

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 / senne@ime.unicamp.br

Abstract. We introduce a globally convergent sequential linear programming method for nonlinear programming. The algorithm is applied to the solution of classic topology optimization problems, as well as to the design of compliant mechanisms. The numerical results suggest that the new algorithm is faster than the globally convergent version of the method of moving asymptotes, a popular method for mechanical engineering applications proposed by Svanberg.

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. 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".

#CAM-316/10. Received: 15/VIII/10. Accepted: 05/I/11.

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

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.

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. We devote Section 4 to our numerical experiments. Finally, Section 5 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

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

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

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

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

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

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

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 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. In this work, we adopt the formulation proposed by Nishiwaki et al. [14], although similar preliminary results were also obtained for the formulations of Sigmund [16] and Lima [11].

Considering that the mechanism belongs to a domain Ω , fixed at a region Γ_d of its boundary $\partial\Omega$, Nishiwaki et al. [14] suggest to decouple the problem into

two distinct load cases. In the first case, shown in Figure 1(a), a load \mathbf{t}^1 is applied to the region $\Gamma_{t1} \subset \partial\Omega$, and a fictitious load \mathbf{t}^2 is applied to a different region $\Gamma_{t2} \subset \partial\Omega$, where the deflection is supposed to occur. The optimal structure for this problem is obtained maximizing the mutual energy of the mechanism, subject to the equilibrium and volume constraints. This problem represents the kinematic behavior of the compliant mechanism.

After the mechanism deformation, the Γ_{t2} region eventually contacts a work-piece. In this case, the mechanism must be sufficiently rigid to resist the reaction force exerted by the work-piece 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_{t2} , and that there is no deflection at the region Γ_{t1} .

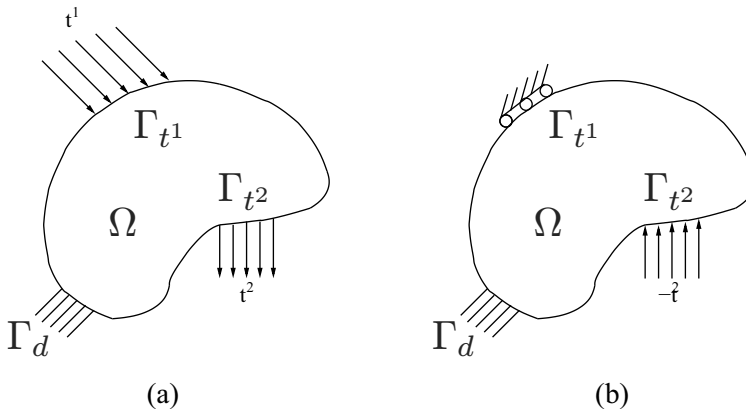


Figure 1 – The two load cases considered in the formulation of Nishiwaki et al. [14].

These load cases are combined into a single optimization problem. After discretization and variable elimination, this problem becomes

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

where \mathbf{f}_a and \mathbf{f}_b are the vectors of nodal forces related to the loads \mathbf{t}^1 and \mathbf{t}^2 shown in Figure 1(a), \mathbf{f}_c is the vector of nodal forces related to the load $-\mathbf{t}^2$ shown in Figure 1(b), and $\mathbf{K}_1(\rho)$ and $\mathbf{K}_2(\rho)$ are the stiffness matrices associated to the first and the second load cases, respectively.

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 to solve topology optimization problems, global convergence results are not fully established. On the other hand, SLP algorithms that are proved to be globally convergent are seldom adopted in practice. 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.

3.1 Description of the method

Consider the nonlinear programming problem

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{c}(\mathbf{x}) = \mathbf{0}, \\ & \mathbf{x} \in \mathbf{X}, \end{aligned} \tag{3}$$

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

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

and vectors $\mathbf{x}_l, \mathbf{x}_u \in \mathbb{R}^n$ define the lower and upper bounds for the components of \mathbf{x} . One should notice that, using slack variables, any nonlinear programming problem may be written in the form (3).

The linear approximations for the objective function and for the equality constraints of (3) in the neighborhood of a point $\mathbf{x} \in \mathbb{R}^n$, are given by

$$\begin{aligned} f(\mathbf{x} + \mathbf{s}) &\approx f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{s} \equiv L(\mathbf{x}, \mathbf{s}), \\ \mathbf{c}(\mathbf{x} + \mathbf{s}) &\approx \mathbf{c}(\mathbf{x}) + \mathbf{A}(\mathbf{x})\mathbf{s}, \end{aligned}$$

where $\mathbf{A}(\mathbf{x}) = [\nabla c_1(\mathbf{x}) \dots \nabla c_m(\mathbf{x})]^T$ is the Jacobian matrix of the constraints. Therefore, given a point \mathbf{x} , we can approximate (3) by the linear programming problem

$$\begin{aligned} \min_{\mathbf{s}} \quad & f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{s} \\ \text{s.t.} \quad & \mathbf{A}(\mathbf{x})\mathbf{s} + \mathbf{c}(\mathbf{x}) = \mathbf{0} \\ & \mathbf{x} + \mathbf{s} \in \mathbf{X}. \end{aligned} \quad (4)$$

A sequential linear programming (SLP) algorithm is an iterative method that generates and solves a sequence of linear problems in the form (4). At each iteration k of the algorithm, a previously computed point $\mathbf{x}^{(k)} \in \mathbf{X}$ is used to generate the linear programming problem. After finding \mathbf{s}_c , an approximate solution for (4), the variables of the original problem (3) are updated according to $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}_c$.

Unfortunately, this scheme has some pitfalls. First, problem (4) may be unlimited even if (3) has an optimal solution. Besides, the linear functions used to define (4) 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 (3). Including the trust region in (4), we get the problem

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

where $\mathbf{s}_l = \max\{-\delta_k, \mathbf{x}_l - \mathbf{x}^{(k)}\}$ and $\mathbf{s}_u = \min\{\delta_k, \mathbf{x}_u - \mathbf{x}^{(k)}\}$.

However, unless $\mathbf{x}^{(k)}$ satisfies the constraints of (3), it is still possible that problem (5) has no feasible solution. In this case, we need not only to improve $f(\mathbf{x}^{(k)} + \mathbf{s})$, but also to find a point that reduces this infeasibility. This can be done, for example, solving the problem

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

where $\mathbf{s}_n^l = \max\{-0.8\delta_k, \mathbf{x}_l - \mathbf{x}^{(k)}\}$, $\mathbf{s}_n^u = \min\{0.8\delta_k, \mathbf{x}_u - \mathbf{x}^{(k)}\}$. Clearly, $M(\mathbf{x}, \mathbf{s})$ is an approximation for the true measure of the infeasibility given by the function

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

Although the square of the Euclidean norm is the usual choice for defining φ (see [9]), due to its differentiability, the one-norm is more appropriate when dealing with an SLP algorithm. Besides avoiding the use of a quadratic function, the one-norm allows the replacement of (6) by the equivalent linear programming problem

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

where $\mathbf{z} \in \mathbb{R}^{m_I}$ is a vector of slack variables corresponding to the infeasible constraints, and $\mathbf{e}^T = [1 \ 1 \ \dots \ 1]$. To see how matrix $\mathbf{E}(\mathbf{x}^{(k)})$ 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 $\mathbf{c}(\mathbf{x}^{(k)})$. In this case, the j -th column of $\mathbf{E}(\mathbf{x}^{(k)})$ is given by

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

A basic feasible point for (7) can be obtained taking $\mathbf{s} = \mathbf{0}$ and $z_j = |c_{i_j}(\mathbf{x}^{(k)})|$, $j = 1, \dots, m_I$.

One should notice that the trust region used in (6) and (7) is slightly smaller than 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 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 $f(\mathbf{x}^{(k)} + \mathbf{s}_c) \ll f(\mathbf{x}^{(k)})$ and $\varphi(\mathbf{x}^{(k)} + \mathbf{s}_c) \ll \varphi(\mathbf{x}^{(k)})$, it is clear that $\mathbf{x}^{(k)} + \mathbf{s}_c$ is a better approximation than $\mathbf{x}^{(k)}$ for the optimal solution of problem (3). 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

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

where $\theta \in (0, 1]$ is a penalty parameter used to balance the roles of f and φ . 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

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

is the actual reduction of the objective function, and

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

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

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

is the predicted reduction of f and

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

is the predicted reduction of the infeasibility.

At the k -th iteration of the algorithm, the step \mathbf{s}_c is accepted if $A_{red} \geq 0.1P_{red}$. If this condition is not verified, δ_k 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. Following a suggestion given by Gomes et al. [9], at the beginning of the k -th iteration, we define

$$\theta_k = \min\{\theta_k^{large}, \theta_k^{sup}\}, \quad (9)$$

where

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

$$\theta_k^{min} = \min\{1, \theta_0, \dots, \theta_{k-1}\}, \quad (11)$$

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

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 (3) is given by Algorithm 1.

In the next subsections we prove that this algorithm is well defined and converges to the solution of (3) under mild conditions. In Section 4, we describe a particular implementation of this SLP method for solving the topology optimization problem.

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 $\min \varphi(\mathbf{x})$. Besides, a point $\mathbf{x} \in \mathbf{X}$

that satisfies $\varphi(\mathbf{x}) = 0$ is said to be *regular* if the gradient vectors of the active constraints at \mathbf{x} are linearly independent (i.e. the linear independence constraint qualification holds at \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 (3), 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 $(\mathbf{s}_0, \mathbf{z}_0) = (\mathbf{0}, -\mathbf{E}(\mathbf{x}^{(k)})^T \mathbf{c}(\mathbf{x}^{(k)}))$ as the feasible (yet not basic) initial point for the restoration problem (7), solved at step 2 of Algorithm 1. Define also

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

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

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

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, $\nabla \bar{M}(\mathbf{x}, \mathbf{s}, \mathbf{z})$ does 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

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

where

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

Algorithm 3.1 General SLP algorithm.

```

1: while a stopping criterion is not satisfied, do
2:   Determine  $\mathbf{s}_n$ , the solution of (7)
3:   if  $\bar{M}(\mathbf{x}^{(k)}, \mathbf{s}_n, \mathbf{z}) = 0$ , then
4:     Starting from  $\mathbf{s}_n$ , determine  $\mathbf{s}_c$ , the solution of (5)
5:   else
6:      $\mathbf{s}_c \leftarrow \mathbf{s}_n$ .
7:   end if
8:   Determine  $\theta_k = \min\{\theta_k^{large}, \theta_k^{sup}, \theta^{max}\}$ 
9:   if  $A_{red} \geq 0.1P_{red}$  then
10:     $\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} + \mathbf{s}_c$ 
11:    if  $A_{red} \geq 0.5P_{red}$ , then
12:       $\delta_{k+1} \leftarrow \min\{2.5\delta_k, \|\mathbf{x}_u - \mathbf{x}_l\|_\infty\}$ 
13:    else
14:       $\delta_{k+1} \leftarrow \delta_{min}$ 
15:    end if
16:    Recompute  $\mathbf{A}$ ,  $\mathbf{E}$  and  $\nabla f$ .
17:     $\theta_{max} \leftarrow 1$ 
18:     $k \leftarrow k + 1$ 
19:  else
20:     $\delta_k \leftarrow \max\{0.25\|\mathbf{s}_c\|_\infty, 0.1\delta_k\}$ 
21:     $\theta_{max} \leftarrow \theta_k$ 
22:  end if
23: end while

```

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

$$P_{red}^{fsb} \geq 0.8\eta\delta_k. \quad (16)$$

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

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

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

$$P_{red} \geq 0.4\eta\delta_k. \quad (18)$$

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

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

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

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

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

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. \square

Lemma 3.3. *Suppose that $\mathbf{x}^{(k)}$ is feasible and regular for (3), 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 (3), 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

$$\begin{aligned} \min \quad & \alpha \nabla f(\mathbf{x}^{(k)})^T \mathbf{d}_t \\ \text{s.t.} \quad & \|\alpha \mathbf{d}_t\|_\infty \leq \delta_k \\ & \alpha \geq 0. \end{aligned}$$

Since this 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 $\|\bar{\alpha} \mathbf{d}_t\|_\infty = \delta_k$, implying 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

$$\begin{aligned} L(\mathbf{x}^{(k)}, \mathbf{0}) - L(\mathbf{x}^{(k)}, \bar{\alpha} \mathbf{d}_t) &= -\bar{\alpha} \nabla f(\mathbf{x}^{(k)})^T \mathbf{d}_t \\ &= -\frac{\delta_k}{\|\mathbf{d}_t\|_\infty} \nabla f(\mathbf{x}^{(k)})^T \mathbf{d}_t = \eta \delta_k. \end{aligned} \quad (20)$$

Combining (20) with 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

$$P_{red} = \theta_k P_{red}^{opt} \geq \theta_k \eta \delta_k. \quad (21)$$

Since (17) also applies in this case, we can combine it with (21) to obtain (19). Therefore, for δ_k sufficiently small, $A_{red} \geq 0.1 P_{red}$ and the step is accepted. \square

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 (3); 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 $\varepsilon_1, \alpha_1, \alpha_2 > 0$ such that, if Algorithm 1 is applied to $\mathbf{x} \in \mathbf{X}$ and $\|\mathbf{x} - \mathbf{x}^*\| \leq \varepsilon_1$, 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 point 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, for a fixed vector \mathbf{x} near \mathbf{x}^* , we can consider the auxiliary linear programming problem

$$\begin{aligned} \min \quad & \tilde{M}(\mathbf{x}, \mathbf{s}, \mathbf{z}) = \mathbf{e}^T \mathbf{z} \\ \text{s.t.} \quad & \mathbf{A}(\mathbf{x})\mathbf{s} + \mathbf{E}(\mathbf{x}^*)\mathbf{z} = -\tilde{\mathbf{c}}(\mathbf{x}) \\ & \mathbf{s}_n^l \leq \mathbf{s} \leq \mathbf{s}_n^u \\ & \mathbf{z} \geq \mathbf{0}, \end{aligned} \tag{22}$$

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 point.

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 (22), 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

$$P_{red} \geq 0.4 \frac{(-\nabla \tilde{M}(\mathbf{x})^T \tilde{\mathbf{d}}_n(\mathbf{x}))}{\|\tilde{\mathbf{d}}_n(\mathbf{x})\|_\infty} \delta \geq 0.1 \frac{(-\nabla \tilde{M}(\mathbf{x}^*)^T \mathbf{d}_n(\mathbf{x}^*))}{\|\mathbf{d}_n(\mathbf{x}^*)\|_\infty} \delta. \quad (23)$$

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

$$\begin{aligned} P_{red} &\geq 0.5 P_{red}^{fsb} \geq -0.5 \nabla \tilde{M}(\mathbf{x})^T \tilde{\mathbf{d}}_n(\mathbf{x}) \\ &\geq -0.25 \nabla \tilde{M}(\mathbf{x}^*)^T \mathbf{d}_n(\mathbf{x}^*). \end{aligned} \quad (24)$$

Finally, let us suppose that, after solving (22), 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 (24) also holds. The desired result follows from (23) and (24), for an appropriate choice of α_1 and α_2 . \square

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 $\{\delta_k \mid k \in K_1\}$ 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

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

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

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 (3).

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

Theorem 3.1. 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

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

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

$$\psi_k \leq \frac{cN}{(k+1)^{1.1}}\theta_{k-1} + \psi_{k-1} - A_{red}^{(k-1)}. \quad (25)$$

Noting that $\theta_k \in [0, 1]$ for all k , and applying (25) 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. \square

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 (3).

Lemma 3.6. *For each feasible and regular point \mathbf{x}^* there exists $\epsilon_0, \hat{\sigma} > 0$ 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 auxiliary problem

$$\begin{aligned} \min \quad & \bar{M}(\mathbf{x}, \mathbf{s}, \mathbf{z}) = \mathbf{e}^T \mathbf{z} \\ \text{s.t.} \quad & \mathbf{A}(\mathbf{x})\mathbf{s} + \mathbf{E}(\mathbf{x})\mathbf{z} = -\mathbf{c}(\mathbf{x}) \\ & \mathbf{x}_l - \mathbf{x} \leq \mathbf{s} \leq \mathbf{x}_u - \mathbf{x} \\ & \mathbf{z} \geq \mathbf{0}. \end{aligned} \tag{26}$$

has an optimal solution $(\mathbf{s}, \mathbf{z}) = (\hat{\mathbf{s}}, \mathbf{0})$. In this case, $\mathbf{A}(\mathbf{x})\hat{\mathbf{s}} = -\mathbf{c}(\mathbf{x})$, so $\|\hat{\mathbf{s}}\|_2 \leq \|\mathbf{c}(\mathbf{x})\|_2/\hat{\sigma}$, where $\hat{\sigma} > 0$ is the smallest singular value of $\mathbf{A}(\mathbf{x})$. Since problem (26) is just (7) without the trust region constraint $\|\mathbf{s}\|_\infty \leq 0.8\delta$, we have

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

proving the first part of the lemma.

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

$$M(\mathbf{x}, \mathbf{0}) - M(\mathbf{x}, \mathbf{s}_n(\mathbf{x}, \delta)) = M(\mathbf{x}, \mathbf{0}) = \|\mathbf{c}(\mathbf{x})\|_1. \tag{27}$$

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,

$$M(\mathbf{x}, \mathbf{0}) - M(\mathbf{x}, \mathbf{s}_n(\mathbf{x}, \delta)) = \delta\|\mathbf{c}(\mathbf{x})\|_1/\|\hat{\mathbf{s}}\|_\infty \geq \hat{\sigma}\delta. \tag{28}$$

The second part of the lemma follows from (27) and (28). \square

Lemma 3.7. *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 (3). 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. As 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

$$\begin{aligned} \min \quad & L(\mathbf{x}, \mathbf{s}_n + \mathbf{s}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T (\mathbf{s}_n + \mathbf{s}) \\ \text{s.t.} \quad & \mathbf{s} = t\mathbf{d}_t, \quad t \geq 0 \\ & \|\mathbf{s}_n + \mathbf{s}\|_\infty \leq \delta \\ & \mathbf{x}_l \leq \mathbf{x} + \mathbf{s}_n + \mathbf{s} \leq \mathbf{x}_u \end{aligned} \tag{29}$$

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

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

where

$$\begin{aligned} \bar{t} &= \min\{1, \Delta_1, \Delta_2\}, \\ \Delta_1 &= \min_{d_{t_i} < 0} \left\{ \frac{\delta + s_{n_i}}{-d_{t_i}}, \frac{x_i + s_{n_i} - x_{l_i}}{-d_{t_i}} \right\}, \\ \Delta_2 &= \min_{d_{t_i} > 0} \left\{ \frac{\delta - s_{n_i}}{d_{t_i}}, \frac{x_{u_i} - x_i - s_{n_i}}{d_{t_i}} \right\}. \end{aligned}$$

Since (29) 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

$$\bar{t} = \min \left\{ 1, \min_{d_{t_i} < 0} \left\{ \frac{\delta + s_{n_i}}{-d_{t_i}} \right\}, \min_{d_{t_i} > 0} \left\{ \frac{\delta - s_{n_i}}{d_{t_i}} \right\} \right\}. \tag{30}$$

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

$$L(\mathbf{s}_n) - L(\mathbf{s}_c) \geq L(\mathbf{s}_n) - L(\mathbf{s}_n + \mathbf{s}_t^d) = -\bar{t} \nabla f(\mathbf{x})^T \mathbf{d}_t. \tag{31}$$

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 $\mathbf{x} \in \{\mathbf{x}^{(k)} \mid k \in K_1, k \geq k_1\}$,

$$\|\mathbf{d}_t\|_\infty \leq c'_1, \quad (32)$$

$$-\nabla f(\mathbf{x})^T \mathbf{d}_t \geq c'_2. \quad (33)$$

From (30) 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 (32) we obtain

$$\bar{t} \geq \min \left\{ 1, \frac{0.2\delta}{c'_1} \right\} = \frac{0.2}{c'_1} \min \left\{ \frac{c'_1}{0.2}, \delta \right\}. \quad (34)$$

Combining (31), (33) and (34), we get, for all $x \in \{\mathbf{x}^{(k)} \mid k \in K_1, k \geq k_0\}$,

$$L(\mathbf{s}_n) - L(\mathbf{s}_c) \geq \frac{0.2c'_2}{c'_1} \min \left\{ \frac{c'_1}{0.2}, \delta \right\}.$$

The desired result is obtained taking $c_1 = \frac{0.2c'_2}{c'_1}$ and $\delta' = \frac{c'_1}{0.2}$. \square

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 (3). 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 $\theta^{\text{sup}}(\mathbf{x}, \delta) = 1$, where θ^{sup} is given by (12) and δ' is defined in Lemma 3.7.

Proof. From Lemma 3.6, 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 3.7 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. \square

Lemma 3.9. *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 (3). Then*

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

Proof. The sequences $\{\theta_k^{\min}\}$ and $\{\theta_k^{\text{large}}\}$ are bounded below and nonincreasing, so both are convergent. Moreover, they converge to the same limit, as $\lim_{k \rightarrow \infty} (\theta_k^{\text{large}} - \theta_k^{\min}) = 0$. Besides, $\theta_{k+1}^{\min} \leq \theta_k \leq \theta_k^{\text{large}}$. 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 3.7, 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

$$|A_{red} - P_{red}| = O(\delta^2). \quad (35)$$

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

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

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

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

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

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

Therefore, for all $\delta \in [\tilde{\delta}/10, \tilde{\delta})$ and all $\mathbf{x} \in \{\mathbf{x}^{(k)} \mid k \in K_1, k \geq k'_3\}$, we have

$$\frac{|A_{red} - P_{red}|}{P_{red}} \leq 0.5, \quad (36)$$

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

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

so (36) also applies.

Thus, for some $\delta \in [\tilde{\delta}/10, \tilde{\delta})$, 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 $A_{red} \geq 0.1P_{red}$, 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. \square

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 (3). 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*

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

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

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

From Lemma 3.6, if $\mathbf{x} \in \{\mathbf{x}^{(k)} \mid k \in K_1, k \geq k_2\}$, we have that

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

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

Lemma 3.11. *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 (3). Then there exist $k_4 > 0$, $\tilde{\theta} \in (0, 1]$ such that, if*

$$\mathbf{x} \in \{\mathbf{x}^{(k)} \mid k \in K_1, k \geq k_4\}, \|\mathbf{c}(\mathbf{x})\|_1 \geq \beta\delta \quad \text{and} \quad \theta \leq \tilde{\theta}$$

satisfies (9)–(12), then $A_{red} \geq 0.1P_{red}$.

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

$$|A_{red} - P_{red}| = \delta O(\|\mathbf{c}(\mathbf{x})\|_1). \quad (37)$$

Since our choice of θ ensures that $P_{red} \geq 0.5[M(\mathbf{0}) - M(\mathbf{s}_c)]$, Lemma 3.6 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 (37) by P_{red} , we get

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

which yields the desired result. \square

Lemma 3.12. *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 (3).*

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 (3). Then, from Lemma 3.9, we have that $\lim_{k \leftarrow \infty} \theta_k = 0$.

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

$$\lim_{k \in K_2} \theta_k^{sup}(\underline{\delta}_k) = 0, \quad (38)$$

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 $\theta_k^{large} \leq 2\theta_k^{min}$,

$$\theta_k^{sup}(\underline{\delta}_k) \leq \tilde{\theta}/2 < 1, \quad \text{and} \quad \theta_k \leq \tilde{\theta}/2. \quad (39)$$

Lemma 3.8 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 (38) and (39),

$$\|\mathbf{c}(\mathbf{x}^{(k)})\|_1 > \beta\underline{\delta}_k \quad (40)$$

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

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

Assume, without loss of generality, that

$$\underline{\delta}_k \leq 0.1\delta' < 0.1\delta_{\min} \tag{41}$$

for all $k \in K_2$, where δ' is defined in Lemma 3.7. 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 (39) 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 3.11 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 (41) implies that

$$\hat{\delta}_k \leq 10\underline{\delta}_k \leq \delta' < \delta_{\min}. \tag{42}$$

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 (38) and Lemma 3.8, we have

$$\begin{aligned} \hat{\theta}_k &= \min\{\theta'_k(\hat{\delta}_k), \theta_k^{\sup}(\hat{\delta}_k)\} = \theta'_k(\hat{\delta}_k) \\ &\geq \min\{\theta'_k(\hat{\delta}_k), \theta_k^{\sup}(\underline{\delta}_k)\} = \theta_k^{\sup}(\underline{\delta}_k) \end{aligned} \tag{43}$$

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

Since $\nabla f(\mathbf{x})$ 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 3.8, (42) and the definition of P_{red} , we have

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

so

$$\frac{|A_{red}(\hat{\theta}_k, \hat{\delta}_k) - P_{red}(\hat{\theta}_k, \hat{\delta}_k)|}{P_{red}(\hat{\theta}_k, \hat{\delta}_k)} = \frac{O(\hat{\delta}^2)}{\hat{\theta}_k c_2 \hat{\delta}_k} = \frac{O(\hat{\delta})}{\hat{\theta}_k}. \tag{44}$$

From Lemma 3.10 and (40), we obtain

$$\underline{\delta}_k / \theta_k^{\sup}(\underline{\delta}_k) = O(\|\mathbf{c}(\mathbf{x}^{(k)})\|_1) \quad \text{for } k \in K_2, k \geq k_5.$$

So, by (42) and (43), 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 (44) 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. \square

Theorem 3.2. *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 (3). 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 (3).*

Proof. This result follows from Theorem 3.1 and Lemma 3.12. □

4 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.

4.1 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 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.

4.2 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 in the literature.

In our experiments, three different schemes were considered, in order to see how they affect the performance of the algorithms. The first one was the density filter proposed by Bruns and Tortorelli [3]. For each element i , this filter replaces ρ_i by a weighted mean of the densities of the elements belonging to a ball B_i with radius r_{\max} .

Other filter we have tested was the Sinh method of Bruns [4], that 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. This filter was chosen because it turns the volume constraint into a nonlinear inequality constraint, so the problem is more difficult to solve.

Finally, we also tried the dilation filter introduced by Sigmund [17]. This filter replaces the density of an element i by the maximum of the densities of the elements that belong to the neighborhood B_i . This filter also turn the volume constraint into a nonlinear constraint, so the related problems are more challenging.

4.3 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].

At every iteration, the CCSA method approximates the objective and constraint functions by convex separable functions and solve the resulting subproblem. This inner iteration is repeated until its objective and constraint func-

tions become greater than or equal to the original functions at the optimal solution of the subproblem. A parameter vector, σ , is used to define upper and lower limits for the variables, as well as to convexify the objective and constraint functions.

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.

We also analyze the effect of the application of the filters presented in Section 4.2, to reduce the formation of checkerboard patterns in the structures. For the density, the dilation and the erosion filters, the penalty parameter p of the SIMP method is set to 1, 2 and 3, consecutively. The Sinh method uses a similar parameter p , that is set to 1 to 6, consecutively.

When the SIMP method is used and $p = 1$ or 2, the SLP and the CCSA algorithms stop whenever Δf , the difference between the objective function of two consecutive iterations, falls below 10^{-3} . For $p = 3$, both algorithms are halted when $\Delta f < 10^{-3}$ for three successive iterations. For the sinh method, if $p = 1, 2$ or 3, we stop the algorithms whenever Δf is smaller than 10^{-3} . If $p = 4, 5$ or 6, we require that $\Delta f < 10^{-3}$ for three successive iterations. Besides, for the density filter, we also define a limit of 1000 iterations for each value of the penalty parameter p used by the SIMP method. When the dilation filter is used, this limit is increased to 2800 iterations. For the Sinh filter, a limit of 500 iterations for each p was adopted. Although not technically sound, this stopping criterion based on the function improvement is quite common in topology optimization.

The initial trust region radius used by the SLP algorithm was set to 0.1. For the CCSA method, the stopping tolerance for the subproblems was set to 10^{-5} . A limit of 20 iterations was also defined for each subproblem. The components of the initial vector σ_0 were set to 0.1.

The tests were performed on a personal computer, with an Intel Pentium T4200 processor (2.0GHz, 4GB RAM), under the Windows Vista operating system. The algorithms were implemented in Matlab.

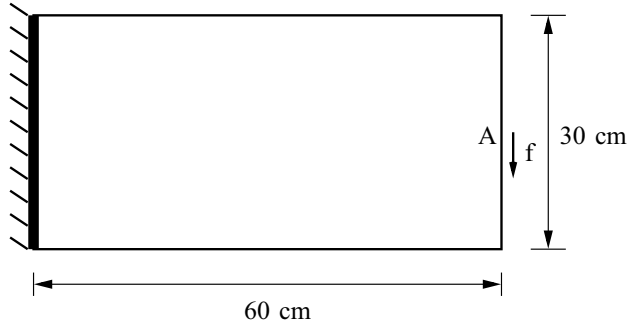


Figure 2 – Design domain for the cantilever beam.

4.4 Cantilever beam design

The first problem we consider is the cantilever beam presented in Figure 2. We suppose that the structure's thickness is $e = 1\text{ cm}$, the Poisson's coefficient is $\sigma = 0.3$ and the Young's modulus of the material is $E = 1\text{ N/cm}^2$. The volume of the optimal structure is limited by 40% of the design domain. A force $f = 1\text{ N}$ is applied downwards in the center of the right edge of the beam. The domain was discretized into 1800 square elements with 1 mm^2 each. The radius of the filters, r_{\max} , was set to 2.5.

The optimal structures for all of the combinations of methods and filters are shown in Figure 3. Table 1 contains 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 cantilever beams shown in Figure 3 are quite similar, suggesting that all of the filters efficiently reduced the formation of checkerboard patterns, as expected. On the other hand, Table 1 shows 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.

4.5 MBB beam design

The second problem we consider is the MBB beam presented in Figure 4. The structure's thickness, the Young's modulus of the material and the Poisson's

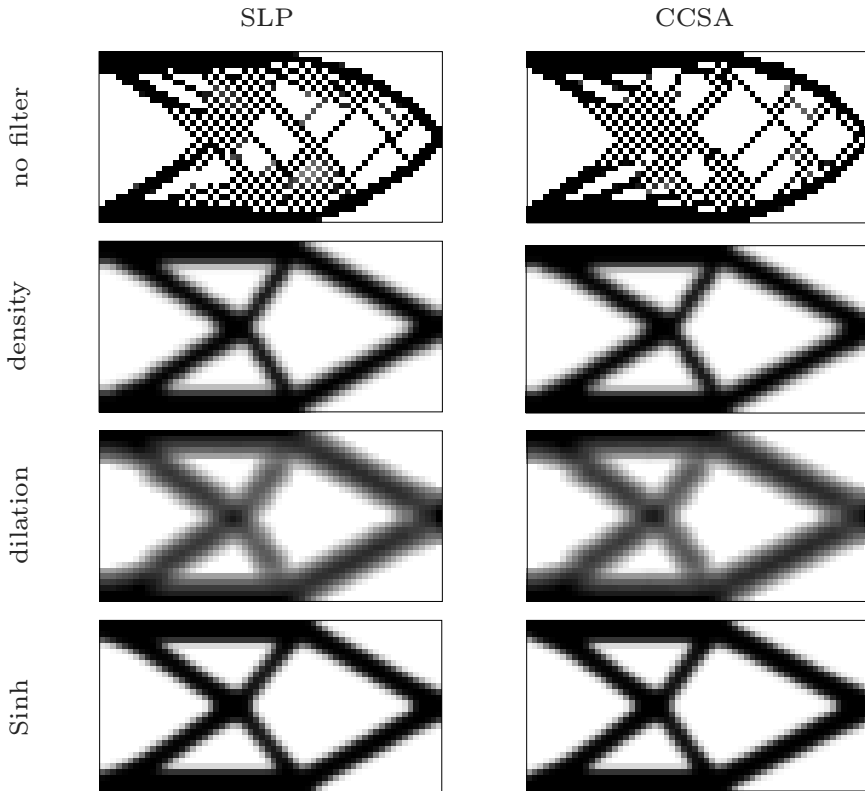


Figure 3 – The cantilever beams obtained.

Filter	Method	Objective	Iterations	Time (s)
None	SLP	75.978	243	51.20
	CCSA	77.904	715	212.87
	Ratio	0.975	0.34	0.241
Density	SLP	72.973	353	95.45
	CCSA	72.960	1172	457.85
	Ratio	1.000	0.301	0.208
Dilation	SLP	73.320	563	650.88
	CCSA	73.569	5542	6679.40
	Ratio	0.997	0.102	0.097
Sinh	SLP	73.784	680	234.52
	CCSA	74.297	2317	1796.20
	Ratio	0.993	0.293	0.131

Table 1 – Results for the cantilever beam.

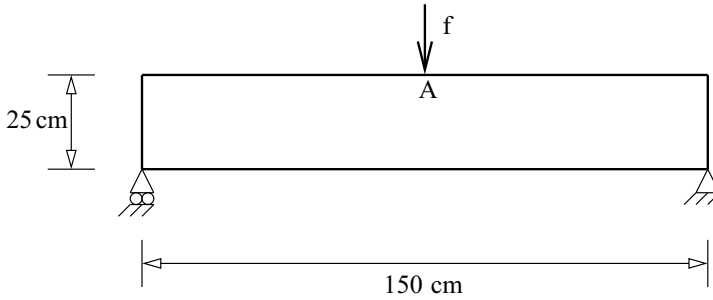


Figure 4 – Design domain for the MBB beam.

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 $f = 1\text{ N}$ is applied downwards in the center of the top edge of the beam. The radius of the filters, r_{\max} , was set to 2.5.

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.

Again, the structures obtained by both methods are similar. The same happens to the values of the objective function, as expected. However, the structure obtained using the density filter has some extra bars. Table 2 shows that the SLP algorithm performs much better than the CCSA method for the MBB beam.

4.6 Gripper mechanism design

Our third problem is the design of a gripper, whose domain is presented in Figure 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. We consider that the structure's thickness is $e = 1\text{ mm}$, the Young's modulus of the material is $E = 210000\text{ N/mm}^2$ 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 $f_a = f_b = 1\text{ N}$. The domain was discretized into 3300 square elements with 1 mm^2 . The filter radius was set to 1.5.

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

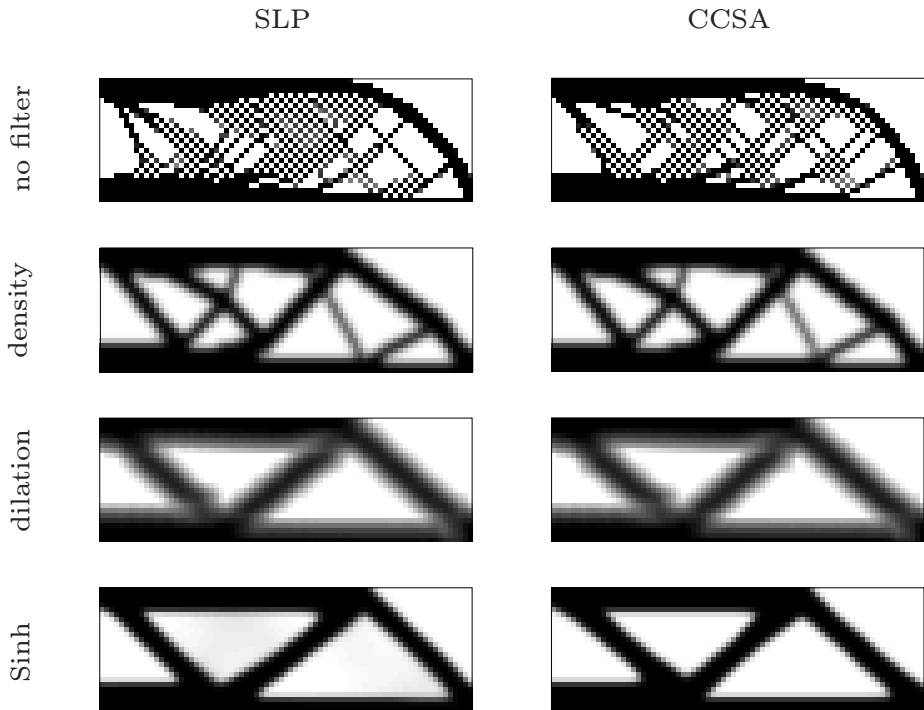


Figure 5 – The MBB beams obtained.

Filter	Method	Objective	Iterations	Time (s)
None	SLP	181.524	173	41.29
	CCSA	181.020	613	346.57
	Ratio	1.003	0.282	0.119
Density	SLP	180.618	1054	224.51
	CCSA	180.380	2516	2203.50
	Ratio	1.001	0.419	0.102
Dilation	SLP	185.472	752	896.03
	CCSA	185.325	5698	7650.40
	Ratio	1.001	0.132	0.117
Sinh	SLP	181.453	857	285.487
	CCSA	186.601	2694	2620.10
	Ratio	0.972	0.318	0.109

Table 2 – Results for the MBB beam.

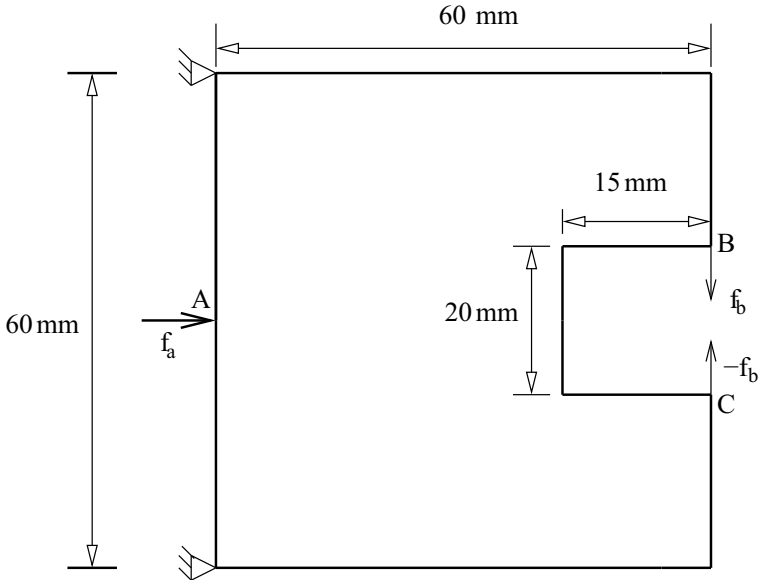


Figure 6 – Design domain for the gripper.

Filter	Method	Objective	Iterations	Time (s)
None	SLP	-5.871	94	46.07
	CCSA	-7.257	527	2139.50
	Ratio	0.809	0.178	0.022
Density	SLP	-2.924	358	136.30
	CCSA	-3.669	1302	4034.10
	Ratio	0.797	0.275	0.034
Dilation	SLP	-2.997	847	520.96
	CCSA	-2.834	2488	12899.00
	Ratio	1.057	0.340	0.040
Sinh	SLP	-1.240	632	255.86
	CCSA	-1.065	2487	14077.00
	Ratio	1.164	0.254	0.018

Table 3 – Results for the gripper mechanism.

Once again, the results presented in Table 3 suggest that the SLP method is better than CCSA, since the SLP algorithm spent much less time to obtain the optimal solution in all cases.

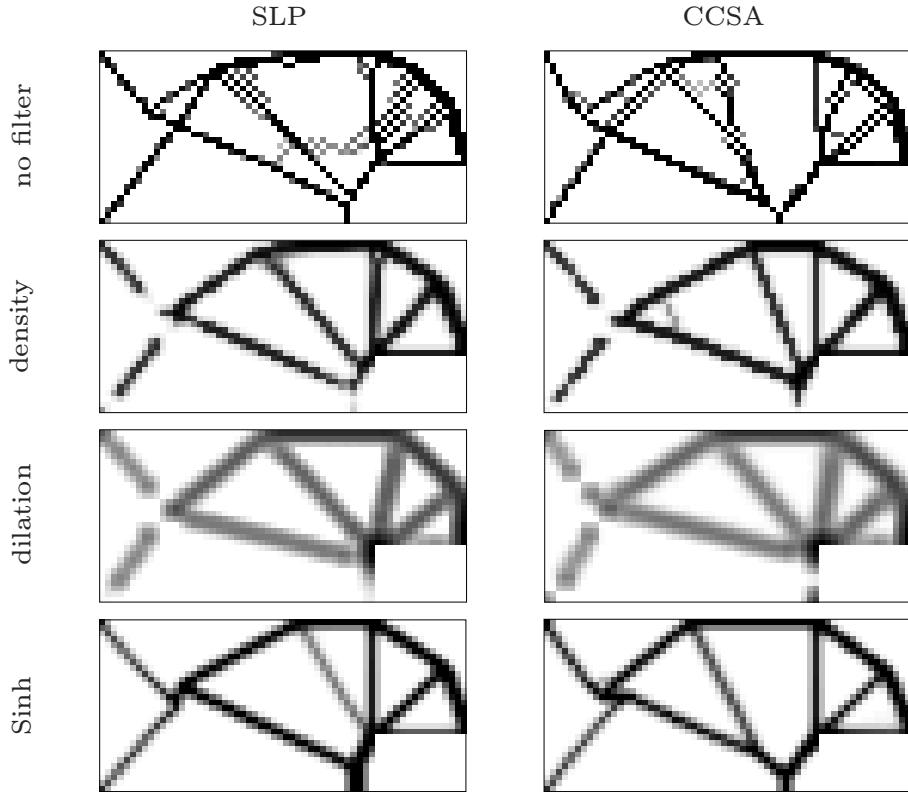


Figure 7 – The grippers obtained.

4.7 Force inverter design

Our last problem is the design of a compliant mechanism known as force inverter. The domain is shown in Figure 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 $e = 1 \text{ mm}$, $\sigma = 0.3$ and $E = 210000 \text{ N/mm}^2$. The volume is limited by 20% of the design domain, and the input and output forces are given by $f_a = f_b = 1 \text{ N}$. The domain was discretized into 3600 square elements with 1 mm^2 . The filter radius was set to 2.5.

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.

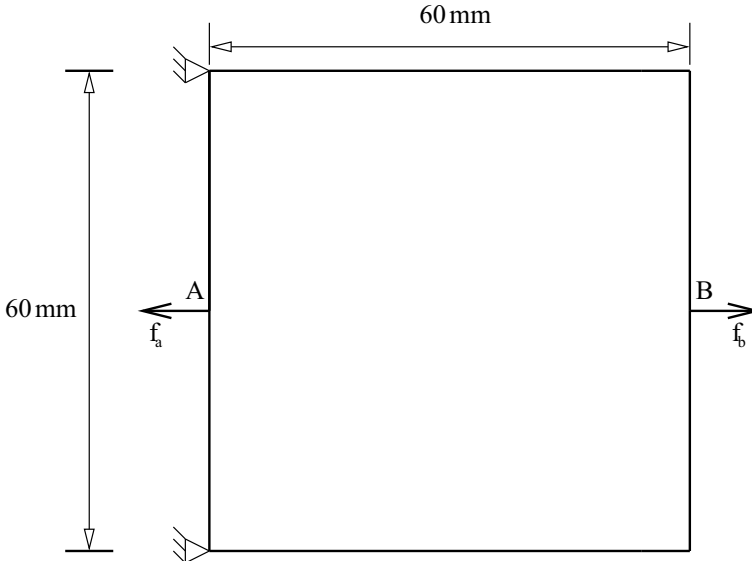


Figure 8 – Design domain for the force inverter.

Filter	Method	Objective	Iterations	Time (s)
None	SLP	-4.006	118	46.49
	CCSA	-4.223	443	485.54
	Ratio	0.949	0.266	0.096
Density	SLP	-1.380	281	131.63
	CCSA	-1.378	2514	3556.50
	Ratio	1.002	0.112	0.037
Dilation	SLP	-1.084	668	916.84
	CCSA	-1.014	5741	2679.80
	Ratio	1.069	0.116	0.342
Sinh	SLP	-0.574	512	248.16
	CCSA	-0.286	820	2019.30
	Ratio	2.007	0.624	0.123

Table 4 – Results for the force inverter.

The structures obtained by the algorithms are fairly similar, with the exception of the Sinh filter. According to Table 4, the SLP algorithms found the best solution for all of the filters. The CCSA method attained the best solution just when no filter was used. As in the previous examples, the SLP method took much less time to converge than the CCSA algorithm.

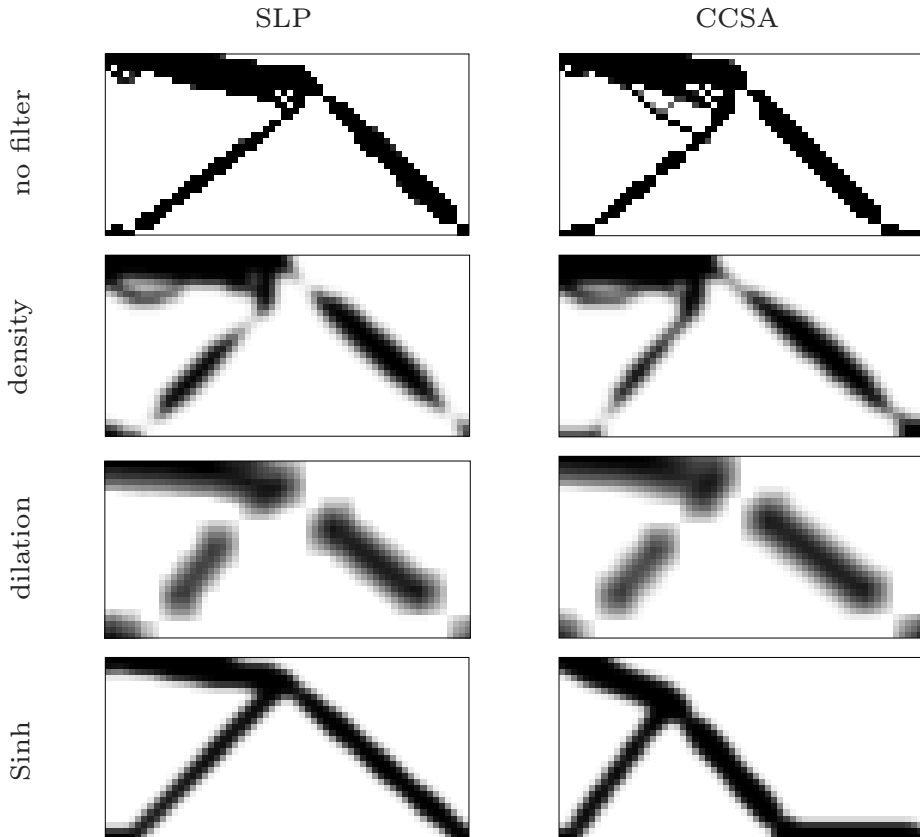


Figure 9 – The force inverters obtained.

5 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 are comparable. All of the filters work well in combination with the SLP algorithm to avoid the occurrence of checkerboards, although the dilation filter seems to be more expensive than the others. The nonlinearity introduced by the dilation and the Sinh filters poses no difficulty for the SLP method. For the combination of

filter and trust region radii tested, the Sinh filter presented the best formed structures. Some of the filters 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 successive 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 successive linear programming algorithm*. Manag. Science **31** (1985), 1312–1351.