Using Optimization to Solve Truss Topology Design Problems

Fernando Bastos * Adelaide Cerveira † Joaquim Gromicho ‡

* Departamento de Estatística e Investigação Operacional, FC, UL,
Lisboa, Portugal
fbastos@fc.ul.pt

† Departamento de Matemática, UTAD,
Vila Real, Portugal
cerveira@utad.pt

† Vrije Universiteit, Amsterdam & ORTEC International,
Gouda, The Netherlands
jgromicho@ortec.nl

Abstract

The design of truss structures is an important engineering activity which has traditionally been done without optimization support. Nowadays we witness an increasing concern for efficiency and therefore engineers seek aid on Mathematical Programming to optimize a design. In this article, we consider a mathematical model where we maximize the stiffness with a volume constraint and bounds in the cross sectional area of the bars, [2]. The basic model is a large-scale non-convex constrained optimization problem but two equivalent problems are considered. One of them is a minimization of a convex non-smooth function in several variables (much less than in the basic model), being only one non-negative. The other is a semidefinite programming problem. We solve some instances using both alternatives and we present and compare the results.

Keywords: truss topology design, stiffness, non-smooth convex programming, descent method, semidefinite programming, duality, interior point methods

Introduction

Truss topology design (TTD) deals with constructions like bridges, cantilevers and roof trusses supporting different loading scenarios. For example, a bridge should withstand forces corresponding to morning or evening rush hour traffic and even to an earthquake.

The selection of an optimal configuration for the structure depends on the used criteria, see for instance Refs. [3, 4, 17, 26, 19, 20]. Possible criteria are, for example, characteristics of rigidity such as stiffness and stability of the construction, the total amount of material used, structure lifetime, etc. In this paper we examine the issue of the stiffness of the truss for a given amount of material: we seek the stiffest truss satisfying equilibrium constraints and restrictions on the cross sectional area of the bars. This results in a large-scale non-convex problem, as we show with some detail.

An equivalent convex minimization problem is presented and solved by a nonsmooth steepest descent algorithm. This approach is unable to handle large TTD problems with tens of nodes and hundreds of bars, [2]. A more efficient alternative reformulation of the basic model as a semidefinite program (SDP), [10], is also considered.

The paper is organized as follows. In section 1 we present the basic notions about TTD problems with a detailed explanation of the problem formulation, emphasizing on the equilibrium constraints. The obtained model is hard to solve, but an easier equivalent convex problem is presented in Section 1.4. In Section 2 we present a reformulation of the last problem as a minimization of a convex non-smooth function with less variables, being only one of them non-negative. In Section 3, we describe a descent algorithm to solve this problem. In Section 5, an alternative reformulation is presented as a semidefinite programming problem. We briefly derive the required linear matrix inequalities, and explore different alternative formulations of the problem, which enable the use of CSDP3.2 package [6, 5]. In order to simplify the exposition we include some important results from linear algebra and Positive Semidefinite Programming (SDP) in Section 4. Finally, in Section 7 we present computational results obtained for both methodologies.

1 Problem Formulation

This section starts by introducing the basic engineering concepts that are important to the design of trusses.

1.1 Trusses, Loads and Compliance

A truss is a two or three dimensional structure composed of bars linked at nodes or joints which may be fixed, free or supported. In this work, we only consider two dimensional trusses. There is no loss of generality since three dimensional trusses can be approached by similar techniques but with a substantial increase on the number of variables. We distinguish the nodes on their *degrees of freedom*. A fixed node has 0 degrees of freedom. In the two dimensional case, a free node has 2 degrees of freedom (it can be moved along each direction on the plane) and, a supported node has just 1. The total number of degrees of freedom of the truss is the sum of the corresponding values on its nodes. The bars are all made of the same material. This material has elastic properties which are assumed linear with Young's modulus E.

When external forces, represented by a vector f, are acting on the nodes the structure deforms until the reaction caused by the deformation of the bars balances the external load. We may describe that deformation by the vector of nodal displacements, u, being the work done by external forces $f^\top u$. We call *compliance* to $\frac{1}{2}f^\top u$. This is a measure of the stiffness

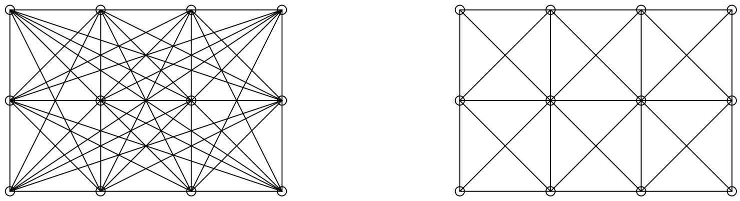


Figure 1: Rich and poor topologies.

of the truss, of its ability to withstand the load: the smaller the compliance the larger the stiffness of the truss with respect to the load.

Initially, we have a basic truss, the so-called *ground structure*, which is a previously chosen set of nodes and connecting bars. Usually we take a mesh of regularly spaced nodes. If we consider all possible links between the nodes we call it the *rich topology*, while if we consider only the links between neighboring nodes we call it the *poor topology*. In Figure 1, we show both alternatives for one set of nodes.

The goal is to find the stiffest truss capable of withstand the given load with a total volume that do not exceed a predefined value. We have to distribute the volume of the truss among the bars in order to get the more rigid construction, i.e., the one that minimizes the compliance. Only the bars with nonzero cross-sectional area are part of the final structure. This is what is called "truss topology design".

In order to formulate the problem, we consider a ground planar structure with k nodes, n degrees of freedom, m tentative bars and an external load $f \in \mathbb{R}^n$. The design variables in the problem are the cross-sectional area of the bars, a_i, with bounds, $L_i \leq a_i \leq U_i$, $i = 1, \ldots, m$. The predefined maximum volume for the structure will be represented by $v(> 0)$. Denoting by s_i the length of bar i, the set of all admissible vectors for the cross-sectional area of the bars is

$$\mathcal{A} = \left\{ a \in \mathbb{R}^m : \sum_{i=1}^{m} a_i s_i \leq v \,,\; L \leq a \leq U \right\}$$

where $a = (a_1, \ldots, a_m)$, $L = (L_1, \ldots, L_m)$ and $U = (U_1, \ldots, U_m)$. We assume the following:

- $0 \leq L_i < U_i, \; i = 1, \ldots, m$;
- $s_i U_i \leq v, \; i = 1, \ldots, m$;
- $\sum_{i=1}^{m} s_i L_i < v < \sum_{i=1}^{m} s_i U_i$.

Typically m is much larger than n.

The truss should be able to withstand the external load. This is assured by the *equilibrium equation*:

$$K(a)u = f \tag{1}$$

where $u \in \mathbb{R}^n$ is the nodal displacement vector and $K(a)$ is the $n \times n$ stiffness matrix of the structure. In the following subsection, we explain the equilibium equation with some detail.

The problem can be formulated as follows ([2, 3, 4, 8]) [1]:

$$(P) \qquad \begin{array}{ll} \min & f^\top u \\ s.t. & K(a)u = f \\ & a \in \mathcal{A} \\ & u \in \mathbb{R}^n. \end{array}$$

Note that problem (P) is non-convex due to the equilibrium equation and has a large number of variables $(n+m)$ and constraints $(n+2m+1)$. To get an idea of the size of TTD problems, we can easily notice that, in the case of the rich topology, we can get up to $m = \frac{1}{2}k(k-1)$ bars being the number of the nodes, k, typically large. Fortunately, this problem can be transformed to an equivalent convex programming problem, as we will see in Section 1.4, which can be rewritten as a non-smooth convex problem with only $n+1$ variables and 1 constraint (see Section 3) or as a semidefinite problem (see Section 5).

1.2 Equilibrium equation

Let a_i and s_i, denote the cross-sectional area and length of bar number i, respectively.

The general law for energy conservation, [7], states that:

$$f^\top u = q^\top \Delta s, \tag{2}$$

where $q \in \mathbb{R}^m$ is the vector of internal bar forces and $\Delta s \in \mathbb{R}^m$ is the vector of the bar elongations.

The stress in bar i, σ_i, given by $\frac{q_i}{a_i}$, measures the intensity of internal forces by unit of area. Each given material has a limit of proportionality, see [7], within which the elastic behavior is linear and the so-called *Hooke's law* is valid:

$$\sigma_i = E\frac{\Delta s_i}{s_i}$$

with E a constant specific to each material, called the Young's modulus.

As $\sigma_i = \frac{q_i}{a_i}$ we can write

$$q_i = \frac{Ea_i}{s_i}\Delta s_i = k_i \Delta s_i$$

where $k_i = E\frac{a_i}{s_i}$ is known as the stiffness of the bar i. Similar equations can be written for all m bars of the structure obtaining

$$q = D\Delta s, \tag{3}$$

where D is a diagonal matrix with $D_{ii} = E\frac{a_i}{s_i}$ for all $i = 1, \ldots, m$.

All deformations are assumed to be small, i.e., it is assumed that the resulting displacements do not significantly affect the geometry of the structure and hence do not affect the forces on the bars [17, 7].

[1] In the objective function, to simplify, we consider twice the compliance.

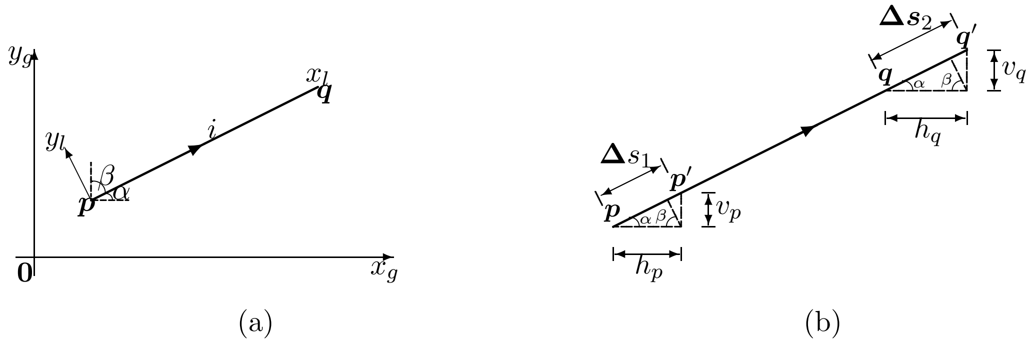


Figure 2: (a) Coordinates of bar i (b) Bar elongation

In order to derive equilibrium constraints we will construct the *compatibility matrix* B. It relates (small) nodal displacements, u, with (small) bar elongations, Δs, and relates nodal forces, f, with bar forces, q, by

$$\Delta s = Bu \ , \ f = B^{\top} q.$$

Consider the bar i in the plan with node $p = (x_p, y_p)$ as its first end, and node $q = (x_q, y_q)$ as its second end (see Figure 2 (a)). We assume that both nodes are free, i.e., that both have two degrees of freedom. The $x_g 0 y_g$ axes refer to the whole structure. The bar itself has a pair of local axes x_l and y_l. Positive direction of x_l is indicated by an arrow which is pointing to the second end of the bar.

The axial external load, f, causes displacements of both end nodes, p and q. In the overall referential, consider $u_p = (h_p, v_p)$ and $u_q = (h_q, v_q)$ where h_p and v_p denotes the horizontal and vertical displacement of node p, respectively, and h_q and v_q are the corresponding quantities for node q. Accordingly, the end nodes of the bar move by the amounts Δs_1 and Δs_2 (cf. Figure 2 (b)) parallel to its px_l axis. Hence the new position of the bar is given by p' and q' as shown in the figure. The elongation of this bar is:

$$\begin{aligned} \Delta s_i &= -\Delta s_1 + \Delta s_2 \\ &= -h_p \cos\alpha - v_p \sin\alpha + h_q \cos\alpha + v_q \sin\alpha, \end{aligned}$$

where α is the angle between bar i and the horizontal positive direction x_g. In matricial form, we can write:

$$\Delta s_i = [\,\cdots \quad \overbrace{-\cos\alpha \quad -\sin\alpha}^{p} \quad \cdots \quad \overbrace{\cos\alpha \quad \sin\alpha}^{q} \quad \cdots\,] \begin{bmatrix} \vdots \\ h_p \\ v_p \\ \vdots \\ h_q \\ v_q \\ \vdots \end{bmatrix}.$$

The row vector $[\,\cdots \quad -\cos\alpha \quad -\sin\alpha \quad \cdots \quad \cos\alpha \quad \sin\alpha \quad \cdots\,]$ is known as the displacement transformation matrix $[B]_i$ for the bar i.

Let us consider now supported nodes. If the first end node, p, is constrained to move only vertically and the second one, q, only horizontally, then we obtain

$$\Delta s_i = [\cdots \quad \overbrace{-\sin\alpha}^{p} \quad \cdots \quad \overbrace{\cos\alpha}^{q} \quad \cdots] \begin{bmatrix} \vdots \\ v_p \\ \vdots \\ h_q \\ \vdots \end{bmatrix}.$$

Other cases are similar. Writing this equation for all the bars of the structure, we obtain the matricial equation

$$\Delta s = Bu, \tag{4}$$

where $B \in \mathbb{R}^{m\times n}$, whose i^{th} line is $[B]_i$, is called the *compatibility matrix* of the structure.

By equations (2) and (4), the equality $f^\top u = q^\top Bu$ holds for every vector u, so:

$$f^\top = q^\top B.$$

Using (3) and (4), we obtain:

$$f = B^\top q = B^\top D\Delta s = B^\top DBu.$$

Defining $K = B^\top DB$, known by *stiffness matrix* of the structure, we obtain the *equilibrium equation*:

$$f = Ku,$$

The matrix K (or $K(a)$ to emphasize that it depends on a) can also be obtained by:

$$K(a) = \sum_{i=1}^{m} a_i s_i K_i \tag{5}$$

where K_i, the *stiffness matrix of bar* i, can be obtained by the formula

$$K_i = b_i b_i^\top, \tag{6}$$

being $b_i = \frac{\sqrt{E}}{s_i}[B]_i$. As we can easily see from (6), K_i is a rank 1 symmetric positive semidefinite matrix. Moreover, from the engineering point of view, it is standard to assume that B has full rank (Ref.[2]), making $K(a) = B^\top DB$ to be positive definite if $a > 0$. In fact, if $a > 0$ then all the diagonal elements of D are greater than zero and so D is positive definite. Furthermore, as B has full rank then $Bx \neq 0$, for all $x \neq 0$ and so,

$$x^\top K(a)x = x^\top B^\top DBx > 0, \text{ for all } x \in \mathbb{R}^n \setminus \{0\}.$$

1.3 Examples

To illustrate the previous concepts, we present two small examples.

In one of them, we consider the structure presented in Figure 3 with 6 nodes and 5 bars. In the lower left corner of the figure the referential to the whole structure is presented. Node

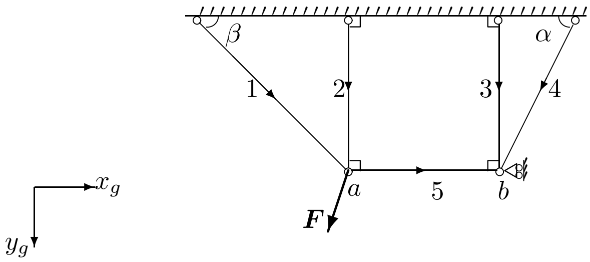


Figure 3: Truss with 5 bars and 3 degrees of freedom.

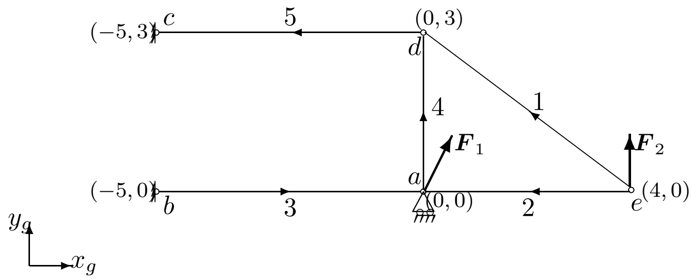


Figure 4: Truss with 5 bars and 5 degrees of freedom.

a is free, node b can be moved in the y_g direction. Nodes a and b have, respectively, 2 and 1 degrees of freedom, while the remaining nodes are fixed.

As the structure has five bars and three degrees of freedom, $B \in M_{5\times 3}$ and $K_i \in M_{3\times 3}$:

$$B = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & \sin\alpha \\ -1 & 0 & 0 \end{bmatrix}$$

$$K_1 = \frac{E}{s_1^2}\begin{bmatrix} \cos^2\beta & \cos\beta\sin\beta & 0 \\ \cos\beta\sin\beta & \sin^2\beta & 0 \\ 0 & 0 & 0 \end{bmatrix}, K_2 = \frac{E}{s_2^2}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$K_3 = \frac{E}{s_3^2}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, K_4 = \frac{E}{s_4^2}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sin^2\alpha \end{bmatrix}, K_5 = \frac{E}{s_5^2}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The vector u of nodal displacements has three components, $u = (h_a, v_a, v_b)$. The horizontal displacement of a is h_a, its vertical displacement is v_a and v_b is the vertical displacement of node b. In Section 7 we present computational results for this structure considering $\beta = 45°$, $\alpha = 60°$ and an external load $\boldsymbol{F}$ acting at node a.

In the other example, we consider the structure presented in Figure 4. In the lower left corner of the figure the referential to the whole structure is presented. The structure has five bars and five nodes. The node coordinates are given in parenthesis. Nodes d and e are free,

Table 1: Structure data.

bar	length	$\cos\alpha$	$\sin\alpha$	1^{st} end node	2^{nd} end node
1	5	-0.8	0.6	e	d
2	4	-1	0	e	a
3	5	1	0	b	a
4	3	0	1	a	d
5	5	-1	0	d	c

node a can be moved in the x_g direction having, respectively, 2, 2 and 1 degrees of freedom, while the remaining nodes are fixed.

From the coordinates of the end nodes of each bar we calculate the length and the direction cosines of the bars. The results are summarized in Table 1.

As the structure has 5 bars and 5 degrees of freedom then $B, K_i \in M_{5\times 5}$. We have

$$B = \begin{bmatrix} 0 & -0.8 & 0.6 & 0.8 & -0.6 \\ -1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

$$K_1 = \frac{E}{25}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.64 & -0.48 & -0.64 & 0.48 \\ 0 & -0.48 & 0.36 & 0.48 & -0.36 \\ 0 & -0.64 & 0.48 & 0.64 & -0.48 \\ 0 & 0.48 & -0.36 & -0.48 & 0.36 \end{bmatrix}, \quad K_2 = \frac{E}{16}\begin{bmatrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$K_3 = \frac{E}{25}\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad K_4 = \frac{E}{9}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad K_5 = \frac{E}{25}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

There are two external loads, $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$, acting in the nodes a and e, respectively, as shown by the depicted arrows. The intensity of load $\boldsymbol{F}_1$ is $20N$ and its angle with $\mathbf{0}x_g$ is $60°$. The intensity of load $\boldsymbol{F}_2$ is $30N$ and the angle is $90°$.

The vector of nodal displacements has five components, $u = (h_a, h_d, v_d, h_e, v_e)$, being h_i the horizontal displacement of node i ($i = a, d, e$) and v_i the vertical displacement of node i ($i = d, e$). As for the previous example, we present in Section 7 some computational results.

1.4 An equivalent large-scaled convex problem - (CP)

Problem (P) is, as already mentioned, hard to solve. However, as shown in [2, 8], it is equivalent to:

$$(CP) \qquad Z_1 = \min_{a\in\mathcal{A}} \max_{u\in\mathbb{R}^n} \left\{2f^\top u - u^\top K(a)u\right\}.$$

This is a convex programming problem. But it is still hard to solve directly. In the next section we present an equivalent optimization problem where we minimize a convex non-smooth function in $n+1$ variables, being only one non-negative and the others free. Later, in Section 5, we also present a reformulation of (CP) as a semidefinite programming problem.

2 A smaller equivalent convex problem - (CP_2)

This section is based mainly on [2]. The model studied in [2] requires the volume of the structure to be equal to a given value, while our version constraints the volume of the structure not to exceed a maximum. This makes the model similar to the semidefinite programming models to be presented later on. To make the present article self-contained we state all the results needed, some of them being modified from those in [2] in order to accommodate for this slight change in the model.

Consider the optimization problem:

$$(CP_2) \qquad Z_2 = \min_{u\in\mathbb{R}^n, \lambda\in\mathbb{R}^+} F(u,\lambda)$$

with

$$F(u,\lambda) := F_0(u,\lambda) + \sum_{i=1}^{m} s_i F_i(u,\lambda) \tag{7}$$

where

$$F_0(u,\lambda) := \lambda v - f^\top u \quad \text{and} \quad F_i(u,\lambda) := \max\left\{\left(\frac{1}{2}u^\top K_i u - \lambda\right)U_i, \left(\frac{1}{2}u^\top K_i u - \lambda\right)L_i\right\}$$

and $\mathbb{R}^+$ is the set of nonnegative real numbers. This is a convex minimization problem with $n+1$ variables and only one constraint. The objective function, F, is convex: it is the sum of several functions, being one of them linear, and the others convex, as they are the maximum of two convex quadratic functions. However, it is non-smooth.

The following theorem sets up a first relation between problems (CP) and (CP_2):

Theorem 2.1 ([2, 8])

$$Z_1 = -2Z_2.$$

Next theorem guarantees the existence of an optimal solution of (CP_2). We present a proof different from the corresponding one in [2, 8].

Theorem 2.2 *There exist $\overline{u}\in\mathbb{R}^n$ and $\overline{\lambda}\in\mathbb{R}^+$ such that*

$$F(\overline{u},\overline{\lambda}) = \min_{u\in\mathbb{R}^n, \lambda\in\mathbb{R}^+} F(u,\lambda).$$

Proof. The function F is convex on $\mathbb{R}^{n+1}$, and so it is continuous on $\mathbb{R}^{n+1}$. For $\lambda \geq 0$, considering $a \in \mathcal{A},\ a > 0$ and the assumptions of page 128, we have

$$\begin{aligned} F(u,\lambda) &\geq \lambda v - f^\top u + \sum_{i=1}^{m} s_i a_i \left(\frac{1}{2}u^\top K_i u - \lambda\right) \\ &= \lambda \left(v - \sum_{i=1}^{m} a_i s_i\right) - f^\top u + \frac{1}{2}u^\top \left(\sum_{i=1}^{m} a_i s_i K_i\right) u \\ &\geq -f^\top u + \frac{1}{2}u^\top \left(\sum_{i=1}^{m} a_i s_i K_i\right) u \\ &\geq -\|f\|\|u\| + \frac{1}{2}\eta_a \|u\|^2, \end{aligned}$$

being the last inequality a consequence of the Cauchy-Schwarz inequality and of the Rayleigh-Ritz theorem, [14]; η_a is the smallest eigenvalue of the positive definite matrix $\sum_{i=1}^{m} a_i s_i K_i$ and so $\eta_a > 0$.

For $\lambda < 0$, considering the assumptions of pages 125 and 128, we have

$$\begin{aligned} F(u,\lambda) &\geq \lambda v - f^\top u + \sum_{i=1}^{m} s_i U_i \left(\frac{1}{2}u^\top K_i u - \lambda\right) \\ &= \lambda \left(v - \sum_{i=1}^{m} U_i s_i\right) - f^\top u + \frac{1}{2}u^\top \left(\sum_{i=1}^{m} U_i s_i K_i\right) u \\ &\geq -f^\top u + \frac{1}{2}u^\top \left(\sum_{i=1}^{m} U_i s_i K_i\right) u \\ &\geq -\|f\|\|u\| + \frac{1}{2}\eta_u \|u\|^2, \end{aligned}$$

where η_u (> 0) is the smallest eigenvalue of the positive definite matrix $\sum_{i=1}^{m} U_i s_i K_i$.

So, $F(u,\lambda) \to +\infty$ when $\|(u,\lambda)\| \to +\infty$. This guarantees that $F(u,\lambda)$ has a minimum on $\mathbb{R}^{n+1}$. Let $\mathcal{X}$ be the set of all the minima of $F(u,\lambda)$ on $\mathbb{R}^{n+1}$.

If $\mathcal{X} \cap (\mathbb{R}^n \times \mathbb{R}^+) \neq \emptyset$, the existence of an optimal solution of (CP_2) is established. So, let us suppose that $\mathcal{X} \cap (\mathbb{R}^n \times \mathbb{R}^+) = \emptyset$. In this case, being $F(u,\lambda)$ convex on $\mathbb{R}^{n+1}$, the minimum on $\mathbb{R}^n \times \mathbb{R}^+$ exists and has to be on the hyperplane $\lambda = 0$. □

Theorem 2.1 defined a first connection between (CP) and (CP_2). The following theorem completes that connection, defining the optimality conditions for (CP_2) and showing how to obtain an optimal solution of (CP) from an optimal solution of (CP_2).

Theorem 2.3 ([2, 8]) *Consider $(\overline{u}, \overline{\lambda}) \in \mathbb{R}^n \times \mathbb{R}^+$ and define the sets*

$$J^- := \left\{i : \frac{1}{2}\overline{u}^\top K_i \overline{u} < \overline{\lambda}\right\},\ J^+ := \left\{i : \frac{1}{2}\overline{u}^\top K_i \overline{u} > \overline{\lambda}\right\}\ J := \left\{i : \frac{1}{2}\overline{u}^\top K_i \overline{u} = \overline{\lambda}\right\}.$$

The pair $(\overline{u}, \overline{\lambda})$ is an optimal solution of problem (CP_2) if and only if there exist $a \in \mathbb{R}^n$ and $\mu \in \mathbb{R}^+$ such that

1. $a_i = L_i$ *if* $i \in J^-$*;*

2. $a_i = U_i$ *if* $i \in J^+$*;*

3. $L_i \leq a_i \leq U_i$ *if* $i \in J$*;*

4. $\sum_{i=1}^{m} a_i s_i K_i \overline{u} = f$*;*

5. $\sum_{i=1}^{m} a_i s_i + \mu = v$*;*

6. $\mu \overline{\lambda} = 0$.

Moreover, the pair $(\overline{u}, a)$ *is an optimal solution for* (CP)*.*

Next, we present a technical result that defines an efficient way to compute $\overline{\lambda}$ for a given $\overline{u}$.

Theorem 2.4 ([2, 8]) *Let* $\overline{u} \in \mathbb{R}$*,*

$$\overline{\lambda} = \arg\min_{\lambda \in \mathbb{R}^+} F(\overline{u}, \lambda),$$

$\{i_1, i_2, \ldots, i_m\}$ *a permutation of* $\{1, 2, \ldots, m\}$ *such that*

$$\overline{u}^\top K_{i_1} \overline{u} \leq \overline{u} K_{i_2} \overline{u} \leq \ldots \leq \overline{u}^\top K_{i_m} \overline{u} \tag{8}$$

and, finally, p*, the largest integer such that*

$$\sum_{j=p}^{m} s_{i_j} U_{i_j} + \sum_{j=1}^{p-1} s_{i_j} L_{i_j} \geq v \qquad (p \leq m).$$

Then

$$\overline{\lambda} = \frac{1}{2} \overline{u}^\top K_{i_p} \overline{u}.$$

In the following section we present an algorithm to solve (CP_2) and, consequently, (CP).

3 A descend Algorithm to solve CP and CP_2

Problem (CP_2) is a convex problem in $\mathbb{R}^n \times \mathbb{R}^+$ where the objective function, F, is non-smooth. Since F is convex and finite, it has a non-empty subdifferential at every point $(u, \lambda) \in \mathbb{R}^n \times \mathbb{R}^+$, $\partial F(u, \lambda)$ ([21]). This set was already characterized in ([8]). Using this information, it is possible to apply algorithms based on the *separation oracles*, such as *cutting plane method* ([9, 16, 18]) or *ellipsoid method* ([22, 23, 24]). These methods are characterized by decreasing the *search domain* until its size be small enough or until other stopping criteria be satisfied. A subgradient, and thus a supporting hyperplane, is all the information needed to reduce the search domain in each step. However these methods are difficult to apply because it is necessary to know in advance a compact set including an optimal solution.

Descent methods are also traditionally used to solve minimization unconstrained problems, $\min_{x\in\mathbb{R}^n} f(x)$. Starting at an initial point x^0, a sequence $\{x^k\}$ is constructed forcing the objective function f to decrease at each iteration:

$$f(x^{k+1}) < f(x^k)\ ,\ k = 0, 1, \ldots \tag{9}$$

To solve (CP_2), where λ is non-negative, we apply a descent method to solve $\min\limits_{(u,\lambda)\in\mathbb{R}^{n+1}} F(u, \lambda)$; if, at iteration k, the obtained value for λ^k is negative, we project the corresponding (u^k, λ^k) over $\mathbb{R}^n \times \mathbb{R}^+$ making $\lambda^k = 0$.

3.1 Descent methods

The next iterate, x^{k+1}, is defined from the current one, x^k, in two steps: first, a descent direction d^k is computed; after, one computes a stepsize $t_k > 0$ such that the new iterate, $x^{k+1} := x^k + t_k d^k$ satisfies the condition $f(x^k + t_k d^k) < f(x^k)$. This procedure is repeated until a stopping criteria is satisfied ([12]).

The success of these kind of methods depend on the choice of the step size t_k and of the direction d^k. They must be carefully chosen. It is known that d is a descent direction of function $f : \mathbb{R}^n \to \mathbb{R}$ at x if one of the following conditions is true:

- $f'(x; d) < 0$, where $f'(x; d)$ is the directional derivative of f at x in the direction d;
- $s^\top d < 0,$ for all $s \in \partial f(x)$;
- $\sigma_{\partial f(x)}(d) < 0$, where $\sigma_{\mathcal{S}}(x) := \sup\{s^\top x : s \in \mathcal{S}\}$ is the *support function* of set $\mathcal{S}$.

If one chooses d such that $f'(x; d)$ be as negative as possible, the so-called *steepest descent* direction is obtained. However, since the function $d \mapsto f'(x; d)$ is positively homogeneous of degree one it is also necessary to bound the length of the direction because any negative directional derivative can be indefinitely extended. In the following result, an easy way to obtain a steepest descent direction, ([12, 8]), is presented.

Lemma 3.1 *Consider a function $f : \mathbb{R}^n \to \mathbb{R}$ such that $f'(x; d)$ exists for each $x, d \in \mathbb{R}^n$ and $d \mapsto f'(x; d)$ is continuous. Under these conditions, the optimal value of*

$$\min_{d\in\mathbb{R}^n} \left\{ f'(x; d) + \frac{1}{2}\|d\|^2 \right\} \tag{10}$$

is finite and non-positive.

Furthermore, this value is negative if and only if there exists d such that $f'(x; d) < 0$.

When f is convex we have the following corollary:

Corollary 3.1 *If $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function and $d \in \mathbb{R}^n$ is an optimal solution of the problem (10) at $\bar{x}$ then $d = 0$ if and only if $\min_{x\in\mathbb{R}^n} f(x) = f(\bar{x})$.*

A steepest descent direction of the function F at (u, λ) is obtained solving the quadratic minimization problem ([12, 2, 8]),

$$(P_d) \qquad \min_{d\in\mathbb{R}^n,\, \delta\in\mathbb{R}} \left\{ F'(u,\lambda;d,\delta) + \frac{1}{2}\left(\|d\|^2 + \delta^2\right) \right\}$$

where $F'(u, \lambda; d, \delta)$ is a directional derivative of F at (u, λ) in the direction (d, δ).

If the optimal solution of (P_d) is $(\hat{d}, \hat{\delta}) = (0, 0)$ then, by Corollary 3.1, the corresponding (u, λ) is the optimal solution.

Using some results about directional derivatives ([21, 12]), we have:

$$\begin{aligned} F'(u,\lambda;d,\delta) = -f^{\top}d + v\delta + \sum_{i\in J^-} s_i L_i\left((K_iu)^{\top}d - \delta\right) + \sum_{i\in J^+} s_i U_i\left((K_iu)^{\top}d - \delta\right) + \\ + \sum_{i\in J} s_i \max\left\{L_i\left((K_iu)^{\top}d - \delta\right)\ ,\ U_i\left((K_iu)^{\top}d - \delta\right)\right\}. \end{aligned}$$

Defining

$$\begin{aligned} \overline{v} &:= v - \sum_{J^-} s_i L_i - \sum_{J^+} s_i U_i\ , \\ \overline{f} &:= f - \sum_{J^-} s_i L_i K_i u - \sum_{J^+} s_i U_i K_i u, \end{aligned} \tag{11}$$

one gets:

$$F'(u,\lambda;d,\delta) = \overline{v}\delta - \overline{f}^{\top}d + \sum_{i\in J}\mu_i$$

with,

$$\mu_i := \max\left\{s_iL_i\left((K_iu)^{\top}d - \delta\right)\ ,\ s_iU_i\left((K_iu)^{\top}d - \delta\right)\right\},\ i \in J,$$

and problem (P_d) can be written as

$$\begin{aligned} (P_d) \qquad & \min_{d,\delta} \quad \overline{v}\delta - \overline{f}d + \sum_{i\in J}\mu_i + \frac{1}{2}\|d\|^2 + \frac{1}{2}\delta^2 \\ & s.t. \quad \mu_i \geq s_iU_i((K_iu)^{\top}d - \delta),\ i \in J \\ & \qquad\;\; \mu_i \geq s_iL_i((K_iu)^{\top}d - \delta),\ i \in J \end{aligned}$$

The optimal solution of (P_d) can be obtained solving its dual ([8])

$$\begin{aligned} (D_d) \qquad & \max_{\tau} \quad \left\{ -\tfrac{1}{2}\left\|\sum_J \tau_iK_iu - \overline{f}\right\|^2 - \tfrac{1}{2}\left\|\sum_J \tau_i - \overline{v}\right\|^2 \right\} \\ & s.t. \quad s_iL_i \leq \tau_i \leq s_iU_i,\ i \in J \end{aligned}$$

where $\tau = (\tau_1, \tau_2, \ldots, \tau_k)$ with $k = \#J$[2].

[2] $\#A$ is the cardinal of set A.

This is a quadratic problem with bounded variables that can be efficiently solved.

Let $\overline{\tau}$ be an optimal solution for (D_d). By the primal-dual relations ([1, 21]) an optimal solution of (P_d), $(\overline{d}, \overline{\delta})$, is given by:

$$\begin{aligned} \overline{d} &= -\left(\sum_{i\in J} \overline{\tau}_i K_i u - \overline{f}\right), \\ \overline{\delta} &= \sum_{i\in J} \overline{\tau}_i - \overline{v}. \end{aligned} \tag{12}$$

Now we are able to apply a steepest descent direction algorithm to solve problem (CP_2) and, consequently, using Theorem 2.3, problem (CP). However, descent methods do not necessarily converge ([25, 12].

An improved convergent version of the descent method is presented in the next section.

3.2 ε-descent methods

A way to avoid the non-convergence of descent methods is to consider the ε-subdifferential of f at x instead of the subdifferential. This concept uses information about the function not only in x but also in a neighbourhood of x.

Next, we present some definitions.

Definition 3.1 ([13]) *A vector $s \in \mathbb{R}^n$ is a ε-**subgradient of** f **at** $x \in \operatorname{dom} f$ if*

$$f(y) \geq f(x) + s^\top (y - x) - \varepsilon,$$

*for each $y \in \mathbb{R}^n$. The ε-**subdifferential**, $\partial_\varepsilon f(x)$, is the set of all ε-subgradient of f at x.*

Definition 3.2 ([13]) *The ε-**directional derivative of** f **at** $x \in \operatorname{dom} f$ **relative to** d is*

$$f'_\varepsilon(x; d) = \sup_{s\in\partial_\varepsilon f(x)} s^\top d.$$

It can be proven that $\partial_\varepsilon f(x)$ is a closed and convex set, for all $\varepsilon > 0$. This implies that $f'_\varepsilon(x; d)$ is always well defined.

Definition 3.3 ([13]) *A nonzero vector $d \in \mathbb{R}^n$ is said to be an ε-**descent direction for** f **at** x if $f'_\varepsilon(x; d) < 0$, in other words, if d defines an hyperplane separating $\partial_\varepsilon f(x)$ and $\{0\}$. A point $x \in \mathbb{R}^n$ is said to be an ε-**minimum of** f if there is no such separating d, i.e. $f'_\varepsilon(x, d) \geq 0$ for all d i.e., $0 \in \partial_\varepsilon f(x)$.*

Proposition 3.1 ([13]) *A direction $d \in \mathbb{R}^n$ is ε-descent if and only if*

$$f(x + td) < f(x) - \varepsilon,$$

for some $t > 0$.

A point $x \in \mathbb{R}^n$ is an ε-minimum of f if and only if it minimizes f within ε, i.e., $f(y) \geq f(x) - \varepsilon$, for all $y \in \mathbb{R}^n$.

Next, we describe an ε-descent algorithm for solving $\min_{x \in \mathbb{R}^n} f(x)$.

A general ε-descent algorithm

Step 0 Start from some $x^0 \in \mathbb{R}^n$. Choose $\varepsilon > 0$. Set $k := 0$.
Step 1 If $0 \in \partial_\varepsilon f(x^k)$ Stop. Otherwise compute d^k, an ε-descent direction.
Step 2 Make a line-search along d^k to obtain a step size $t_k > 0$ such that
$f(x^k + t_k d^k) < f(x^k) - \varepsilon$.
Step 3 Set $x^{k+1} := x^k + t_k d^k$. Replace k by $k+1$ and loop to **Step 1**.

In the following we will describe an ε-descent algorithm for problem (CP_2) which simultaneously solves problem (CP). This algorithm is similar to the one presented in [2], differing only in what is needed due to the volume constraint being an inequality in our case.

For $\varepsilon > 0$ define the following index sets:

$$\hat{J} := \left\{ i : \left| \frac{1}{2} u^\top K_i u - \lambda \right| \leq \frac{\varepsilon}{s_i U_i - s_i L_i} \right\},$$
$$\hat{J}^+ := \left\{ i : \frac{1}{2} u^\top K_i u - \lambda > \frac{\varepsilon}{s_i U_i - s_i L_i} \right\}$$

and

$$\hat{J}^- := \left\{ i : \frac{1}{2} u^\top K_i u - \lambda < -\frac{\varepsilon}{s_i U_i - s_i L_i} \right\}.$$

As in (11), consider

$$\hat{v} := v - \sum_{i \in \hat{J}^+} s_i U_i - \sum_{i \in \hat{J}^-} s_i L_i,$$
$$\hat{f} := f - \sum_{i \in \hat{J}^+} s_i U_i K_i u - \sum_{i \in \hat{J}^-} s_i L_i K_i u.$$

The vector (d, δ) is a $\varepsilon-$descent direction for problem (CP_2) if it is an optimal solution of the following quadratic problem

$$(\hat{P}_d) \qquad \min_{d,\delta,\mu} \left\{ \hat{v}\delta - \hat{f}^\top d + \sum_{i \in \hat{J}} \mu_i + \frac{1}{2}\|d\|^2 + \frac{1}{2}\delta^2 \right\}$$
$$s.t. \quad s_i U_i \left(d^\top K_i u - \delta + p_i \right) - \mu_i \leq 0 \,,\ i \in \hat{J}$$
$$s_i L_i \left(d^\top K_i u - \delta + p_i \right) - \mu_i \leq 0 \,,\ i \in \hat{J}$$

with

$$p_i := \frac{1}{2} u^\top K_i u - \lambda.$$

Problem $(\hat{P}_d)$ is a perturbation of problem (P_d). In fact, for every $i \in \hat{J}$ we have $|p_i| \leq \frac{\varepsilon}{s_i(U_i - L_i)}$. For small values of ε we have $\hat{J} \approx J$ and, for $\varepsilon = 0$ both problems coincide.

The dual problem of $(\hat{P}_d)$ is the following quadratic optimization problem ([8]):

$$(\hat{D}_d) \qquad \max_{\tau} \left\{ -\sum_{i\in\hat{J}} \tau_i p_i - \frac{1}{2}\left\|\sum_{i\in\hat{J}} \tau_i K_i u - f\right\|^2 - \frac{1}{2}\left\|\sum_{i\in\hat{J}} \tau_i - v\right\|^2 \right\}$$
$$s.t. \quad s_i L_i \le \tau_i \le s_i U_i \;,\; i \in \hat{J}.$$

Let $\hat{\tau}$ be an optimal solution for problem $(\hat{D}_d)$. As in (12), by the primal-dual relationships, the optimal solution of problem $(\hat{P}_d)$, $(\hat{d}, \hat{\delta})$, is given by:

$$\begin{aligned} \hat{d} &= -\left(\sum_{i\in\hat{J}} \hat{\tau}_i K_i u - \hat{f}\right), \\ \hat{\delta} &= \sum_{i\in\hat{J}} \hat{\tau}_i - \hat{v}. \end{aligned} \tag{13}$$

By Corollary 3.1, $\hat{d} = 0$ and $\hat{\delta} = 0$ if and only if (u, λ) is an ε-optimal solution for problem (CP_2).

Having an $\varepsilon-$descent direction, $(\hat{d}, \hat{\delta})$, the stepsize can be obtain by:

$$\arg\min_{\alpha\ge 0} F(u + \alpha\hat{d}, \lambda + \alpha\hat{\delta}).$$

Here we use an inexact line search of the Armijo-Goldstein type as it was made by Ben-Tal and Bendsøe in [2]. The rule is given in **Step 2**(d) of the following algorithm.

Next, we present a ε-descent algorithm to obtain an ε-optimal solution for problem (CP_2).

An ε-descent algorithm to solve (CP_2)

Step 0 Choose $\varepsilon > 0$, $\delta > 0$, $0 < \theta < \frac{1}{2}$, set $k := 0$;
Step 1 *initialization*
(a) Choose an initial value $a^0 : a^0 > 0, L \le a^0 \le U, \sum_{i=1}^m a_i^0 s_i \le v$;
(b) Solve the system $\sum_{i=1}^m a_i^0 s_i K_i u = f$. Let u^0 be its solution;
(c) Compute λ_0 in the following way: consider a permutation $(i_1,\ i_2,\ \ldots, i_m)$ of the set $\{1,\ 2, \ldots, m\}$ such that
$u^{0\top} K_{i_1} u^0 \le u^{0\top} K_{i_2} u^0 \le \ldots \le u^{0\top} K_{i_m} u^0$;
let p be the largest integer such that
$\sum_{j=p}^m s_i U_{i_j} + \sum_{j=1}^{p-1} s_i L_{i_j} \ge v \qquad (p \le m)$
then $\lambda_0 := \frac{1}{2} u^{0\top} K_{i_p} u^0$;
Step 2 *iteration* $k+1$ (u^k and λ_k known):
(a) Determine the index sets $\hat{J}_k$, $\hat{J}_k^-$, $\hat{J}_k^+$ and compute $\hat{v}^k$ and $\hat{f}^k$;
(b) Solve the problem $(\hat{P}_d)$ to obtain $(\hat{d}^k, \hat{\delta}^k)$
[solve $(\hat{D}_d)$ to obtain $\hat{\tau}^k$ and compute $(\hat{d}^k,\ \hat{\delta}^k)$ by formula (13)]
(c) If $\max(\|\hat{d}^k\|, |\hat{\delta}_k|) < \delta$ Stop.
(d) Compute the stepsize α_k as been the largest $\alpha > 0$ such that

$$F(u^k + \alpha \hat{d}^k, \lambda_k + \alpha \hat{\delta}_k) \leq F(u^k, \lambda_k) - \alpha\theta(\|d^k\|^2 + \hat{\delta}_k^2) \qquad (*)$$

Note: an approximation for α_k can be obtained as:
let $l(k)$ be the largest integer such that $\alpha = (\frac{1}{2})^{l(k)}$ verify (*),
then $\alpha_k = (\frac{1}{2})^{l(k)}$.

(e) Set:
$u^{k+1} := u^k + \alpha_k \hat{d}^k$,
$\lambda^{k+1} := \lambda^k + \alpha_k \hat{\delta}_k$.
If $\lambda^{k+1} < 0$ then consider $\lambda^{k+1} := 0$

(f) Replace k by $k+1$ and loop to **Step 2**;

With this algorithm, we obtain (u^k, λ^k) as an ε-optimal solution for problem (CP_2) corresponding to the ε-optimal value $Z_{2\varepsilon} = F(u^k, \lambda^k)$.

Using the relations between problems (CP) and (CP_2), the ε-optimal solution for problem (CP) is (a, u^k) with $a_i = L_i$ for $i \in \hat{J}_k^-$, $a_i = U_i$ for $i \in \hat{J}_k^+$ and $a_i = \frac{\hat{\tau}_j^k}{s_i}$ for $i \in \hat{J}_k$ and j the corresponding index in vector τ $(1 \leq j \leq \#J_k)$. The ε-optimal value is $Z_{1\varepsilon} = -2F(u^k, \lambda^k)$.

As we will see, (CP) can be formulated as a positive semidefinite problem. Before doing so, we present, in the next section, some useful results from linear algebra and semidefinite programming.

4 Semidefinite Programs

In this section, we review some fundamental properties of positive semidefinite matrices. We also introduce a standard form of the primal-dual pair of positive semidefinite programs (SDP). For a more complete explanation see Refs. [14, 15, 11].

Our notation is quite standard: $M_{n,m}$ denotes the set of $n \times m$ matrices, M_n the set of square matrices of dimension n, and S_n the set of the symmetric ones. The *trace* of matrix A, $\mathbf{Tr}(A)$, is the sum of the diagonal elements of A; $\operatorname{diag}(A)$ is the vector of the diagonal entries of $A \in S_n$; $\operatorname{Diag}(x)$ is the diagonal matrix with the vector x on its diagonal.

Definition 4.1
*$A \in S_n$ is **positive semidefinite** ($A \succeq 0$ or $A \in S_n^+$) if $x^\top Ax \geq 0$ for all $x \in \mathbb{R}^n$*
*$A \in S_n$ is **positive definite** ($A \succ 0$ or $A \in S_n^{++}$) if $x^\top Ax > 0$ for all $x \in \mathbb{R}^n \setminus \{0\}$.*

It is easy to prove that S_n^+ is a closed convex cone. This cone induces a partial order on the set of the symmetric matrices: for $A, B \in S_n$, $A \succeq B$ ($A \succ B$) if $A - B \in S_n^+$ ($A - B \in S_n^{++}$).

The standard formulation for the primal-dual pair of problems in positive semidefinite programming is given by

$$(PSDP) \quad \begin{array}{ll} & \inf\ c^\top x \\ & s.t.\ \sum_{i=1}^m x_i F_i + F_0 = F(x)\,, \\ & F(x) \succeq 0 \end{array} \qquad (DSDP) \quad \begin{array}{ll} \sup & -\mathbf{Tr}(F_0 Z) \\ s.t. & \mathbf{Tr}(F_i Z) = c_i,\ i = 1, \ldots, m \\ & Z \succeq 0 \end{array} \quad (14)$$

where $x \in \mathbb{R}^m$ is the primal vector variables, Z is the dual matrix variable, which has the same block structure as the given symmetric matrices $F_0, F_1, \ldots, F_m$, and $c \in \mathbb{R}^m$ is a given vector. In the previous formulation we considered just one semidefinite matrix variable, $F(x)$. This is not restrictive. In fact, any semidefinite program with several semidefinite matrices variables of varying dimensions can be formulated equivalently within standard $(PSDP)$, using the following result:

$$A_1 \succeq (\succ) 0, A_2 \succeq (\succ) 0, \ldots, A_m \succeq (\succ) 0 \Longleftrightarrow \begin{bmatrix} A_1 & 0 & \ldots & 0 \\ 0 & A_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & A_m \end{bmatrix} \succeq (\succ) 0. \tag{15}$$

The optimal value of $(DSDP)$ is a lower bound on the optimal value of $(PSDP)$. This property is called the **weak duality property**. There is also a **strong duality property**, similar to the one in linear programming:

Theorem 4.1 (Strong duality) *Assume that there exists a strictly feasible solution $\hat{Z}$ for $(DSDP)$ and let*

$$p^* = \inf \left\{ c^\top x : \sum_{i=1}^{m} x_i F_i + F_0 \succeq 0 \right\}$$

and

$$d^* = \sup \left\{ -\mathbf{Tr}(F_0 Z) : \mathbf{Tr}(F_i Z) = c_i,\ i = 1, \ldots, m,\ Z \succeq 0 \right\}.$$

Then $p^* = d^*$ and, if p^* is finite, it is attained for some x feasible for $(PSDP)$.

It is easy to see that linear programming is a special case of semidefinite programming. Several other convex optimization problems can be formulated as semidefinite programs. To see this, an helpful tool is the Schur Complement Theorem:

Theorem 4.2 (Schur Complement) *Let $A \in S_r^{++}$, $B \in S_q$ and $C \in M_{r,q}$. Then*

$$\begin{bmatrix} A & C \\ C^\top & B \end{bmatrix} \succeq\ (\succ)\ 0 \iff B \succeq\ (\succ)\ C^\top A^{-1} C.$$

The following lemmas are often used results about positive semidefinite matrices that we will need later.

Lemma 4.1 *If $A \in S_n^+$, then*

- $a_{ii} \geq 0$, $i = 1, \ldots, n$;

- $a_{ii} = 0 \Rightarrow a_{ij} = a_{ji} = 0,\ j = 1, \dots, n.$

Lemma 4.2 *Let $A, B \succeq 0$. Then, $\mathbf{Tr}(AB) \geq 0$ and $\mathbf{Tr}(AB) = 0$ if and only if $AB = 0$.*

The following lemma is also frequently used:

Lemma 4.3 ([10]) *For $f \in \mathbb{R}^n$ and $A \in S_n$,*

$$\begin{bmatrix} \tau & f^\top \\ f & A \end{bmatrix} \succeq 0 \Longleftrightarrow \tau + u^\top A u - 2u^\top f \geq 0 \ , \ \forall \ u \in \mathbb{R}^n.$$

In the next section, we briefly show how problem (CP) can be formulated as a semidefinite programming problem.

5 An SDP formulation for truss structure design

We can write (CP) as

$$\begin{aligned} & \min_{\tau,a} \ \tau \\ s.t. \quad & \tau \geq 2f^\top u - u^\top K(a) u \ , \ \forall \ u \in \mathbb{R}^n \\ & a \in \mathcal{A}, \end{aligned}$$

where $\mathcal{A} = \{a \in \mathbb{R}^m : \sum_{i=1}^m a_i s_i \leq v,\ L \leq a \leq U\ \}$. Using Lemma 4.3, we can write problem (CP) as:

$$\begin{aligned} & \min_{a,\tau} \ \tau \\ s.t. \quad & \begin{bmatrix} \tau & f^\top \\ f & K(a) \end{bmatrix} \succeq 0 \\ & \sum_{i=1}^m a_i s_i \leq v \\ & a - L \geq 0 \\ & -a + U \geq 0 \end{aligned}$$

The last two inequalities may be written as the following linear matrix inequality:

$$\begin{bmatrix} \mathrm{Diag}(a - L) & \\ & \mathrm{Diag}(-a + U) \end{bmatrix} \succeq 0.$$

Consequently, using (15), we obtain the following semidefinite formulation

$$
(SCP) \qquad \begin{array}{ll} \min\limits_{a,\tau} & \tau \\ s.t. & \begin{bmatrix} \tau & f^\top & & & \\ f & K(a) & & & \\ & & -\sum a_i s_i + v & & \\ & & & \mathrm{Diag}(a-L) & \\ & & & & \mathrm{Diag}(-a+U) \end{bmatrix} \succeq 0. \end{array}
$$

If we consider

$$
\begin{aligned}
a_i &:= L_i + \frac{1}{m+1} \min \left\{ \frac{1}{s_i} \left(v - \sum_{j=1}^{m} L_j s_j \right), \min_{j=1,\dots,m} \{U_j - L_j\} \right\}, \; i = 1, \dots, m, \\
\tau &:= f^\top (K(a))^{-1} f + 1
\end{aligned}
$$

we get a strictly feasible solution: using the assumptions of Section 1, we can conclude that $a > L$, $a < U$, $\sum_{i=1}^m a_i s_i < v$, $a_i > 0$ for $i = 1, \dots, m$, $K(a) \succ 0$ and, finally, $\tau > 0$. With this and applying Theorem 4.2, we have

$$
\begin{bmatrix} \tau & f^\top \\ f & K(a) \end{bmatrix} \succ 0.
$$

Using (15) we conclude immediately that the solution is strictly feasible.

Problem (SCP) is already an instance of $(PSDP)$ in variables $a_1, \dots, a_m, \tau$. To see this, just define the following matrices:

$$
F_0 := \begin{bmatrix} 0 & f^\top & & & \\ f & 0 & & & \\ & & v & & \\ & & & \mathrm{Diag}(-L) & \\ & & & & \mathrm{Diag}(U) \end{bmatrix}, \quad F_{m+1} := \begin{bmatrix} 1 & 0^\top & & & \\ 0 & 0 & & & \\ & & 0 & & \\ & & & 0 & \\ & & & & 0 \end{bmatrix}
$$

and

$$
F_i := \begin{bmatrix} 0 & 0^\top & & & \\ 0 & s_i K_i & & & \\ & & -s_i & & \\ & & & \mathrm{Diag}(e_i) & \\ & & & & \mathrm{Diag}(-e_i) \end{bmatrix}, \; i = 1, \dots, m,
$$

where e_i is the unitary vector with component i equal to 1.

All the matrices has the same block structure: one symmetric block of dimension $n+1$ and 3 diagonal blocks, one of dimension 1 and the others of dimension m.

In the following subsections, we derive the dual of (SCP), we get a new semidefinite programming problem equivalent to that dual. In section 6 we conclude that the dual of this new problem is equivalent to problem (P).

5.1 The dual problem of (SCP)

Using (14), the dual of problem (SCP) is given by:

$$\begin{aligned} \max\ & -\mathbf{Tr}(F_0 Z) \\ s.t.\ & \mathbf{Tr}(F_i Z) = 0,\ \ i = 1, \ldots, m \\ & \mathbf{Tr}(F_{m+1} Z) = 1 \\ & Z \succeq 0, \end{aligned}$$

being F_0, F_1, $\ldots$, F_{m+1} the matrices defined in the previous section and Z the dual variable with the following block structure:

$$Z := \begin{bmatrix} \lambda & z^\top & & & \\ z & \Sigma & & & \\ & & \theta & & \\ & & & \Omega' & \\ & & & & \Omega'' \end{bmatrix},$$

where $\lambda \in \mathbb{R}, z \in \mathbb{R}^n, \Sigma \in S_{n\times n}, \theta \in \mathbb{R}$ and Ω', Ω'' are $m \times m$ diagonals matrices.

The dual problem can be written as

$$(DSCP)\qquad \begin{aligned} \max_{z,\theta,\Omega',\Omega'',\Sigma}\ & -2f^\top z - v\theta + \sum_{i=1}^{m} L_i \Omega'_{ii} - \sum_{i=1}^{m} U_i \Omega''_{ii} \\ s.t.\ & \mathbf{Tr}(K_i \Sigma) = \theta + \frac{1}{s_i}\left(-\Omega'_{ii} + \Omega''_{ii}\right),\ i = 1, \ldots, m \\ & \begin{bmatrix} 1 & z^\top \\ z & \Sigma \end{bmatrix} \succeq 0 \\ & \theta \geq 0 \\ & \Omega'_{ii} \geq 0\,,\ \ i = 1, \ldots, m \\ & \Omega''_{ii} \geq 0\,,\ \ i = 1, \ldots, m. \end{aligned}$$

The objective function does not depend on matrix Σ. This fact and the structure of the first two constraints, suggest the possibility of having an equivalent problem without Σ. This would be a much simpler problem than $(DSCP)$.

5.2 An equivalent problem to $(DSCP)$

Lets define the problem

$$
(\widetilde{DSCP}) \quad
\begin{array}{ll}
\max\limits_{z,\theta,\Omega',\Omega''} & -2f^\top z - v\theta + \sum\limits_{i=1}^{m} L_i\Omega'_{ii} - \sum\limits_{i=1}^{m} U_i\Omega''_{ii} \\
s.t. & \begin{bmatrix} 1 & z^\top b_i \\ b_i^\top z & \theta + \frac{1}{s_i}\left(-\Omega'_{ii} + \Omega''_{ii}\right) \end{bmatrix} \succeq 0 \,, \; i = 1, \ldots, m \\
& \theta \geq 0 \\
& \Omega'_{ii} \geq 0 \,, \;\; i = 1, \ldots, m \\
& \Omega''_{ii} \geq 0 \,, \;\; i = 1, \ldots, m
\end{array}
\qquad (16)
$$

In the following theorems, we will prove the equivalence between $(DSCP)$ and $(\widetilde{DSCP})$.

Theorem 5.1 *A feasible solution of problem $(DSCP)$ corresponds to a feasible solution of problem $(\widetilde{DSCP})$ with the same objective value.*

Proof. Let $(z, \theta, \Omega', \Omega'', \Sigma)$ be a feasible solution of $(DSCP)$.We know, by Theorem 4.2, that

$$
\begin{bmatrix} 1 & z^\top \\ z & \Sigma \end{bmatrix} \succeq 0 \Leftrightarrow \Sigma \succeq zz^\top.
$$

Then, as $K_i \succeq 0$, applying Lemma 4.2 and using the equality constraint defined in $(DSCP)$, we obtain,

$$
\mathbf{Tr}(K_i zz^\top) \leq \mathbf{Tr}(K_i\Sigma) = \theta + \frac{1}{s_i}\left(-\Omega'_{ii} + \Omega''_{ii}\right) \,, \; i = 1, \ldots, m.
$$

As, by (6), $K_i = b_i b_i^\top$, $\mathbf{Tr}(K_i zz^\top) = \mathbf{Tr}\left(b_i b_i^\top zz^\top\right) = z^\top b_i b_i^\top z$. Then

$$
z^\top b_i b_i^\top z = (b_i^\top z)^\top b_i^\top z \leq \theta + \frac{1}{s_i}\left(-\Omega'_{ii} + \Omega''_{ii}\right) \,, \; i = 1, \ldots, m,
$$

which is equivalent to

$$
\begin{bmatrix} 1 & z^\top b_i \\ b_i^\top z & \theta + \frac{1}{s_i}\left(-\Omega'_{ii} + \Omega''_{ii}\right) \end{bmatrix} \succeq 0, \; i = 1, \ldots, m.
$$

So, $(z, \theta, \Omega', \Omega'')$ is a feasible solution of $(\widetilde{DSCP})$. It has, obviously, the same objective value as $(z, \theta, \Omega', \Omega'', \Sigma)$ in $(DSCP)$. □

Theorem 5.2 *An optimal solution of problem $(\widetilde{DSCP})$ corresponds to an optimal solution of problem $(DSCP)$.*

Proof. Let $(\hat{z}, \hat{\theta}, \hat{\Omega}', \hat{\Omega}'')$ be an optimal solution of $(\widetilde{DSCP})$. By (16), we obtain

$$\hat{\theta} + \frac{1}{s_i}\left(-\hat{\Omega}'_{ii} + \hat{\Omega}''_{ii}\right) \geq b_i^\top \hat{z}\, \hat{z}^\top b_i = \mathbf{Tr}\left((b_i b_i^\top)\, \hat{z}\hat{z}^\top\right)\ ,\ i = 1, \ldots, m.$$

As $K_i = b_i b_i^\top$, we can write, for each i,

$$\hat{\theta} + \frac{1}{s_i}\left(-\hat{\Omega}'_{ii} + \hat{\Omega}''_{ii}\right) \geq \mathbf{Tr}\left(K_i\, \hat{z}\hat{z}^\top\right).$$

We will prove that the previous inequality is satisfied, for all i, as an equality, at the optimal solution. In fact, let us suppose that, for an index i,

$$\hat{\theta} + \frac{1}{s_i}\left(-\hat{\Omega}'_{ii} + \hat{\Omega}''_{ii}\right) > \mathbf{Tr}\left(K_i\, \hat{z}\hat{z}^\top\right).$$

With $\hat{z}$, $\hat{\Omega}''_{ii}$ and $\hat{\theta}$ constants, we can get a greater value for $\hat{\Omega}'_{ii}$ such that

$$\hat{\theta} + \frac{1}{s_i}\left(-\hat{\Omega}'_{ii} + \hat{\Omega}''_{ii}\right) = \mathbf{Tr}\left(K_i\, \hat{z}\hat{z}^\top\right)\ (\geq 0).$$

Therefore, if $L_i > 0$ we obtain another feasible solution for $(\widetilde{DSCP})$ with objective value greater than the optimal one. This is an absurd. If $L_i = 0$ we obtained another feasible solution for $(\widetilde{DSCP})$ that satisfies the equality and the objective value is equal to the optimal one. So, there is an optimal solution of $(\widetilde{DSCP})$ that satisfies the equality for every i.

Considering $\hat{\Sigma} = \hat{z}\hat{z}^\top$, we get a feasible solution for $(DSCP)$ with the objective value equal to the optimal value of $(\widetilde{DSCP})$. Applying Theorem 5.1 we conclude that $(\hat{z}, \hat{\theta}, \hat{\Omega}', \hat{\Omega}'', \hat{\Sigma})$ is an optimal solution for $(DSCP)$. □

Defining the matrices

$$H_i(z, \theta, \Omega', \Omega'') = \begin{bmatrix} 1 & b_i^\top z \\ b_i^\top z & \theta + \frac{1}{s_i}(-\Omega'_{ii} + \Omega''_{ii}) \end{bmatrix},\ i = 1, \ldots, m,$$

problem $(\widetilde{DSCP})$ can be written as

$$\begin{aligned} \max_{z,\theta,\Omega',\Omega''} \quad & -2f^\top z - v\theta + \sum_{i=1}^{m} L_i \Omega'_{ii} - \sum_{i=1}^{m} U_i \Omega''_{ii} \\ s.t. \quad & A := \begin{bmatrix} H_1(z,\theta,\Omega',\Omega'') & & & & & \\ & \ddots & & & & \\ & & H_m(z,\theta,\Omega',\Omega'') & & & \\ & & & \theta & & \\ & & & & \Omega' & \\ & & & & & \Omega'' \end{bmatrix} \succeq 0. \end{aligned}$$

This problem has $2m+n+1$ variables and the constraint matrix has m symmetric blocks of dimension 2, one block of dimension 1 and two diagonal blocks of dimension m.

Problem $(\widetilde{DSCP})$ is strictly feasible: if we consider, for example, $z=0$, $\theta=1$, $\Omega'=I, \Omega''=I$, we will get $H_i = I_2$ and the constraint is strictly verified. By Theorem 4.1, this implies that the optimal value of $(\widetilde{DSCP})$ is equal to the optimal value of its dual.

Problem $(\widetilde{DSCP})$ is not the dual of (SCP), it is an equivalent problem to that dual. Let us think about the dual of $(\widetilde{DSCP})$. We will show, in the next section, that this dual is an alternative formulation of (SCP).

6 The dual problem of $(\widetilde{DSCP})$

The dual of problem $(\widetilde{DSCP})$ can be obtained as before, by casting the problem in the standard $(PSDP)$ format and then writing down the dual using (14). Nevertheless, in this case, it looks simpler to derive the dual using directly the Lagrangian duality theory.

Considering the dual variable, $B \succeq 0$, defined as

$$B := \begin{bmatrix} B_1 & & & & & \\ & \ddots & & & & \\ & & B_m & & & \\ & & & \xi & & \\ & & & & \Lambda & \\ & & & & & \Phi \end{bmatrix}, \; B_i := \begin{bmatrix} \phi_i & \beta_i \\ \beta_i & \gamma_i \end{bmatrix}, \; i=1,\ldots,m,$$

where $\phi_i, \beta_i, \gamma_i \in \mathbb{R}, \; i=1,\ldots,m, \; \xi \in \mathbb{R}$, Λ and Φ are $m \times m$ diagonal matrices, the **Lagrangian function** is given by

$$\begin{aligned} L(z,\theta,\Omega',\Omega'';\phi,\beta,\gamma,\xi,\Lambda,\Phi) &:= -2f^\top z - v\theta + \sum_{i=1}^m L_i \Omega'_{ii} - \sum_{i=1}^m U_i \Omega''_{ii} + \mathbf{Tr}(AB) \\ &= \sum_{i=1}^m \phi_i + 2\sum_{j=1}^n z_j \left(-f_j + \sum_{i=1}^m \beta_i (b_i)_j\right) + \theta\left(-v + \sum_{i=1}^m \gamma_i + \xi\right) \\ &\quad + \sum_{i=1}^m \Omega'_{ii}\left(L_i - \frac{\gamma_i}{s_i} + \Lambda_{ii}\right) + \sum_{i=1}^m \Omega''_{ii}\left(-U_i + \frac{\gamma_i}{s_i} + \Phi_{ii}\right) \end{aligned}$$

and the Lagrangian dual of the problem $(\widetilde{DSCP})$ is given by:

$$\min_{\phi,\beta,\gamma,\xi,\Lambda,\Phi} \; \max_{z,\theta,\Omega',\Omega''} L(z,\theta,\Omega',\Omega'';\phi,\beta,\gamma,\xi,\Lambda,\Phi).$$

We can easily see that the inner maximization problem is bounded from above only when the following conditions hold:

$$v = \sum_{i=1}^{m} \gamma_i + \xi, \quad f_j = \sum_{i=1}^{m} \beta_i (b_i)_j,\ j = 1, \ldots, n,$$
$$L_i = \frac{\gamma_i}{s_i} - \Lambda_{ii}, \quad U_i = \frac{\gamma_i}{s_i} + \Phi_{ii},\ i = 1, \ldots, m.$$

Under these conditions, the maximum value is

$$\max_{z,\theta,\Omega',\Omega''} L(z, \theta, \Omega', \Omega''; \phi, \beta, \gamma, \xi, \Lambda, \Phi) = \sum_{i=1}^{m} \phi_i.$$

The dual problem can now be written as

$$\begin{aligned} \min_{\phi,\beta,\gamma,\xi,\Lambda,\Phi} \quad & \sum_{i=1}^{m} \phi_i \\ s.t. \quad & f = \sum_{i=1}^{m} \beta_i b_i \\ & \sum_{i=1}^{m} \gamma_i + \xi = v \\ & L_i = \frac{\gamma_i}{s_i} - \Lambda_{ii}\ ,\ i = 1, \ldots, m \\ & U_i = \frac{\gamma_i}{s_i} + \Phi_{ii}\ ,\ i = 1, \ldots, m \\ & \begin{bmatrix} \phi_i & \beta_i \\ \beta_i & \gamma_i \end{bmatrix} \succeq 0,\ i = 1, \ldots, m \\ & \xi \geq 0,\ \Lambda \succeq 0,\ \Phi \succeq 0. \end{aligned}$$

As we know,

$$\begin{bmatrix} \phi_i & \beta_i \\ \beta_i & \gamma_i \end{bmatrix} \succeq 0 \Leftrightarrow \phi_i \geq 0,\ \gamma_i \geq 0,\ \phi_i \gamma_i \geq \beta_i^2.$$

So, when $\gamma_i > 0$, we get

$$\phi_i \geq \frac{\beta_i^2}{\gamma_i}.$$

If we suppose that, at an optimal solution,

$$\phi_i > \frac{\beta_i^2}{\gamma_i},$$

as $\beta_i^2/\gamma_i > 0$, it is obvious that we can lower that value of ϕ_i to β_i^2/γ_i obtaining yet a feasible solution, with a lower objective value. This is an absurd, so we must have $\phi_i = \beta_i^2/\gamma_i$ at an optimal solution. When $\gamma_i = 0$, we get $\beta_i = 0$. If, at an optimal solution we have $\phi_i > 0$, again we can lower the value of ϕ_i to 0 obtaining yet a feasible solution with a lower objective value. This is an absurd and, so, at an optimal solution, we must have $\phi_i = 0$.

Then, at an optimal solution,

$$\begin{aligned}
&\gamma_i = 0 \Rightarrow \beta_i = 0,\ \phi_i = 0 \\
&\gamma_i > 0 \Rightarrow \phi_i = \frac{\beta_i^2}{\gamma_i}.
\end{aligned}$$

Moreover, variable ξ can be viewed as slack variable and left out of the problem i.e.,

$$\sum_{i=1}^{m} \gamma_i + \xi = v\ ,\ \xi \geq 0\ \Leftrightarrow\ \sum_{i=1}^{m} \gamma_i \leq v.$$

Defining the sets

$$\begin{aligned}
\mathcal{I}_0 &= \{i \in \{1, \ldots, m\} : \gamma_i = 0\} \\
\mathcal{I}_> &= \{i \in \{1, \ldots, m\} : \gamma_i > 0\}\,,
\end{aligned}$$

the problem can be written as

$$(D\widetilde{DSCP}) \qquad \begin{aligned}
\min_{\beta,\gamma,\Lambda,\Phi}\ & \sum_{i \in \mathcal{I}_>} \frac{\beta_i^2}{\gamma_i} \\
s.t.\quad & f = \sum_{i=1}^{m} \beta_i b_i \\
& \sum_{i=1}^{m} \gamma_i \leq v \\
& L_i = \frac{\gamma_i}{s_i} - \Lambda_{ii},\ i = 1, \ldots, m \\
& U_i = \frac{\gamma_i}{s_i} + \Phi_{ii},\ i = 1, \ldots, m \\
& \gamma_i \geq 0,\ i = 1, \ldots, m \\
& \beta_i = 0,\ i \in \mathcal{I}_0 \\
& \Lambda \succeq 0,\ \Phi \succeq 0.
\end{aligned}$$

The following diagram summarizes the relations between all the problems obtained.

$$\begin{array}{ccccc}
(P) & \Longleftrightarrow & (CP) & \Longleftrightarrow & (SCP) \\
 & & & & \big\downarrow {\scriptstyle duality} \\
 & & (\widetilde{DSCP}) & \Longleftrightarrow & (DSCP) \\
 & & \big\downarrow {\scriptstyle duality} & & \\
 & & (D\widetilde{DSCP}) & &
\end{array}$$

It is reasonable to expect that some equivalence relation holds between problems (CP) and $(D\widetilde{DSCP})$ and thus between the original problem (P) and $(D\widetilde{DSCP})$. Consider the following change of variables in $(D\widetilde{DSCP})$:

$$\begin{cases} \gamma_i = a_i s_i \geq 0 \\ \beta_i = \gamma_i b_i^\top u \end{cases},\quad \text{for } i = 1, \ldots, m.$$

The objective function becomes,

$$\sum_{i\in\mathcal{I}_>}\frac{\beta_i^2}{\gamma_i}=\sum_{i\in\mathcal{I}_>}\frac{\beta_i\gamma_i b_i^\top u}{\gamma_i}=\sum_{i=1}^{m}\beta_i b_i^\top u=f^\top u,$$

where the last equality comes from the first constraint in $(D\widetilde{DSCP})$. For the constraints, we obtain,

$$\begin{aligned}
&\sum_{i=1}^{m}\gamma_i\le v\Leftrightarrow\sum_{i=1}^{m}a_is_i\le v,\\
&f=\sum_{i=1}^{m}\beta_i\, b_i\Leftrightarrow f=\sum_{i=1}^{m}a_is_ib_ib_i^\top u\Leftrightarrow f=\sum_{i=1}^{m}a_is_iK_iu,\\
&L_i=\frac{\gamma_i}{s_i}-\Lambda_{ii}\text{ and }\Lambda_{ii}\ge 0\;\Leftrightarrow\; a_i\ge L_i\;,\; i=1,\ldots,m,\\
&U_i=\frac{\gamma_i}{s_i}+\Phi_{ii}\text{ and }\Phi_{ii}\ge 0\;\Leftrightarrow\; a_i\le U_i\;,\; i=1,\ldots,m.
\end{aligned}$$

Clearly, problem $(\widetilde{DSCP})$ coincides with (P).

7 Computational Results and Conclusions

In this section, we describe some computational experiments we made and we present and compare the obtained results.

Used hardware

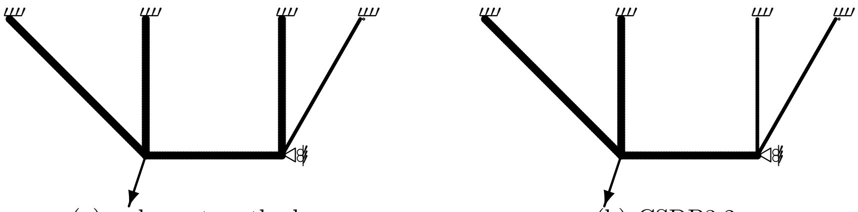
We used for all the described experiments a PC with a 1GHz Celeron processor, with 112MB of RAM, using the Windows Me operating system. The main purpose is compare the performance of the presented ε-descent algorithm with the semidefinite approach. In addition, we also made a brief comparison of the performance when we solve (SCP) and when we use $(\widetilde{DSCP})$ in the semidefinite approach.

Used software

- To solve (CP_2), we coded in PASCAL a variant of the previous described ε-descent algorithm. In this variant, we consider, instead of a constant value of ε, a strategy of beginning with a 'large' value of $\varepsilon\in[5\times 10^-6,\,5\times 10^{-1}]$, decreasing the current value after a defined number of successful iterations and increasing it after a certain amount of iterations without progress.
- To solve (SCP) and $(\widetilde{DSCP})$, we used the Brian Borchers's CSDP3.2 package, [6, 5]. This package implements a predictor-corrector variant of the primal-dual interior-point algorithm for semidefinite programming, from Helmberg, Rendl, Vanderbei and Wolkowicz.

Table 2: Results for the trusses of Figures 3 and 4.

	ε-descent method				CSDP3.2		
	# it	ε	ε-opt. value	Fig.	# it	opt. value	Fig.
Figure 3	8	$5,0 \times 10^{-5}$	$0,0897$	5(a)	24	$0,0897$	5(b)
Figure 4	588	$4,0 \times 10^{-3}$	$0,2610$	6(a)	26	$0,2616$	6(b)



(a) ε-descent method (b) CSDP3.2

Figure 5: Optimal solution for the structure in Figure 3.

Comparing the ε-descent algorithm with the semidefinite approach: used trusses, results and conclusions

For all the considered trusses, we used $E = 69$ GPa $= 6,9 \times 10^{10}$ N/m^2, the Young's modulus of the aluminium.

1) The first two cases are of a type different from the others. In these cases, the geometry is considered defined and the goal is to compute the cross sectional area of each bar.

   We used the trusses already presented in Section 1.3 at Figures 3 and 4. In both, we consider $L_i = 0$ and $U_i = 3$, $i = 1, \ldots, m$. For the truss of Figure 3, we consider the total volume, v, equal to 30 and for the truss of Figure 4, $v = 50$.

   The results are summarised at Table 2. In Figure 5, we can see graphical presentations of the optimal solutions for the truss corresponding to Figure 3. In Figure 6, the same for the truss corresponding to Figure 4.

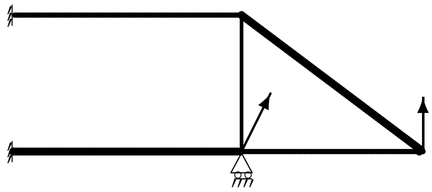
   The obtained optimal values for the truss of Figure 3 are similar. This is a consequence of the small value of the final ε. But, observing the thickness of the bars in Figure 5, it is obvious that the optimal solutions are not similar. This possibly indicates the existence of alternative optimal solutions.

   The final ε for the truss of Figure 4 is not so small and, as consequence, the optimal values are different.

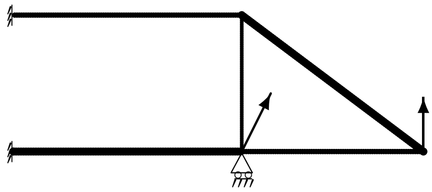
2) Several computational experiments of a different type have been made. In these experiments, we have been concerned, not only with the design of the truss, but also with its topology. We considered three basic cases and some variants of each one:

   a) A truss with a 3×11 nodes mesh, with $v = 60$, $L_i = 0$ and $U_i = 7$, $i = 1, \ldots, m$. See Figures 7-10.

   b) A truss with a 9×4 nodes mesh, with $v = 40$, $L_i = 0$ and $U_i = 4$, $i = 1, \ldots, m$. See Figures 11-14.



(a) ε-descent method



(b) CSDP3.2

Figure 6: Optimal solution for the structure in Figure 4.

Table 3: Structure 3x11, 9x4 and 13x5 corresponds to Figures 7-15.

Ground structure		**ε-Optimal solution**				**Optimal solution**		
n	m	**# iter**	ε	**ε-opt. value**	**Fig.**	**# iter**	**opt. value**	**Fig.**
62	92	460	2.03×10^{-4}	12.0974	7 (a)	31	12.0997	7 (b)
62	344	480	2.01×10^{-2}	10.5586	8 (a)	44	10.8118	8 (b)
62	92	339	1.51×10^{-3}	5.9114	9 (a)	33	5.9514	9 (b)
62	344	755	1.29×10^{-4}	5.0452	10 (a)	55	5.0464	10 (b)
54	107	1546	5.00×10^{-6}	3.3974	11 (a)	40	3.3974	11 (b)
54	409	839	5.00×10^{-6}	3.3000	12 (a)	43	3.3005	12 (b)
68	107	2304	5.00×10^{-6}	6.8582	13 (a)	41	6.8583	13 (b)
68	409	547	3.64×10^{-3}	6.6070	14 (a)	50	6.6116	14 (b)
138	242	–	–	–	–	38	14.3770	15 (a)
138	1718	–	–	–	–	44	10.5602	15 (b)

c) A truss with a 15×5 nodes mesh, with $v = 100$, $L_i = 0$ and $U_i = 3$, $i = 1, \ldots, m$. See Figure 15.

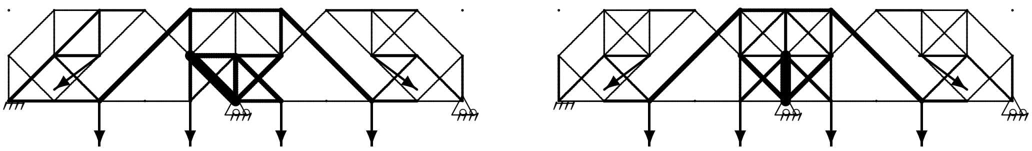
The variants were obtained considering different load patterns, different nodes support and two different ground topologies:

- The **rich topology**, where each node is connected with all the others, excluding superposition.
- The **poor topology**, where each node is only connected to its imediate neighbors.

For the 3×11 nodes structure, we considered two different load patterns. For the 9×4 nodes structure, we considered two different load patterns and nodes support.

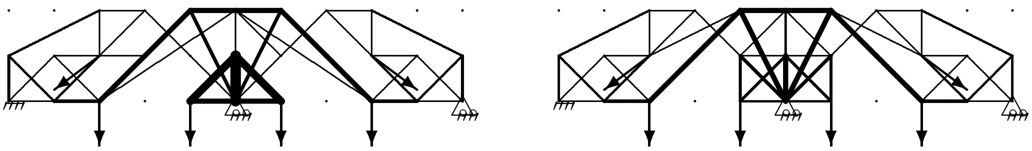
In Table 3 we present some characteristics and results of the solved examples:

- the number of degrees of freedom, n, and the number of bars, m, in the ground structure;
- the number of iterations needed to solve problem (CP_2) with the ε-descent method, the final ε, the ε-optimal value, $Z_{1\varepsilon}$, and the reference to the figure that presents the corresponding final solution;



(a) ε-descent method (b) CSDP3.2

Figure 7: Solution for the 3×11 ground structure, considering the poor topology.



(a) ε-descent method (b) CSDP3.2

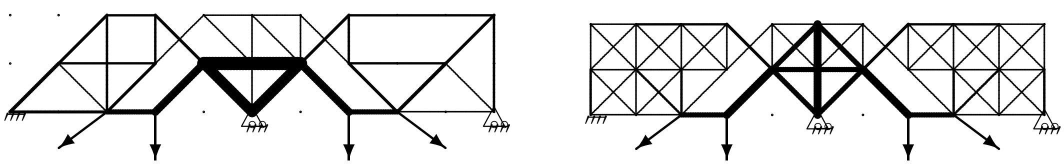
Figure 8: Solution for the 3×11 ground structure, considering the rich topology.

- the number of iterations needed to solve problem (SCP) using CSDP 3.2, the corresponding optimal value and the reference to the figure that presents the optimal solution.

Figures 7-10 are graphical presentations of the obtained optimal solutions for the 3×11 nodes structure, considering both topologies and both solution methods. The same for Figures 11-14 and for the 9×4 nodes structure. Here the variant is obtained changing, not only the load scenario, but also the nodes support. Finally, in Figure 15, we graphically present the optimal solution for the 15×5 nodes structure. In this case, the ε-descent method was not able to solve the corresponding problem: the computation time per iteration was too high.

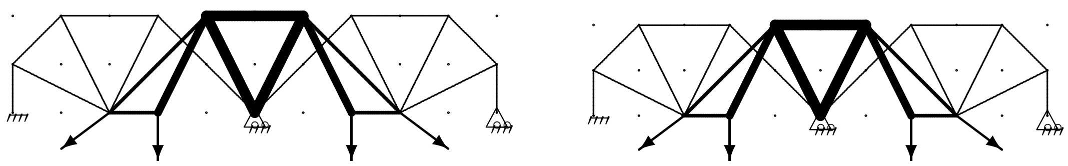
Analysing the results, we can conclude that there is a clear superiority of the semidefinite programming approach: less iterations, more precision and more solved problems.

We have also measured with a wristwatch the time needed to solve the problems and we noticed that, even in the smaller problems, the semidefinite programming approach is clearly better.



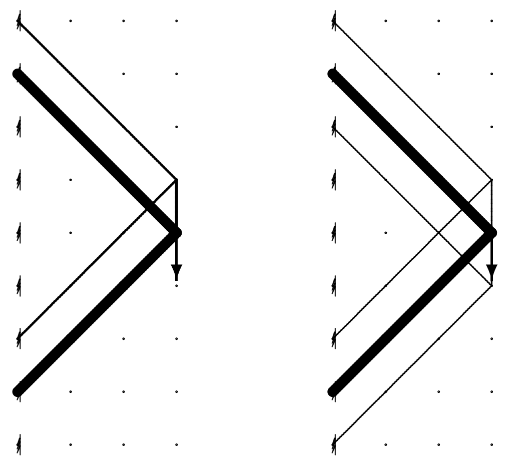
(a) ε-descent method (b) CSDP3.2

Figure 9: Same as Figure 7, but with a different load pattern.



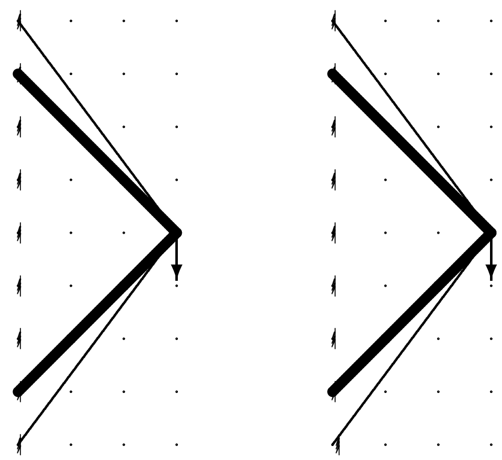
(a) ε-descent method (b) CSDP3.2

Figure 10: Same as Figure 8, but with a different load pattern.



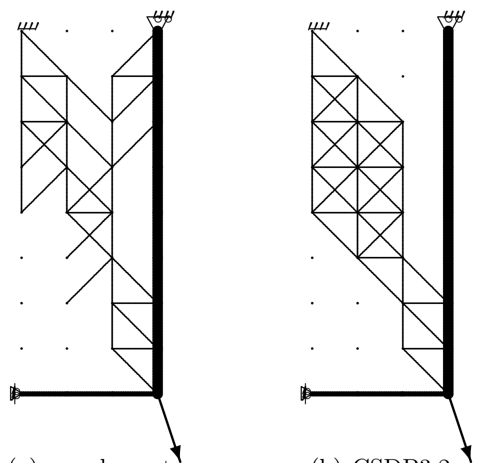
(a) ε-desc. meth. (b) CSDP3.2

Figure 11: Solution for the 9×4 ground structure, considering the poor topology.



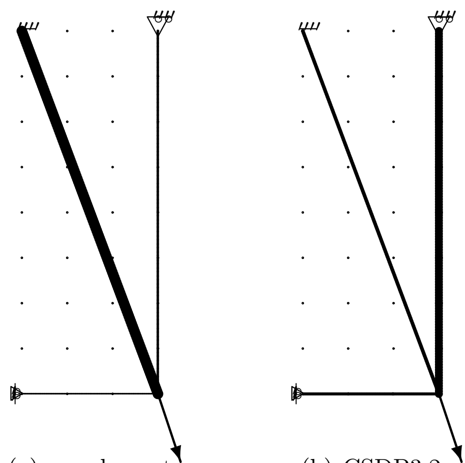
(a) ε-desc. meth. (b) CSDP3.2

Figure 12: Solution for the 9×4 ground structure, considering the rich topology.



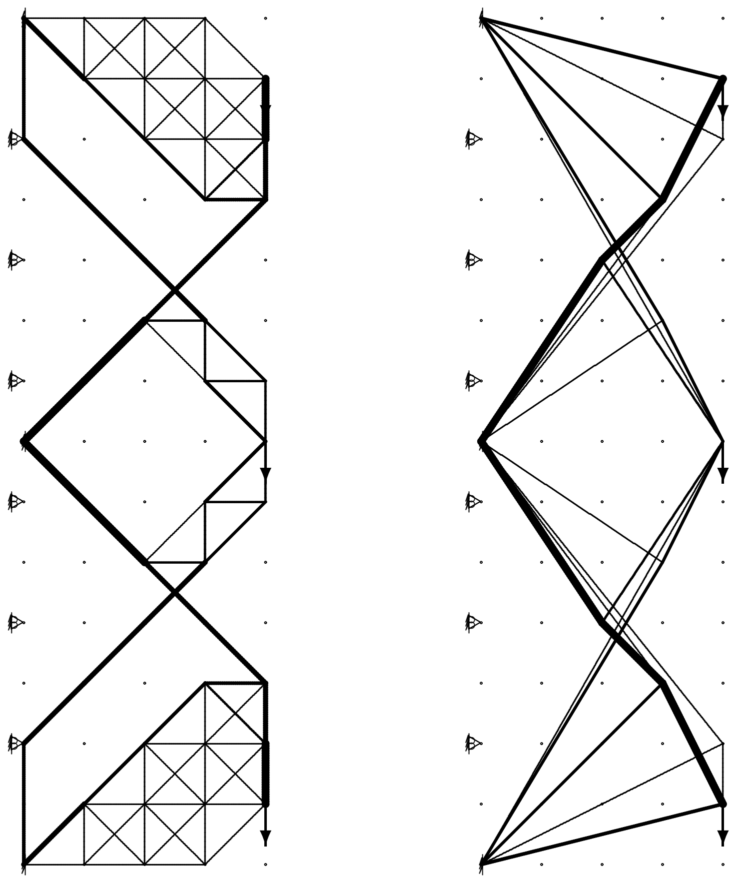
(a) ε-descent method (b) CSDP3.2

Figure 13: Same as Figure 11, but with different load pattern and nodes support.



(a) ε-descent method (b) CSDP3.2

Figure 14: Same as Figure 12, but with different pattern and nodes support, as in Figure 13.



(a) Poor topology (b) Rich topology

Figure 15: Optimal solution for the 15×5 ground structure, using CSDP3.2.

Comparing (SCP) with $(\widetilde{DSCP})$: used trusses and results

In Table 4 we compare some characteristics and the performance of both problems, (SCP) and $(\widetilde{DSCP})$, for some of the examples presented:

- the number of variables, nv, the number of blocks in the constraint matrix, nb, and their dimensions, "size of blocks";
- the number of iterations needed to solve each problem using CSDP3.2 and the reference to the figure that presents the optimal solution.

We used some of the presented trusses in the preceding experiments.

The matricial structure is very different as we can see at the columns "size of blocks". In spite of this, the only significant difference is in the number of iterations: it is always clearly greater in problem (SCP) than in problem $(\widetilde{DSCP})$. When we compare the needed time to solve the problems, there are no evidence of superiority of any of the problems.

Table 4: Comparison of (SCP) and $(\widetilde{DSCP})$.

Fig.	Problem (SCP)				Problem $(\widetilde{DSCP})$			
	nv	nb	size of blocks	# it.	nv	nb	size of blocks	# it.
7(b)	93	4	$\{63, 1, 92, 92\}$	31	245	95	{2,...,2,1,92,92}	21
8(b)	345	4	$\{63, 1, 344, 344\}$	44	751	347	{2,...,2,1,344,344}	26
9(b)	93	4	$\{63, 1, 92, 92\}$	33	245	95	{2,...,2,1,92,92}	23
10(b)	345	4	$\{63, 1, 344, 344\}$	55	751	347	{2,...,2,1,344,344}	27
11(b)	108	4	$\{55, 1, 107, 107\}$	40	269	110	{2,...,2,1,107,107}	25
12(b)	410	4	$\{55, 1, 409, 409\}$	43	873	412	{2,...,2,1,409,409}	29
13(b)	108	4	$\{69, 1, 107, 107\}$	41	283	110	{2,...,2,1,107,107}	26
14(b)	410	4	$\{69, 1, 409, 409\}$	50	887	502	{2,...,2,1,409,409}	26
15(a)	243	4	$\{139, 1, 242, 242\}$	38	623	245	$\{2, \ldots, 2, 1, 242, 242\}$	27
15(b)	1719	4	$\{139, 1, 1718, 1718\}$	44	3575	1721	{2,...,2,1,1718,1718}	32

References

[1] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, John Wiley & Sons, New York, 1993.

[2] A. Ben-Tal and M. P. Bendsøe, *A new method for optimal truss topology design*, SIAM Journal Optimization, 3 (1993), pp. 322–358.

[3] A. Ben-Tal and A. Nemirovski, *Potential reduction polynomial time method for truss topology design*, SIAM Journal Optimization, 4 (1994), pp. 596–612.

[4] A. Ben-Tal and A. Nemirovski, *Optimal design of engineering structures*, Optima, 47 (1995), pp. 4–8.

[5] B. Borchers, *CSDP, 3.2 User's Guide*, Optimization Methods and Software, 11 (1999), pp. 597–611.

[6] B. Borchers, *CSDP, A C library for Semidefinite Programming*, Optimization Methods and Software, 11 (1999), pp. 613–623.

[7] C. M. Branco, *Mecânica dos Materiais*, Fundação Calouste Gulbenkian, Lisboa, 1994.

[8] M. A. C. Cerveira, *Optimização do desenho de estruturas*, master's thesis, Universidade de Lisboa, Portugal, 1997.

[9] E. W. Cheney and A. A. Goldstein, *A Newton's method for convex programming and Tchebycheff approximation*, Numeric Mathematics, 1 (1959), pp. 253–268.

[10] E. de Klerk, C. Roos, and T. Terlaky, *Semi-definite problems in truss topology optimization*, Tech. Report Nr. 95-128, Faculty of Technical Mathematics and Informatics, Delft University of Technology, November 1995.

[11] C. Helmberg, *Semidefinite programming for combinatorial optimization*, tech. report, Konrad-Zuse-Zentrum für Informationstecghnik Berlin, 2000.

[12] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algoritms I: Fundamentals*, vol. 305 of A Series of Comprehensive Studies in Mathematics, Springer-Verlag, Berlin, 1993.

[13] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algoritms II: Advanced Theory and Bundle Methods*, vol. 305 of A Series of Comprehensive Studies in Mathematics, Springer-Verlag, Berlin, 1993.

[14] R. A. Horn and C. R. Jonhson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.

[15] R. A. Horn and C. R. Jonhson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, 1991.

[16] J. E. Kelley, *The cutting plane method for solving convex problems*, Journal of the Society for the Industrial and Applied Mathematics, 8 (1960), pp. 703–712.

[17] U. Kirsch, *Optimum Structural Design: Concepts, Methods and Applications*, McGraw-Hill, New York, 1981.

[18] D. G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley, Reading Massachusetts, 1984.

[19] J. M. Mulvey, R. J. Vanderbei, and S. A. Zenios, *Robust optimization of large-scale systems*, Operations Research, 43 (1995), pp. 264–281.

[20] M. Patriksson and J. Petersson, *A subgradient method for contact structural optimization*, LiTH-MAT-R-1995-25, (1995).

[21] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, New Jersey, 1970.

[22] N. Z. Shor, *Convergence rate of the gradient descent method with dilation of the space*, Cambridge, 6 (1970), pp. 102–108.

[23] N. Z. Shor, *Utilization of the operation of space dilation in the minimization of convex functions*, Cambridge, 6 (1970), pp. 7–15.

[24] N. Z. Shor, *Cut-off method with space extension in convex programming problems*, Cambridge, 13 (1977), pp. 94–96.

[25] N. Z. Shor, *Minimization Methods for Non-Differentiable Functions*, Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1985.

[26] G. N. Vanderplaats, *Numerical Optimization Techniques for Engineering Design: With Applications*, Series in Mechanical Engineering, McGraw-Hill, New York, 1984.