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

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 (4).

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 (4) 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 $\mathbf{x}$, (4) can be approximated by the linear programming problem

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

A sequential linear programming (SLP) algorithm is an iterative method that generates and solves a sequence of linear problems in the form (3.1). 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 (3.1), the variables of the original problem (4) are updated according to

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

Unfortunately, this scheme has some pitfalls. First, problem (3.1) may be unlimited even in the case problem (4) has an optimal solution. Besides, the linear functions used to define (3.1) 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 a *trust region* constraint such as $\|\mathbf{s}\|_\infty \leq \delta$, 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 (4).

Including the trust region in (3.1), we get the problem

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

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

However, unless $\mathbf{x}^{(k)}$ satisfies the constraints of (4), it is still possible that problem (5) 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 problem

$$\begin{array}{ll} \min & M(\mathbf{x},\mathbf{s}) = ||\mathbf{A}(\mathbf{x})\mathbf{s} + \mathbf{c}(\mathbf{x})||_1 \\ \text{s.t.} & \mathbf{s}_n^l \leq \mathbf{s} \leq \mathbf{s}_n^u \end{array} \tag{6}$$

where $\mathbf{s}_n^l = \max\{-0.8\delta, \mathbf{x}_l - \mathbf{x}\}$, $\mathbf{s}_n^u = \min\{0.8\delta, \mathbf{x}_u - \mathbf{x}\}$. After solving (6), $\mathbf{x}$, f and $\mathbf{c}$ are updated, in order to make (5) feasible.

Clearly, $M(\mathbf{x}, \mathbf{s})$ is an approximation for the true measure of the infeasibility given by the function

$$\varphi(\mathbf{x}) = \|\mathbf{c}(\mathbf{x})\|_1.$$

In practice, (6) is replaced by the equivalent linear programming problem

$$\begin{array}{lll} & \min & \bar{M}(\mathbf{x}, \mathbf{s}, \mathbf{z}) = \mathbf{e}^T\mathbf{z} \\ & \text{s.t.} & \mathbf{A}(\mathbf{x})\mathbf{s} + \mathbf{E}(\mathbf{x})\mathbf{z} = -\mathbf{c}(\mathbf{x}) \\ (7) & & \mathbf{s}_n^l \leq \mathbf{s} \leq \mathbf{s}_n^u \\ & & \mathbf{z} \geq \mathbf{0}. \end{array}$$

where $\mathbf{z} \in \mathbb{R}^{m_I}$ is a vector of slack variables corresponding to the m_I infeasible constraints, $\mathbf{E}(\mathbf{x}) \in \mathbb{R}^{m \times m_I}$ is a matrix formed by columns of $\mathbf{I}$ or $-\mathbf{I}$, and $\mathbf{e}^T = [1\ 1 \dots 1]$.

Problem (7) is the usual phase 1 problem of the two-phase method for linear programming. To see how matrix $\mathbf{E}(\mathbf{x})$ is constructed, let $\mathbf{I}_i$ represent the i-th column of the identity matrix and suppose that $\{i_1, i_2, \dots, i_{m_I}\}$ are the indices of the nonzero components of $\mathbf{c}(\mathbf{x})$. In this case, the j-th column of $\mathbf{E}(\mathbf{x})$ is given by

$$\mathbf{E}_j(\mathbf{x}) = \begin{cases} \mathbf{I}_{i_j}, & \text{if } c_{i_j}(\mathbf{x}) < 0, \\ -\mathbf{I}_{i_j}, & \text{if } c_{i_j}(\mathbf{x}) > 0. \end{cases}$$

One should notice that the trust region used in (6) and (7) is slightly smaller that the region adopted in (5). This trick is used to give (5) 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 (5) and (6) 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 (4). 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

$$\psi(\mathbf{x}, \theta) = \theta f(\mathbf{x}) + (1-\theta)\varphi(\mathbf{x}), \tag{8}$$

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 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

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

where

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

is the actual reduction of the objective function, and

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

is the reduction of the infeasibility.

The predicted reduction of the merit function is defined as

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

where

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

is the predicted reduction of f and

$$P_{red}^{fsb} = M(\mathbf{x}, \mathbf{0}) - M(\mathbf{x}, \mathbf{s}_c) = ||\mathbf{c}(\mathbf{x})||_1 - ||\mathbf{A}(\mathbf{x})\mathbf{s}_c + \mathbf{c}(\mathbf{x})||_1$$

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 (8) 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.

Following a suggestion given by Gomes et al. [9], at the beginning of the k-th iteration, we define

$$\theta_k = \min\{\theta_k^{large}, \theta_k^{sup}\}, \tag{9}$$

where

$$\theta_k^{large} = \left[1 + \frac{N}{(k+1)^{1.1}}\right] \theta_k^{min}, \tag{10}$$

$$\theta_k^{min} = \min\left\{1, \theta_0, \ldots, \theta_{k-1}\right\}, \tag{11}$$

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

Whenever the step is rejected, θ_k is recomputed. However, this parameter is not allowed to increase within the same iteration.

The constant $N \geq 0$, 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 $\mathbf{x}^{(0)} \in \mathbf{X}$ and an initial trust region radius $\delta_0 \geq \delta_{min} > 0$ are available.

A new SLP method for solving problem (4) is given by Algorithm 1, where we denote $\mathbf{A} \equiv \mathbf{A}(\mathbf{x}^{(k)})$, $\mathbf{E} \equiv \mathbf{E}(\mathbf{x}^{(k)})$, $\nabla f \equiv \nabla f(\mathbf{x}^{(k)})$ and $\mathbf{c} \equiv \mathbf{c}(\mathbf{x}^{(k)})$. 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 (4) 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

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

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 3.1, we suppose that $\mathbf{x}^{(k)}$ is not φ-stationary and (6) is infeasible. Lemma 3.2 deals with the case in which $\mathbf{x}^{(k)}$ is not φ-stationary, but (6) is feasible. Finally, in Lemma 3.3, we suppose that $\mathbf{x}^{(k)}$ is feasible and regular for (4), but does not satisfy the KKT conditions of this problem.

Lemma 3.1. *Suppose that $\mathbf{x}^{(k)}$ is not φ-stationary and that the condition stated in step 3 of Algorithm 1 is not satisfied. Then after a finite number of step rejections, $\mathbf{x}^{(k)} + \mathbf{s}_c$ is accepted.*

Proof. Define $(\mathbf{s}_0, \mathbf{z}_0) = (\mathbf{0},\ -\mathbf{E}(\mathbf{x}^{(k)})^T\mathbf{c})$ as the feasible (yet not basic) initial solution for the restoration problem (7), solved at step 2 of Algorithm 1. Define also

$$\mathbf{d}_n = (\mathbf{d}_s,\ \mathbf{d}_z) = P_{N(\mathbf{x}^{(k)})}(-\nabla \bar{M}(\mathbf{x}^{(k)}, \mathbf{s}_0, \mathbf{z}_0)), \tag{13}$$

Algorithm 1 General SLP algorithm.

while a stopping criterion is not satisfied, **do**

  Determine $\mathbf{s}_n$, the solution of

$$\begin{array}{ll} \min & \mathbf{e}^T\mathbf{z} \\ \text{s.t.} & \mathbf{A}\mathbf{s} + \mathbf{E}\mathbf{z} = -\mathbf{c} \\ & \mathbf{s}_n^l \leq \mathbf{s} \leq \mathbf{s}_n^u \\ & \mathbf{z} \geq \mathbf{0}. \end{array}$$

  if $\bar{M}(\mathbf{x}^{(k)}, \mathbf{s}_n, \mathbf{z}) = 0$, **then**

    Starting from $\mathbf{s}_n$, determine $\mathbf{s}_c$, the solution of

$$\begin{array}{ll} \min & \nabla f^T\mathbf{s} \\ \text{s.t.} & \mathbf{A}\mathbf{s} = -\mathbf{c} \\ & \mathbf{s}_l \leq \mathbf{s} \leq \mathbf{s}_u. \end{array}$$

  else

    $\mathbf{s}_c \leftarrow \mathbf{s}_n$.

  end if

  Determine $\theta_k = \min\{\theta_k^{large}, \theta_k^{sup}, \theta^{max}\}$

  if $A_{red} \geq 0.1P_{red}$ **then**

    $\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} + \mathbf{s}_c$

    if $A_{red} \geq 0.5P_{red}$, **then**

      $\delta_{k+1} \leftarrow \min\{2.5\delta_k, \|\mathbf{x}_u - \mathbf{x}_l\|_\infty\}$

    else

      $\delta_{k+1} \leftarrow \delta_{min}$

    end if

    Recompute $\mathbf{A}$, $\mathbf{E}$ and ∇f.

    $\theta_{max} \leftarrow 1$

    $k \leftarrow k + 1$

  else

    $\delta_k \leftarrow \max\{0.25\|\mathbf{s}_c\|_\infty, 0.1\delta_k\}$

    $\theta_{max} \leftarrow \theta_k$

  end if

end while

where $P_{N(\mathbf{x})}$ denotes the orthogonal projection onto the set

$$(14)\ N(\mathbf{x}) = \big\{(\mathbf{s},\mathbf{z}) \in \mathbb{R}^{n+m_I} \quad | \quad \mathbf{A}(\mathbf{x})\mathbf{s} + \mathbf{E}(\mathbf{x})\mathbf{z} = -\mathbf{c}(\mathbf{x}), \\ \mathbf{x}_l - \mathbf{x} \le \mathbf{s}_n \le \mathbf{x}_u - \mathbf{x},\ \mathbf{z} \ge \mathbf{0}\big\}.$$

For a fixed $\mathbf{x}$, $\bar{M}(\mathbf{x},\mathbf{s},\mathbf{z})$ is a linear function of $\mathbf{s}$ and $\mathbf{z}$. In this case, $\nabla\bar{M}(\mathbf{x},\mathbf{s},\mathbf{z})$ do not depend on these variables, and we can write $\nabla\bar{M}(\mathbf{x})$ for simplicity.

If $\mathbf{x}^{(k)}$ is not φ-stationary and $\bar{M}(\mathbf{x}^{(k)},\mathbf{s}_n,\mathbf{z}) > 0$, the reduction of the infeasibility generated by $\mathbf{s}_c \equiv \mathbf{s}_n$ satisfies

$$(15) \qquad \begin{aligned} P_{red}^{fsb} &\ge M(\mathbf{x}^{(k)},\mathbf{0}) - \bar{M}(\mathbf{x}^{(k)},\mathbf{s}_0 + \bar{\alpha}\mathbf{d}_s, \mathbf{z}_0 + \bar{\alpha}\mathbf{d}_z) \\ &= -\bar{\alpha}\mathbf{e}^T\mathbf{d}_z = -\bar{\alpha}\nabla\bar{M}(\mathbf{x}^{(k)})^T\mathbf{d}_n > 0 \end{aligned}$$

where $\bar{\alpha} = \max\{\alpha \in (0,1] \mid \|\alpha\mathbf{d}_n\|_\infty \le 0.8\delta_k\}$.

After rejecting the step and reducing δ_k a finite number of times, we eventually get $\|\bar{\alpha}\mathbf{d}_n\|_\infty = 0.8\delta_k$. In this case, defining $\eta = -\nabla\bar{M}(\mathbf{x}^{(k)})^T\mathbf{d}_n/\|\mathbf{d}_n\|_\infty$, we have

$$(16) \qquad P_{red}^{fsb} \ge 0.8\eta\delta_k.$$

Now, doing a Taylor expansion, we get

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

so

$$\varphi(\mathbf{x}^{(k)} + \mathbf{s}_c) = \|\mathbf{c}(\mathbf{x}^{(k)} + \mathbf{s}_c)\|_1 = M(\mathbf{x}^{(k)},\mathbf{s}_c) + O(\|\mathbf{s}_c\|^2).$$

Analogously, we have

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

Therefore, for δ_k sufficiently small, $A_{red}(\delta_k) = P_{red}(\delta_k) + O(\delta_k^2)$, so

$$(17) \qquad |A_{red}(\delta_k) - P_{red}(\delta_k)| = O(\delta_k^2).$$

Our choice of θ_k ensures that $P_{red} \ge 0.5P_{red}^{fsb}$. Thus, from (16), we get

$$(18) \qquad P_{red} \ge 0.4\eta\delta_k.$$

Finally, from (17) and (18), we obtain

$$\left| \frac{A_{red}(\delta_k)}{P_{red}(\delta_k)} - 1 \right| = O(\delta_k). \tag{19}$$

Therefore, $A_{red} \geq 0.1 P_{red}$ for δ_k sufficiently small, and the step is accepted. □

Lemma 3.2. *Suppose that $\mathbf{x}^{(k)}$ is not φ-stationary and that the condition stated in step 3 of Algorithm 1 is satisfied. Then after a finite number of step rejections, $\mathbf{x}^{(k)} + \mathbf{s}_c$ is accepted.*

Proof. Let $\delta_k^{(0)}$ be the trust region radius at the beginning of the k-th iteration, and $\mathbf{s}_a$ be the solution of

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

Since $\mathbf{x}^{(k)}$ is not φ-stationary, $\|\mathbf{s}_a\|_\infty > 0$. Now, supposing that the step is rejected j times, we get $\delta_k^{(j)} \leq 0.25^j \delta_k^{(0)}$. Thus, after $\left\lceil \log_2 \sqrt{0.8\delta_k^{(0)}/\|\mathbf{s}_a\|_\infty} \right\rceil$ attempts to reduce δ_k, $\mathbf{s}_n$ is rejected and Lemma 3.1 applies. □

Lemma 3.3. *Suppose that $\mathbf{x}^{(k)}$ is feasible and regular for (4), but does not satisfy the KKT conditions of this problem. 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 (4), then we have $\mathbf{d}_t = P_\Upsilon(-\nabla f(\mathbf{x}^{(k)})) \neq \mathbf{0}$, where P_Υ denotes the orthogonal projection onto the set

$$\Upsilon = \{\mathbf{s} \in \mathbb{R}^n \mid \mathbf{A}(\mathbf{x}^{(k)})\mathbf{s} = \mathbf{0},\ \mathbf{x}_l - \mathbf{x}^{(k)} \leq \mathbf{s} \leq \mathbf{x}_u - \mathbf{x}^{(k)}\}.$$

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

$$\begin{array}{ll} \min & \alpha \nabla f(\mathbf{x}^{(k)})^T \mathbf{d}_t \\ \text{s.t.} & \|\alpha \mathbf{d}_t\|_\infty \leq \delta_k \\ & \alpha > 0. \end{array} \tag{20}$$

Since (20) is a linear programming problem, $\bar{\alpha}\mathbf{d}_t$ belongs to the boundary of the feasible set. Therefore, after reducing δ_k a finite number of times, we get $\|\bar{\alpha}\mathbf{d}_t\|_\infty = \delta_k$, which means that $\bar{\alpha} = \delta_k / \|\mathbf{d}_t\|_\infty$.

Moreover, $\eta = -\nabla f(\mathbf{x}^{(k)})^T \mathbf{d}_t / \|\mathbf{d}_t\|_\infty > 0$, so we have

$$\begin{aligned} L(\mathbf{x}^{(k)}, \mathbf{0}) - L(\mathbf{x}^{(k)}, \bar{\alpha}\mathbf{d}_t) &= -\bar{\alpha}\nabla f(\mathbf{x}^{(k)})^T \mathbf{d}_t \\ &= -\frac{\delta_k}{\|\mathbf{d}_t\|_\infty}\nabla f(\mathbf{x}^{(k)})^T \mathbf{d}_t \\ &= \eta\, \delta_k. \end{aligned} \tag{21}$$

Combining (21) and the fact that $\mathbf{s}_c$ is the solution of (5), 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_k, and

$$P_{red} = \theta_k P_{red}^{opt} \geq \theta_k \eta\, \delta_k. \tag{22}$$

Since (17) also applies in this case, we can combine it with (22) to obtain (19). Therefore, for δ_k 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 (4); or when $\mathbf{x}^{(k)}$ is φ-stationary, but infeasible; or even when $\mathbf{x}^{(k)}$ is feasible but not regular.

We will now investigate what happens when Algorithm 1 generates an infinite sequence of iterates. Our aim is to prove that the limit points of this sequence are φ-stationary. The results shown below follow the line adopted in [9].

Lemma 3.4. *If $\mathbf{x}^* \in \mathbf{X}$ is not φ-stationary, then there exists ε_1, α_1, $\alpha_2 > 0$ such that, if Algorithm 1 is applied to $\mathbf{x} \in \mathbf{X}$ and $\|\mathbf{x} - \mathbf{x}^*\| \leq$*

ε_1, *then*

$$P_{red}(\mathbf{x}) \geq \min\{\alpha_1\delta, \alpha_2\}.$$

Proof. Let $(\mathbf{s}_0^*, \mathbf{z}_0^*) = (\mathbf{0},\ -\mathbf{E}(\mathbf{x}^*)^T\mathbf{c}(\mathbf{x}^*))$ be a feasible initial solution and $(\mathbf{s}_n^*, \mathbf{z}^*)$ be the optimal solution of (7) for $\mathbf{x} \equiv \mathbf{x}^*$.

If $\mathbf{x}^*$ is not φ-stationary, there exists $\varepsilon > 0$ such that, for all $\mathbf{x} \in \mathbf{X}$, $\|\mathbf{x} - \mathbf{x}^*\| \leq \varepsilon$, the constraints that are infeasible at $\mathbf{x}^*$ are also infeasible at $\mathbf{x}$. Thus, we can consider the auxiliary linear programming problem

$$(23)\qquad \begin{array}{ll} \min & \tilde{M}(\mathbf{x},\mathbf{s},\mathbf{z}) = \mathbf{e}^T\mathbf{z} \\ \text{s.t.} & \mathbf{A}(\mathbf{x})\mathbf{s} + \mathbf{E}(\mathbf{x}^*)\mathbf{z} = -\tilde{\mathbf{c}}(\mathbf{x}) \\ & \mathbf{s}_n^l \leq \mathbf{s} \leq \mathbf{s}_n^u \\ & \mathbf{z} \geq \mathbf{0}, \end{array}$$

where $\tilde{\mathbf{c}}_i(\mathbf{x}) = \mathbf{c}_i(\mathbf{x})$ if $\mathbf{c}_i(\mathbf{x}^*) > 0$ and $\tilde{\mathbf{c}}_i(\mathbf{x}) = 0$ if $\mathbf{c}_i(\mathbf{x}^*) = 0$. We denote $(\tilde{\mathbf{s}}_n, \tilde{\mathbf{z}})$ the optimal solution of this problem and $(\mathbf{s}_0, \mathbf{z}_0) = (\mathbf{0},\ -\mathbf{E}(\mathbf{x}^*)^T\tilde{\mathbf{c}}(\mathbf{x}))$ a feasible initial solution.

Following (13), let us define

$$\tilde{\mathbf{d}}_n(\mathbf{x}) = P_{\tilde{N}(\mathbf{x})}(-\nabla\tilde{M}(\mathbf{x})),$$

where $\tilde{N}(\mathbf{x})$ is defined as in (14), using $\mathbf{E}(\mathbf{x}^*)$ and $\tilde{\mathbf{c}}(\mathbf{x})$. One should notice that $\tilde{\mathbf{d}}_n(\mathbf{x}^*) = \mathbf{d}_n(\mathbf{x}^*) = P_{N(\mathbf{x}^*)}(-\nabla\bar{M}(\mathbf{x}^*))$.

Due to the continuity of $\tilde{\mathbf{d}}_n$, there must exist $\varepsilon_1 \in (0,\ \varepsilon]$ such that, for all $\mathbf{x} \in \mathbf{X}$, $\|\mathbf{x} - \mathbf{x}^*\| \leq \varepsilon_1$,

$$-\nabla\tilde{M}(\mathbf{x})^T\tilde{\mathbf{d}}_n(\mathbf{x}) \geq -\frac{1}{2}\nabla\bar{M}(\mathbf{x}^*)^T\mathbf{d}_n(\mathbf{x}^*) > 0$$

and

$$0 < \|\tilde{\mathbf{d}}_n(\mathbf{x})\|_\infty \leq 2\|\mathbf{d}_n(\mathbf{x}^*)\|_\infty.$$

Now, let us consider two situations. Firstly, suppose that, after solving (23), we get $\tilde{M}(\mathbf{x}, \tilde{\mathbf{s}}_n, \tilde{\mathbf{z}}) > 0$. In this case, if $\|\tilde{\mathbf{d}}_n(\mathbf{x})\|_\infty \geq 0.8\delta$, then from (18) we have

$$(24)\qquad P_{red} \geq 0.4\frac{(-\nabla\tilde{M}(\mathbf{x})^T\tilde{\mathbf{d}}_n(\mathbf{x}))}{\|\tilde{\mathbf{d}}_n(\mathbf{x})\|_\infty}\delta \geq 0.1\frac{(-\nabla\bar{M}(\mathbf{x}^*)^T\mathbf{d}_n(\mathbf{x}^*))}{\|\mathbf{d}_n(\mathbf{x}^*)\|_\infty}\delta.$$

On the other hand, if $\|\tilde{\mathbf{d}}_n(\mathbf{x})\|_\infty < 0.8\delta$, then from (15) and our choice of θ,

$$\begin{aligned} P_{red} &\geq 0.5P_{red}^{fsb} \geq -0.5\nabla\tilde{M}(\mathbf{x})^T\tilde{\mathbf{d}}_n(\mathbf{x}) \\ &\geq -0.25\nabla\bar{M}(\mathbf{x}^*)^T\mathbf{d}_n(\mathbf{x}^*). \end{aligned} \tag{25}$$

Finally, let us suppose that, after solving (23), we get $\tilde{M}(\mathbf{x}, \tilde{\mathbf{s}}_n, \tilde{\mathbf{z}}) = 0$. In this case, $P_{red}^{fsb} = \tilde{M}(\mathbf{x}, \tilde{\mathbf{s}}_0, \tilde{\mathbf{z}}_0)$, i.e. P_{red}^{fsb} is maximum, so (25) also holds.

The desired result follows from (24) and (25), for an appropriate choice of α_1 and α_2. □

Lemma 3.5. *Suppose that* $\mathbf{x}^*$ *is not* φ*-stationary and that* 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 exist* $\alpha_3 > 0$ *and* $\bar{k} > 0$ *such that, for* $k \in K_1$, $k \geq \bar{k}$, *we have* $A_{red} \geq \alpha_3$.

Proof. For $k \in K_1$ large enough, we have $\|\mathbf{x} - \mathbf{x}^*\| \leq \varepsilon_1$, where ε_1 is defined in Lemma 3.4. In this case, from Lemma 3.1 we deduce that the step is never rejected whenever its norm is smaller than some $\delta_1 > 0$. Thus, δ_k is bounded away from zero. Moreover, from our step acceptance criterion and Lemma 3.4, we obtain

$$A_{red} \geq 0.1P_{red} \geq 0.1\min\{\alpha_1\delta_1,\ \alpha_2\}.$$

The desired result is achieved choosing $\alpha_3 = 0.1\min\{\alpha_1\delta_1,\ \alpha_2\}$. □

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 (4).

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

Theorem 3.6. *Suppose that H1 holds. If* $\{\mathbf{x}^{(k)}\}$ *is an infinite sequence generated by Algorithm 1, then every limit point of* $\{\mathbf{x}^{(k)}\}$ *is* φ*-stationary.*

Proof. To simplify the notation, let us write $f_k = f(\mathbf{x}^{(k)})$, $\varphi_k = \varphi(\mathbf{x}^{(k)})$, $\psi_k = \psi(\mathbf{x}^{(k)}, \theta_k)$, and $A_{red}^{(k)} = A_{red}(\mathbf{x}^{(k)}, \mathbf{s}_c^{(k)}, \theta_k)$. From (8), we have that

$$\begin{aligned} \psi_k &= \theta_k f_k + (1-\theta_k)\varphi_k \\ &\quad -[\theta_{k-1} f_k + (1-\theta_{k-1})\varphi_k] + [\theta_{k-1} f_k + (1-\theta_{k-1})\varphi_k] \\ &= (\theta_k - \theta_{k-1}) f_k - (\theta_k - \theta_{k-1})\varphi_k + \theta_{k-1} f_k + (1-\theta_{k-1})\varphi_k \\ &= (\theta_k - \theta_{k-1})(f_k - \varphi_k) + \psi_{k-1} - A_{red}^{(k-1)}. \end{aligned}$$

Besides, from (9)-(11), we also have that

$$\theta_k - \theta_{k-1} \leq \frac{N}{(k+1)^{1.1}} \theta_{k-1}.$$

Hypothesis H1 implies that there exists an upper bound $c > 0$ such that $|f_k - \varphi_k| \leq c$ for all $k \in \mathbb{N}$, so

$$\psi_k \leq \frac{cN}{(k+1)^{1.1}} \theta_{k-1} + \psi_{k-1} - A_{red}^{(k-1)}. \tag{26}$$

Noting that $\theta_k \in [0,1]$ for all k, and applying (26) recursively, we get

$$\psi_k \leq \sum_{j=1}^{k} \frac{cN}{(j+1)^{1.1}} + \psi_0 - \sum_{j=0}^{k-1} A_{red}^{(j)}.$$

Since the series $\sum_{j=1}^{\infty} \frac{cN}{(j+1)^{1.1}}$ is convergent, the inequality above may be written as

$$\psi_k \leq \tilde{c} - \sum_{j=0}^{k-1} A_{red}^{(j)}.$$

Let us now suppose that $\mathbf{x}^* \in X$ is a limit point of $\{\mathbf{x}^{(k)}\}$ that is not φ-stationary. Then, from Lemma 3.5, there exists $\alpha_3 > 0$ such that $A_{red}^{(k)} \geq \alpha_3$ for an infinite set of indices. Besides, $A_{red}^{(k)} > 0$ for all k. Thus, ψ_k is unbounded below, which contradicts Hypothesis H1, proving the lemma. □

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 (4).

Lemma 3.7. *For each feasible and regular point* $\mathbf{x}^*$ *there exists* ϵ_0*,* $\hat{\sigma} > 0$ *such that, whenever Algorithm 1 is applied to* $\mathbf{x} \in \mathbf{X}$ *that satisfies* $\|\mathbf{x} - \mathbf{x}^*\| \leq \epsilon_0$*, we have*

$$\|\mathbf{s}_n\|_\infty \leq \|\mathbf{c}(\mathbf{x})\|_1/\hat{\sigma}.$$

and

$$M(\mathbf{x}, \mathbf{0}) - M(\mathbf{x}, \mathbf{s}_n(\mathbf{x}, \delta)) \geq \min\{\|\mathbf{c}(\mathbf{x})\|_1, \hat{\sigma}\delta\}.$$

Proof. Since $\mathbf{A}(\mathbf{x})$ is Lipschitz continuous, for each $\mathbf{x}^*$ that is feasible and regular, there exists ϵ_0 such that, for all $\mathbf{x} \in \mathbf{X}$ satisfying $\|\mathbf{x} - \mathbf{x}^*\| \leq \epsilon_0$, $\mathbf{A}(\mathbf{x})$ has full row rank and the linear programming problem

$$(27) \qquad \begin{array}{ll} \min & \bar{M}(\mathbf{x}, \mathbf{s}, \mathbf{z}) = \mathbf{e}^T\mathbf{z} \\ \text{s.t.} & \mathbf{A}(\mathbf{x})\mathbf{s} + \mathbf{E}(\mathbf{x})\mathbf{z} = -\mathbf{c}(\mathbf{x}) \\ & \mathbf{x}_l - \mathbf{x} \leq \mathbf{s} \leq \mathbf{x}_u - \mathbf{x} \\ & \mathbf{z} \geq \mathbf{0}. \end{array}$$

has an optimal solution $(\mathbf{s}, \mathbf{z}) = (\hat{\mathbf{s}}, 0)$. In this case, $\mathbf{A}(\mathbf{x})\hat{\mathbf{s}} = -\mathbf{c}(\mathbf{x})$, so $\|\hat{\mathbf{s}}\|_2 \leq \|\mathbf{c}(x)\|_2/\hat{\sigma}$, where $\hat{\sigma} > 0$ is the smallest singular value of $\mathbf{A}(\mathbf{x})$.

Adding the trust region constraint $\|\mathbf{s}\|_\infty \leq 0.8\delta$ to (27), we get (7) (that is, the linear programming problem solved at step 2 of Algorithm 1). In this case,

$$\|\mathbf{s}_n\|_\infty \leq \|\hat{\mathbf{s}}\|_\infty \leq \|\hat{\mathbf{s}}\|_2 \leq \|\mathbf{c}(\mathbf{x})\|_2/\hat{\sigma} \leq \|\mathbf{c}(\mathbf{x})\|_1/\hat{\sigma},$$

proving the first part of the lemma.

If $(\hat{\mathbf{s}}, 0)$ is also feasible for (7), then $\mathbf{s}_n = \hat{\mathbf{s}}$, and we have

$$(28) \qquad M(\mathbf{x}, \mathbf{0}) - M(\mathbf{x}, \mathbf{s}_n(\mathbf{x}, \delta)) = M(\mathbf{x}, \mathbf{0}) = \|\mathbf{c}(\mathbf{x})\|_1.$$

On the other hand, if $\|\hat{\mathbf{s}}\|_\infty > 0.8\delta$, then we can define $\hat{\mathbf{s}}_n = \delta\hat{\mathbf{s}}/\|\hat{\mathbf{s}}\|_\infty$ and $\hat{\mathbf{z}}_n = (1 - \delta/\|\hat{\mathbf{s}}\|_\infty)\mathbf{z}_0$ (where $\mathbf{z}_0$ is the $\mathbf{z}$ vector corresponding to $\mathbf{s} = \mathbf{0}$), so $(\hat{\mathbf{s}}_n, \hat{\mathbf{z}}_n)$ is now feasible for (7). Moreover, since $\bar{M}(\mathbf{x}, \mathbf{0}, \mathbf{z}_0) = \|\mathbf{c}(\mathbf{x})\|_1$, $\bar{M}(\mathbf{x}, \hat{\mathbf{s}}, \mathbf{0}) = 0$, and $\bar{M}$ is a linear function of $\mathbf{s}$

and $\mathbf{z}$, we have $M(\mathbf{x}, \mathbf{s}_n(\mathbf{x}, \delta)) = \bar{M}(\mathbf{x}, \hat{\mathbf{s}}_n, \hat{\mathbf{z}}_n) = (1-\delta/\|\hat{\mathbf{s}}\|_\infty)\|\mathbf{c}(\mathbf{x})\|_1$. Thus,

$$M(\mathbf{x}, \mathbf{0}) - M(\mathbf{x}, \mathbf{s}_n(\mathbf{x}, \delta)) = \delta\|\mathbf{c}(\mathbf{x})\|_1/\|\hat{\mathbf{s}}\|_\infty \geq \hat{\sigma}\delta. \tag{29}$$

The second part of the lemma follows from (28) and (29). □

Lemma 3.8. *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 (4). Then there exist c_1,k_1,$\delta' > 0$ such that, for $\mathbf{x} \in \{\mathbf{x}^{(k)} \mid k \in K_1,\ k \geq k_1\}$, whenever $\bar{M}(\mathbf{x}, \mathbf{s}_n, \mathbf{z}) = 0$ at step 3 of Algorithm 1, we have*

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

Proof. Analogously to what was done in Lemma 3.3, let us define $\mathbf{d}_t = P_\Gamma(-\nabla f(\mathbf{x}))$, where

$$\Gamma = \{\mathbf{s} \in \mathbb{R}^n \mid \mathbf{A}(\mathbf{x})\mathbf{s} = \mathbf{0},\ \mathbf{x}_l \leq \mathbf{x} + \mathbf{s}_n + \mathbf{s} \leq \mathbf{x}_u\}.$$

Let us also denote $\mathbf{s}_t^d$ the solution of

$$\begin{array}{ll} \min & L(\mathbf{x}, \mathbf{s}_n + \mathbf{s}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{s}_n + \mathbf{s}) \\ \text{s.t.} & \mathbf{s} = t\mathbf{d}_t, \quad t \geq 0 \\ & \|\mathbf{s}_n + \mathbf{s}\|_\infty \leq \delta \\ & \mathbf{x}_l \leq \mathbf{x} + \mathbf{s}_n + \mathbf{s} \leq \mathbf{x}_u \end{array} \tag{30}$$

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

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

where

$$\bar{t} = \min\left\{1, \Delta_1, \Delta_2\right\},$$

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

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

Since (30) is a linear programming problem and $\nabla f(\mathbf{x})^T \mathbf{d}_t < 0$, we conclude that $\tilde{t} = \bar{t}$. Besides, $t = 1$ satisfies $\mathbf{x}_l \leq \mathbf{x} + \mathbf{s}_n + \mathbf{s} \leq \mathbf{x}_u$, so

$$\bar{t} = \min \left\{ 1,\ \min_{d_{t_i} < 0} \left\{ \frac{\delta + s_{n_i}}{-d_{t_i}} \right\},\ \min_{d_{t_i} > 0} \left\{ \frac{\delta - s_{n_i}}{d_{t_i}} \right\} \right\}. \tag{31}$$

Remembering that $\mathbf{s}_c$ is the solution of (5), we obtain

$$L(\mathbf{s}_n) - L(\mathbf{s}_c) \geq L(\mathbf{s}_n) - L(\mathbf{s}_n + \mathbf{s}_t^d) = -\bar{t}\, \nabla f(\mathbf{x})^T \mathbf{d}_t. \tag{32}$$

Since $P_\Gamma(-\nabla f(\mathbf{x}))$ is a continuous function of $\mathbf{x}$, and $\mathbf{x}^*$ is regular and feasible but not stationary, there exist $c_1',\ c_2' > 0$ and $k_1 \geq 0$ such that, for all $\mathbf{x} \in \{\mathbf{x}^{(k)} \,|\, k \in K_1,\ k \geq k_1\}$,

$$\|\mathbf{d}_t\|_\infty \leq c_1' \tag{33}$$

and

$$-\nabla f(\mathbf{x})^T \mathbf{d}_t \geq c_2'. \tag{34}$$

From (31) and the fact that $\|\mathbf{s}_n\|_\infty \leq 0.8\delta_k$, we have that

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

Thus, from (33) we obtain

$$\bar{t} \geq \min \left\{ 1,\ \frac{0.2\delta}{c_1'} \right\} = \frac{0.2}{c_1'} \min \left\{ \frac{c_1'}{0.2},\ \delta \right\}. \tag{35}$$

Combining (32), (34) and (35), 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 result is obtained taking $c_1 = \dfrac{0.2c_2'}{c_1'}$ and $\delta' = \dfrac{c_1'}{0.2}$. $\square$

Lemma 3.9. *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*

(4). Then there exist $\beta, c_2, k_2 > 0$ such that, whenever $\mathbf{x} \in \{\mathbf{x}^{(k)} \mid k \in K_1,\ k \geq k_2\}$ and $\|\mathbf{c}(\mathbf{x})\|_1 \leq \beta\delta_k$,

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

and

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

where θ^{sup} is given by (12) and δ' is defined in Lemma 3.8.

Proof. From Lemma 3.7, we obtain

$$\|\mathbf{s}_n\|_\infty \leq \|\mathbf{c}(\mathbf{x})\|_1/\hat{\sigma} \leq \beta\delta_k/\hat{\sigma}.$$

Therefore, defining $\beta = 0.8\hat{\sigma}$, we get $\|\hat{\mathbf{s}}\|_\infty \leq 0.8\delta_k$, so $\bar{M}(\mathbf{x}, \mathbf{s}_n, \mathbf{z}) = 0$ at step 3 of Algorithm 1.

From Lemma 3.8 and the Lipschitz continuity of $\nabla f(\mathbf{x})$, we can define $k_2 \geq 0$ such that

$$\begin{aligned} L(\mathbf{0}) - L(\mathbf{s}_c) &\geq L(\mathbf{s}_n) - L(\mathbf{s}_c) - |L(\mathbf{0}) - L(\mathbf{s}_n)| \\ &\geq c_1 \min\{\delta,\ \delta'\} - O(\|\mathbf{c}(\mathbf{x})\|), \end{aligned}$$

for all $\mathbf{x} \in \{\mathbf{x}^{(k)} \mid k \in K_1,\ k \geq k_2\}$. Thus, choosing β conveniently, we prove the first statement of the Lemma.

To prove the second part of the lemma, we note that

$$P_{red}^{fsb} = M(\mathbf{0}) - M(\mathbf{s}_c) = M(\mathbf{0}) - M(\mathbf{s}_n) = \|\mathbf{c}(\mathbf{x})\|_1,$$

so

$$P_{red}^{opt} - 0.5P_{red}^{fsb} \geq c_2 \min\{\delta,\ \delta'\} - 0.5\|\mathbf{c}(\mathbf{x})\|_1.$$

Thus, for an appropriate choice of β, we obtain $P_{red} > 0.5P_{red}^{fsb}$ for $\theta = 1$, and we get the desired result. □

Lemma 3.10. *Let $\{\mathbf{x}^{(k)}\}$ be an infinite sequence generated by Algorithm 1. Suppose that H1 holds, and 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 (4). Then $\lim_{k\to\infty} \theta_k = 0$.*

Proof. The sequences $\{\theta_k^{min}\}$ and $\{\theta_k^{large}\}$ are bounded below and nonincreasing, so both are convergent. Moreover, they converge to the same limit, as $\lim_{k\leftarrow\infty}(\theta_k^{large} - \theta_k^{min}) = 0$. Besides, $\theta_{k+1}^{min} \leq \theta_k \leq \theta_k^{large}$. Therefore, $\{\theta_k\}$ is convergent.

Suppose, for the purpose of obtaining a contradiction, that the infinite sequence $\{\theta_k\}$ does not converge to 0. Thus, there must exist $k_3 \geq k_2$ and $\widehat{\theta}_U > \widehat{\theta}_L > 0$ such that $\widehat{\theta}_L \leq \theta_k \leq \widehat{\theta}_U$ for $k \geq k_3$.

Now, suppose that $\mathbf{x} \in \{\mathbf{x}^{(k)} \mid k \in K_1,\, k \geq k_3\}$, and $M(\mathbf{x}, \mathbf{s}_n) = 0$. In this case, from Lemma 3.8, we obtain

$$P_{red} \geq \theta[L(\mathbf{x}, \mathbf{0}) - L(\mathbf{x}, \mathbf{s}_c)] \geq \theta c_1 \min\{\delta,\, \delta'\} - O(\|\mathbf{c}(\mathbf{x})\|_1).$$

Since θ is not increased if the step is rejected, for each θ tried at the iteration that corresponds to x, we have that

$$P_{red} \geq \widehat{\theta}_L c_1 \min\{\delta,\, \delta'\} - O(\|\mathbf{c}(\mathbf{x})\|_1).$$

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

$$|A_{red} - P_{red}| = O(\delta^2). \tag{36}$$

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

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

Let $k_3' \geq k_3$ be an iteration index such that, for all $\mathbf{x} \in \{\mathbf{x}^{(k)} \mid k \in K_1,\, k \geq k_3'\}$, and for all θ tried at the iteration that corresponds to x, we have

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

If, in addition, $\delta \in [\widetilde{\delta}/10,\, \widetilde{\delta})$, then

$$P_{red} \geq \widehat{\theta}_L c_1 \widetilde{\delta}/20.$$

Therefore, for all $\delta \in [\widetilde{\delta}/10,\, \widetilde{\delta})$ and all $\mathbf{x} \in \{\mathbf{x}^{(k)} \mid k \in K_1,\, k \geq k_3'\}$, we have

$$\frac{|A_{red} - P_{red}|}{P_{red}} \leq 0.5, \tag{37}$$

On the other hand, if $M(\mathbf{x}, \mathbf{s}_n) > 0$, then $P_{red}^{opt} = 0$, so $P_{red} = (1-\theta)P_{red}^{fsb}$. In this case, from (29) and the fact that θ is not increased if the step is rejected, we get

$$P_{red} \geq (1 - \widehat{\theta}_U)\hat{\sigma}\delta.$$

Using (36) again, there exists $\widetilde{\delta} \in (0, \delta_{min})$ such that, if $\delta \in (0, \widetilde{\delta})$ and $\mathbf{x} \in \{\mathbf{x}^{(k)} \mid k \in K_1, k \geq k_3\}$,

$$|A_{red} - P_{red}| \leq (1 - \widehat{\theta}_U)\hat{\sigma}\widetilde{\delta}/2,$$

so (37) also applies.

Thus, for some $\delta \in [\widetilde{\delta}/10, \widetilde{\delta})$, the step is accepted, which means that δ_k is bounded away from zero for $k \in K_1$, $k \geq k_3'$, so P_{red} is also bounded away from zero.

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

Lemma 3.11. *Let $\{\mathbf{x}^{(k)}\}$ be an infinite sequence generated by Algorithm 1. Suppose that H1 holds, and 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 (4). If $\mathbf{x} \in \{\mathbf{x}^{(k)} \mid k \in K_1, k \geq k_2\}$ and $\|\mathbf{c}(\mathbf{x})\|_1 \geq \beta\delta$, then*

$$\delta/\theta^{sup} = O(\|\mathbf{c}(\mathbf{x})\|_1).$$

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

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

From Lemma 3.7, if $\mathbf{x} \in \{\mathbf{x}^{(k)} \,|\, k \in K_1,\, k \geq k_2\}$, we have that

$$\begin{aligned} \frac{1}{2\theta^{sup}} &= 1 + \frac{L(\mathbf{s}_c) - L(\mathbf{s}_n)}{M(\mathbf{0}) - M(\mathbf{s}_n)} + \frac{L(\mathbf{s}_n) - L(\mathbf{0})}{M(\mathbf{0}) - M(\mathbf{s}_n)} \\ &\leq 1 + \frac{|L(\mathbf{0}) - L(\mathbf{s}_n)|}{M(\mathbf{0}) - M(\mathbf{s}_n)} \\ &\leq 1 + \frac{O(\|\mathbf{c}(\mathbf{x})\|_1)}{\min\{\|\mathbf{c}(\mathbf{x})\|_1, \hat{\sigma}\delta\}} \leq 1 + \frac{O(\|\mathbf{c}(\mathbf{x})\|_1)}{\min\{\beta, \hat{\sigma}\}\,\delta}. \end{aligned}$$

Therefore, since $\|\mathbf{c}(\mathbf{x})\|_1 \geq \beta\delta$, we have $\delta/\theta^{sup} = O(\|\mathbf{c}(\mathbf{x})\|_1)$. □

Lemma 3.12. *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 (4). Then there exist $k_4 > 0, \tilde{\theta} \in (0, 1]$ such that, if $\mathbf{x} \in \{\mathbf{x}^{(k)} \,|\, k \in K_1,\, k \geq k_4\}$, $\|\mathbf{c}(\mathbf{x})\|_1 \geq \beta\delta$ and $\theta \leq \tilde{\theta}$ satisfies (9)-(12), then $A_{red} \geq 0.1 P_{red}$.*

Proof. From the fact that $\nabla f(\mathbf{x})$ and $\mathbf{A}(\mathbf{x})$ are Lipschitz continuous, we may write $A_{red} = P_{red} + O(\delta^2)$. Now, supposing that $\|\mathbf{c}(\mathbf{x})\|_1 \geq \beta\delta$, we have

$$|A_{red} - P_{red}| = \delta\, O(\|\mathbf{c}(\mathbf{x})\|_1). \tag{38}$$

Since our choice of θ ensures that $P_{red} \geq 0.5[M(\mathbf{0}) - M(\mathbf{s}_c)]$, Lemma 3.7 implies that, for $k \in K_1$ sufficiently large,

$$P_{red} \geq 0.5 \min\{\|\mathbf{c}(\mathbf{x})\|_1, \hat{\sigma}\delta\} \geq 0.5 \min\{\beta, \hat{\sigma}\}\,\delta.$$

Thus, dividing both sides of (38) by P_{red}, we get

$$\left|\frac{A_{red}}{P_{red}} - 1\right| = O(\|\mathbf{c}(\mathbf{x})\|_1),$$

which yields the desired result. □

Lemma 3.13. *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 Hypothesis H1 holds. Then, there exists a limit point of $\{\mathbf{x}^{(k)}\}$ that is a stationary point of problem (4).*

Proof. Following the guidelines of Lemma 13 of [9], we note that, by Hypothesis H1, there exists a convergent subsequence $\{\mathbf{x}^{(k)}\}_{k\in K_1}$. Suppose, for the purpose of obtaining a contradiction, that the limit point of this subsequence is not a stationary point of (4). Then, from Lemma 3.10, we have that

$$\lim_{k\leftarrow\infty} \theta_k = 0.$$

Since (10)-(11) imply that $\theta_k^{large} > \min\{1, \theta_0, \theta_1, \ldots, \theta_{k-1}\}$, there must exist an infinite subset $K_2 \subset K_1$ such that

$$\lim_{k\in K_2} \theta_k^{sup}(\underline{\delta}_k) = 0, \tag{39}$$

where $\underline{\delta}_k$ is one of the trust region radii tested at iteration k. Therefore, there also exists $\tilde{\theta}$, $k_5 > 0$ such that, for all $k \in K_2$, $k \geq k_5$, we have $\theta_k^{large} \leq 2\theta_k^{min}$,

$$\theta_k^{sup}(\underline{\delta}_k) \leq \tilde{\theta}/2 < 1, \quad \text{and} \quad \theta_k \leq \tilde{\theta}/2. \tag{40}$$

Lemma 3.9 assures that $\theta_k^{sup}(\delta) = 1$ for all $k \in K_2$ whenever $\|\mathbf{c}(\mathbf{x}^{(k)})\|_1 \leq \beta\delta$. So, by (39) and (40),

$$\|\mathbf{c}(\mathbf{x}^{(k)})\|_1 > \beta\underline{\delta}_k \tag{41}$$

for all $k \in K_2$. Therefore, since $\|\mathbf{c}(\mathbf{x}^{(k)})\|_1 \to 0$,

$$\lim_{k\in K_2} \underline{\delta}_k = 0.$$

Assume, without loss of generality, that

$$\underline{\delta}_k \leq 0.1\delta' < 0.1\delta_{min} \tag{42}$$

for all $k \in K_2$, where δ' is defined in Lemma 3.8. Thus, $\underline{\delta}_k$ cannot be the first trust region radius tried at iteration k. Let us call $\hat{\delta}_k$ the trust region radius tried immediately before $\underline{\delta}_k$, and $\hat{\theta}_k$ the penalty parameter associated to this rejected step. By (40) and the choice of the penalty parameter, we get $\hat{\theta}_k \leq \tilde{\theta}$ for all $k \in K_2, k \geq k_5$. Therefore, Lemma 3.12 applies, so

$$\|\mathbf{c}(\mathbf{x}^{(k)})\|_1 < \beta\hat{\delta}_k.$$

for all $k \in K_2, k \geq k_5$. Moreover, since $\underline{\delta}_k \geq 0.1\hat{\delta}_k$, inequality (42) implies that

$$\hat{\delta}_k \leq 10\underline{\delta}_k \leq \delta' < \delta_{min}. \tag{43}$$

Let us define $\theta'(\hat{\delta}_k) = \theta_k^{large}$ if $\hat{\delta}_k$ was the first trust region radius tested at iteration k, and $\theta'(\hat{\delta}_k) = \theta(\delta'_k)$ otherwise, where δ'_k is the penalty parameter tried immediately before $\hat{\delta}_k$ at iteration k.

From (9)-(12), the fact that θ is not allowed to increase within an iteration, equation (39) and Lemma 3.9, we have

$$\begin{aligned} \hat{\theta}_k &= \min\{\theta'_k(\hat{\delta}_k),\ \theta_k^{sup}(\hat{\delta}_k)\} = \theta'_k(\hat{\delta}_k) \\ &\geq \min\{\theta'_k(\hat{\delta}_k),\ \theta_k^{sup}(\underline{\delta}_k)\} = \theta_k^{sup}(\underline{\delta}_k) \end{aligned} \tag{44}$$

for all $k \in K_2, k \geq k_5$.

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

$$|A_{red}(\hat{\theta}_k, \hat{\delta}_k) - P_{red}(\hat{\theta}_k, \hat{\delta}_k)| = O(\hat{\delta}^2).$$

for all $k \in K_2, k \geq k_5$. Besides, by Lemma 3.9, (43) and the definition of P_{red}, we obtain

$$P_{red}(\hat{\theta}_k, \hat{\delta}_k) \geq \hat{\theta}_k c_2 \hat{\delta}_k,$$

so

$$\frac{|A_{red}(\hat{\theta}_k, \hat{\delta}_k) - P_{red}(\hat{\theta}_k, \hat{\delta}_k)|}{P_{red}(\hat{\theta}_k, \hat{\delta}_k)} = \frac{O(\hat{\delta}^2)}{\hat{\theta}_k c_2 \hat{\delta}_k} = \frac{O(\hat{\delta})}{\hat{\theta}_k}. \tag{45}$$

From Lemma 3.11 and (41), we obtain $\underline{\delta}_k/\theta_k^{sup}(\underline{\delta}_k) = O(\|\mathbf{c}(\mathbf{x}^{(k)})\|_1$ for $k \in K_2, k \geq k_5$. So, by (43) and (44), we also have $\hat{\delta}_k/\theta_k^{sup}(\underline{\delta}_k) = O(\|\mathbf{c}(\mathbf{x}^{(k)})\|_1$. Therefore, from the feasibility of $\mathbf{x}^*$, the right-hand side of (45) tends to zero for $k \in K_2, k \geq k_5$. This implies that, for k large enough, $A_{red} \geq 0.1P_{red}$ for $\hat{\delta}_k$, contradicting the assumption that $\hat{\delta}_k$ was rejected. □

Theorem 3.14. *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 (4). 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 (4).*

Proof. This result follows from Theorem 3.6 and Lemma 3.13. □

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 [7]). 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 [3]; Sigmund [16]). However, more elaborate approaches, such as the the Sinh method of Bruns [4] and the morphology-based filters proposed by Sigmund [17], 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. **Density filter.** A very simple filter was proposed by Bruns and Tortorelli [3] 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

$$\phi_i \equiv \phi_i(\boldsymbol{\rho}) = \sum_{j \in B_i} \frac{\omega_j(s_{ij})}{\omega_i} \rho_j, \tag{46}$$

where

$$\omega_j(s_{ij}) = \begin{cases} \frac{\exp(-s_{ij}^2/2(r/3)^2)}{2\pi(r/3)} & \text{if } s_{ij} \le r, \\ 0 & \text{if } s_{ij} > r, \end{cases} \tag{47}$$

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

$$\omega_i = \sum_{j \in B_i} \omega_j(s_{ij}). \tag{48}$$

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

4.2. **Morphology-based filters.** Sigmund [17] 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

$$\widetilde{\rho}_i = \frac{1}{\beta} \log \left(\frac{\sum_{j \in B_i} \exp(\beta \rho_j)}{\sum_{j \in B_i} 1} \right), \tag{49}$$

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 [17], so ρ_i is replaced by

$$\bar{\rho}_i = 1 - \frac{1}{\beta} \log \left(\frac{\sum_{j \in B_i} \exp(\beta(1 - \rho_j))}{\sum_{j \in B_i} 1} \right), \tag{50}$$

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

It is easy to see that

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

Unfortunately, choosing a large β may result in numerical instabilities. Thus, Sigmund [17] 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.

4.3. **Sinh filter.** The Sinh method of Bruns [4] 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, $\eta_1(\boldsymbol{\rho})$, is employed in the objective function of the topological optimization problem, while the second, $\eta_2(\boldsymbol{\rho})$, replaces the true density in the constraints.

Bruns [4] has proposed several definitions for $\eta_1(\boldsymbol{\rho})$ and $\eta_2(\boldsymbol{\rho})$. The basic Sinh method is obtained combining

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

and

$$\eta_{2_i}(\boldsymbol{\rho}) = 1 - \frac{\sinh\{p[1 - \phi_i(\boldsymbol{\rho})]\}}{\sinh(p)}, \quad i = 1, \ldots, n_{el}, \tag{51}$$

where $\phi_i(\boldsymbol{\rho})$ is computed according to (46)–(48), and p is a penalty factor.

One disadvantage of this approach is that, due to the presence of the sinh function in (51), 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 and 4 constitute the core of Algorithm 1. The implementation of the remaining steps is straightforward.

Step 2 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 4. One should note, however, that the bounds on the variables defined at Steps 2 and 4 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.

On the other hand, when the constraints given in Step 2 are incompatible, the step $\mathbf{s}_c$ is just the solution obtained by the simplex algorithm at the end of phase 1. 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 4 directly. In case the algorithm fails to obtain a feasible solution, then Steps 2 need to be performed. Fortunately, in the solution of topology optimization, the feasible region of (5) 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 [20].

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. [14] formulation mentioned in section 2. Some preliminary results with the formulations of Lima [11] and Sigmund [16] 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 the density, the dilation and the erosion filters. In all cases, the penalty parameter p was set to 1, 2 and 3, consecutively. When the sinh method was adopted, the parameter p given in (51) was set to 1 to 6, consecutively.

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 (49) and (50)).

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.

Although not technically sound, this stopping criterion based on the function improvement is quite common in topology optimization.

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 $E = 1\,N/cm^2$. The volume of the optimal structure is limited by 40% of the design domain. A force $f = 1\,N$ is applied downwards in the center of the right edge of the beam.

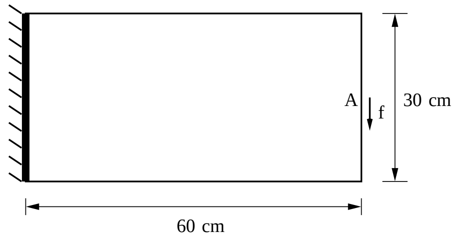

FIGURE 2. Design domain for the cantilever beam.

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.

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.

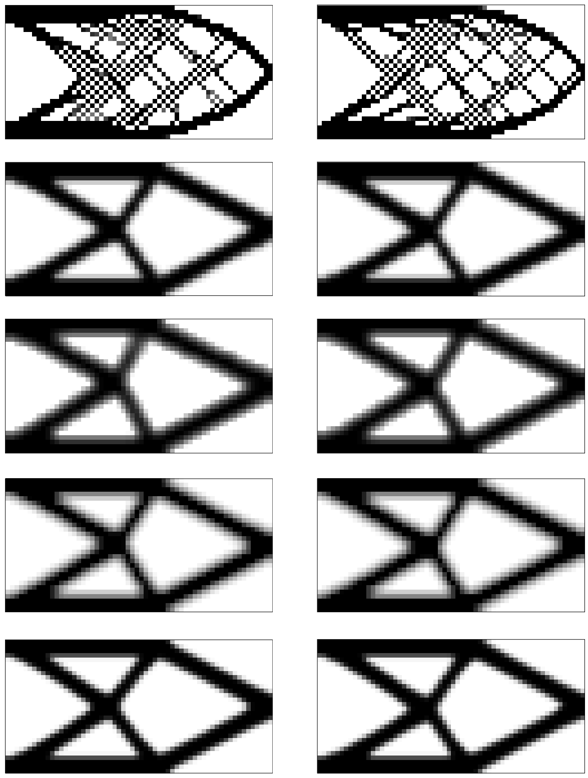

FIGURE 3. The cantilever beams obtained using various filter and method combinations. The left column presents the topologies generated by the SLP method, while the right column present the topologies found by CCSA. The first pair of structures was obtained without using a filter. The remaining pairs were generated using the density, the dilation, the erosion and the sinh filters, respectively.

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 $f = 1\,N$ is applied downwards in the center of the top edge of the beam.

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**
Density filter ($r_{min} = 2.0$)				
SLP	0.05	69.8934	381	180.80
CCSA	0.15	69.8444	947	2171.00
Ratio	-	1.001	**0.402**	**0.083**
Dilation filter ($r_{min} = 1.0$)				
SLP	0.10	80.3363	1058	691.20
CCSA	0.05	79.7601	1500	8533.30
Ratio	-	1.007	**0.705**	**0.081**
Erosion filter ($r_{min} = 1.0$)				
SLP	0.10	63.8413	953	557.08
CCSA	0.05	63.8197	1416	2168.50
Ratio	-	1.000	**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**

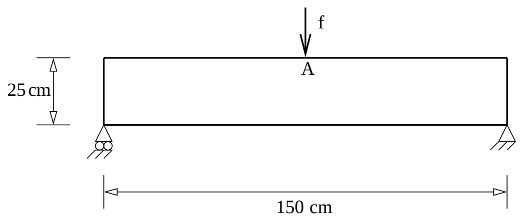

FIGURE 4. Design domain for the MBB beam.

The domain was discretized into 3750 square elements with $1\,mm^2$ each. The optimal structures 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.

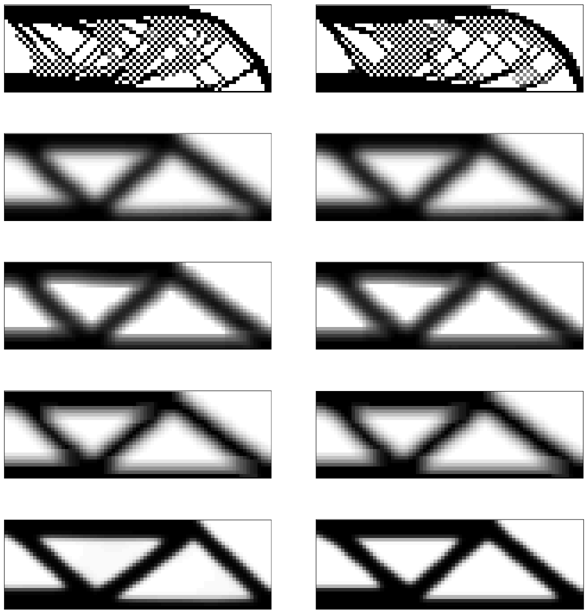

FIGURE 5. The MBB beams obtained using various filter and method combinations. The left column presents the topologies generated by the SLP method, while the right column present the topologies found by CCSA. The first pair of structures was obtained without using a filter. The remaining pairs were generated using the density, the dilation, the erosion and the sinh filters, respectively.

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.

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**
Density filter ($r_{min} = 5.0$)				
SLP	0.05	181.2884	921	1046.00
CCSA	0.10	181.3468	1500	5339.40
Ratio	-	1.000	**0.614**	**0.196**
Dilation filter ($r_{min} = 2.0$)				
SLP	0.10	197.3618	1293	1094.20
CCSA	0.05	208.5159	1500	8394.70
Ratio	-	**0.947**	**0.862**	**0.130**
Erosion filter ($r_{min} = 2.0$)				
SLP	0.10	163.4499	1348	971.11
CCSA	0.05	163.4950	1500	2344.60
Ratio	-	1.000	**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**

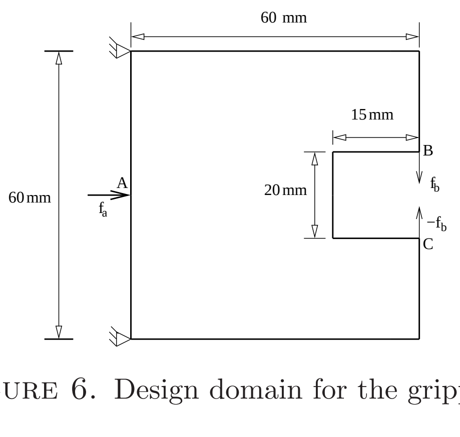

FIGURE 6. Design domain for the gripper.

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 $1\,mm^2$.

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**
Density filter ($r_{min} = 2.0$)				
SLP	0.20	-2.0760	683	459.72
CCSA	0.15	-0.3821	1444	5943.40
Ratio	-	**5.433**	**0.473**	**0.077**
Dilation filter ($r_{min} = 1.0$)				
SLP	0.15	-1.9921	1164	933.41
CCSA	0.05	-0.1284	1500	6258.20
Ratio	-	**15.515**	**0.776**	**0.149**
Erosion filter ($r_{min} = 1.0$)				
SLP	0.25	-3.0952	1328	1058.70
CCSA	0.05	-0.2641	1500	4837.40
Ratio	-	**11.720**	**0.885**	**0.219**
Sinh filter ($r_{min} = 1.5$)				
SLP	0.10	-4.2026	614	382.51
CCSA	0.10	-3.2741	2389	8224.40
Ratio	-	**1.284**	**0.257**	**0.047**

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

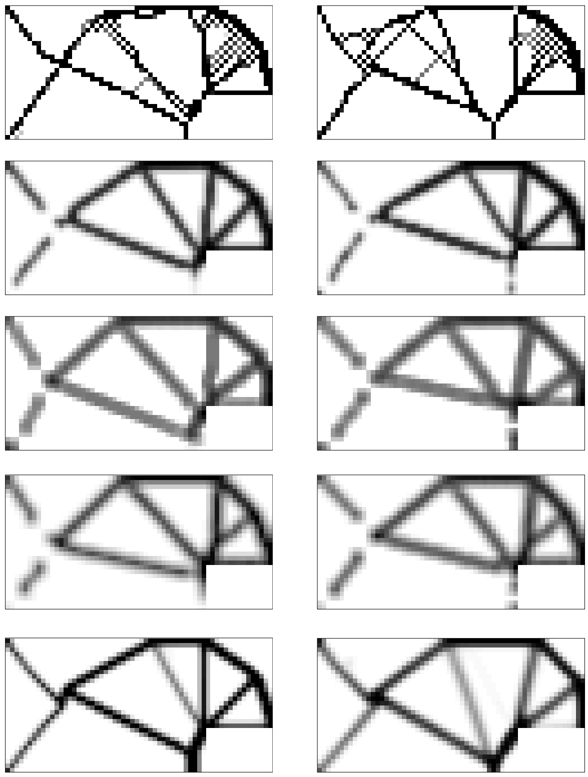

FIGURE 7. The grippers obtained using various filter and method combinations. The left column presents the topologies generated by the SLP method, while the right column present the topologies found by CCSA. The first pair of mechanisms was obtained without using a filter. The remaining pairs were generated using the density, the dilation, the erosion and the sinh filters, respectively.

Once again, the grippers shown in Figure 7 and the results presented in Table 3 suggest that the SLP method is better than CCSA. The SLP algorithm has obtained the best solution for all of the filters. Besides, 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 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 $1\,mm^2$.

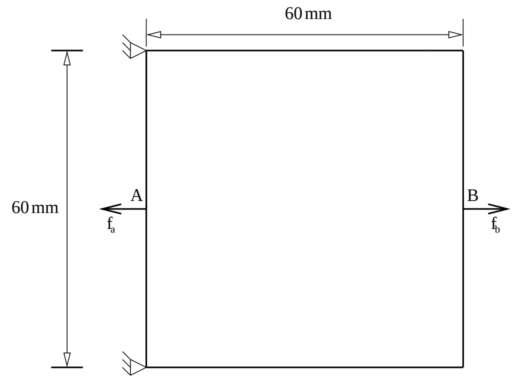

FIGURE 8. Design domain for the force inverter.

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.

According to Table 4, the SLP algorithms found the best solution for four types of filter. The CCSA method attained a better solution just for the Sinh filter. However, 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.

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**
Density filter ($r_{min} = 3.0$)				
SLP	0.05	0.0361	618	638.91
CCSA	0.10	0.1250	1205	4372.00
Ratio	-	**0.289**	**0.513**	**0.146**
Dilation filter ($r_{min} = 1.0$)				
SLP	0.10	0.0184	918	845.14
CCSA	0.20	0.1436	1463	6160.30
Ratio	-	**0.128**	**0.627**	**0.137**
Erosion filter ($r_{min} = 1.0$)				
SLP	0.05	-0.0619	902	840.34
CCSA	0.10	0.1051	1424	4517.10
Ratio	-	-	**0.633**	**0.186**
Sinh filter ($r_{min} = 1.5$)				
SLP	0.10	-4.7174	663	517.81
CCSA	0.05	-4.7698	534	1103.30
Ratio	-	0.989	1.242	**0.469**

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.

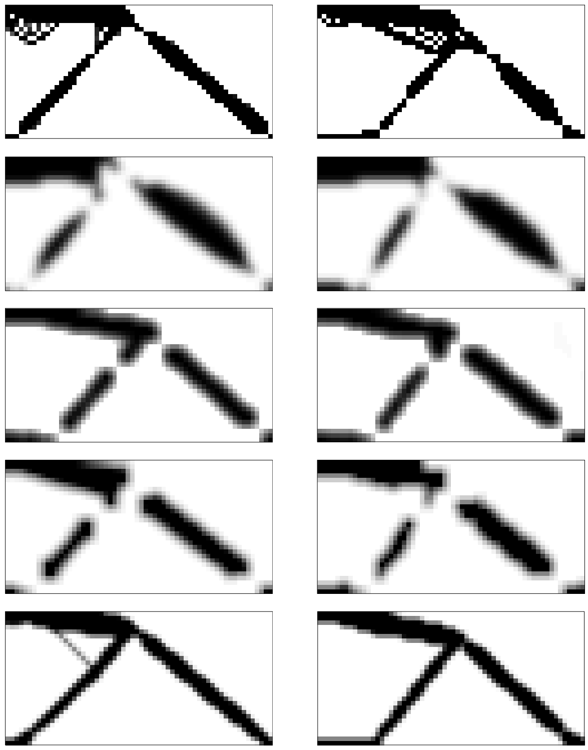

Figure 9. The force inverters obtained using various filter and method combinations. The left column presents the topologies generated by the SLP method, while the right column present the topologies found by CCSA. The first pair of mechanisms was obtained without using a filter. The remaining pairs were generated using the density, the dilation, the erosion and the sinh filters, respectively.

As we can observe, the filters have avoided the occurrence of checkerboards. However, some of them allowed the formation of one node hinges. The implementation of hinge elimination strategies, following the suggestions of Silva [18], 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. [15], Min and Kim [13], and Luo et al. [12].

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

References

[1] M.P. Bendsøe, *Optimal shape design as a material distribution problem*, Struct. Optim., **1** (1989), 193–202.

[2] M.P. Bendsøe and N. Kikuchi, *Generating optimal topologies in structural design using a homogenization method*, Comput. Methods Appl. Mech. Eng., **71** (1988), 197–224.

[3] T.E. Bruns and D.A. Tortorelli, *An element removal and reintroduction strategy for the topology optimization of structures and compliant mechanisms*, Int. J. Numer. Methods Eng., **57** (2003), 1413–1430.

[4] T.E. Bruns, *A reevaluation of the SIMP method with filtering and an alternative formulation for solid-void topology optimization*, Struct. Multidiscipl. Optim., **30** (2005), 428–436.

[5] R.H. Byrd, N.I.M Gould and J. Nocedal, *An active-set algorithm for nonlinear programming using linear programming and equality constrained subproblems*, Report OTC 2002/4, Optimization Technology Center, Northwestern University, Evanston, IL, USA, (2002).

[6] C.M. Chin and R. Fletcher, *On the global convergence of an SLP-filter algorithm that takes EQP steps*, Math. Program., **96** (2003), 161–177.

[7] A.R. Díaz and O. Sigmund, *Checkerboard patterns in layout optimization*, Struct. and Multidiscipl. Optim., **10** (1995), 40–45.

[8] R. Fletcher and E. Sainz de la Maza, *Nonlinear programming and nonsmooth optimization by sucessive linear programming*, Math. Program., **43** (1989), 235–256.

[9] F.A.M. Gomes, M.C. Maciel and J.M. Martínez, *Nonlinear programming algorithms using trust regions and augmented Lagrangians with nonmonotone penalty parameters*, Math. Program., **84** (1999), 161–200.

[10] N. Kikuchi, S. Nishiwaki, J.S.F. Ono and E.C.S. Silva, *Design optimization method for compliant mechanisms and material microstructure*, Comput. Methods Appl. Mech. Eng., **151** (1998), 401–417.

[11] C.R. Lima, *Design of compliant mechanisms using the topology optimization method* (in portuguese). Dissertation, University of São Paulo, SP, Brazil, (2002).

[12] Z. Luo, L. Chen, J. Yang, Y. Zhang and K. Abdel-Malek, *Compliant mechanism design using multi-objective topology optimization scheme of continuum structures*, Struct. Multidiscipl. Optim., **30** (2005), 142–154.

[13] S. Min and Y. Kim, *Topology Optimization of Compliant Mechanism with Geometrical Advantage*, JSME Int. J. Ser. C, **47**(2) (2004), 610–615.

[14] S. Nishiwaki, M.I. Frecker, M. Seungjae and N. Kikuchi, *Topology optimization of compliant mechanisms using the homogenization method*, Int. J. Numer. Methods Eng., **42** (1998), 535–559.

[15] C.B.W. Pedersen, T. Buhl and O. Sigmund, *Topology synthesis of large-displacement compliant mechanisms*, Int. J. Num. Methods Eng., **50** (2001), 2683–2705.

[16] O. Sigmund, *On the design of compliant mechanisms using topology optimization*, Mech. Struct. Mach., **25** (1997), 493–524.

[17] O. Sigmund, *Morphology-based black and white filters for topology optimization*, Struct. Multidiscipl. Optim., **33**(4-5) (2007), 401–424.

[18] M.C. Silva, *Topology optimization to design hinge-free compliant mechanisms* (in portuguese). Dissertation, University of São Paulo, SP, Brazil, (2007).

[19] K. Svanberg, *The Method of Moving Asymptotes - A new method for structural optimization*, Int. J. Numer. Methods Eng., **24** (1987), 359–373.

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

[21] Z. Zhang, N. Kim and L. Lasdon, *An improved sucessive linear programming algorithm*, Manag. Science **31** (1985), 1312–1351.